openSUSE

10.3

www.novell.com

September 27, 2007

Reference

SUSE
A NOVELL BUSINESS

Reference

All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither Novell, Inc., SUSE LINUX Products GmbH, the authors, nor the translators shall be held liable for possible errors or the consequences thereof.

Contents

About This Guide xiii

Part I Advanced Deployment Scenarios 1

1 Remote Installation 3

1.1 Installation Scenarios for Remote Installation . . . 3
1.2 Setting Up the Server Holding the Installation Sources . . . 12
1.3 Preparing the Boot of the Target System . . . 22
1.4 Booting the Target System for Installation . . . 31
1.5 Monitoring the Installation Process . . . 36

2 Advanced Disk Setup 41

2.1 Using the YaST Partitioner . . . 41
2.2 LVM Configuration . . . 49
2.3 Soft RAID Configuration . . . 55

Part II Administration 61

3 Online Update 63

3.1 YaST Online Update . . . 63
3.2 Update from the Command Line with zypper . . . 66

4 YaST in Text Mode 69

4.1 Navigation in Modules . . . 70
4.2 Restriction of Key Combinations . . . 71

4.3 YaST Command Line Options 72

5 Updating the System and System Changes 75

5.1 Updating the System 75
5.2 Software Changes from Version to Version 78

6 RPM—the Package Manager 85

6.1 Verifying Package Authenticity 86
6.2 Managing Packages: Install, Update, and Uninstall 86
6.3 RPM and Patches 87
6.4 Delta RPM Packages 89
6.5 RPM Queries 90
6.6 Installing and Compiling Source Packages 93
6.7 Compiling RPM Packages with build 95
6.8 Tools for RPM Archives and the RPM Database 96

7 Printer Operation 97

7.1 The Workflow of the Printing System 99
7.2 Methods and Protocols for Connecting Printers 99
7.3 Installing the Software 100
7.4 Network Printers 100
7.5 Graphical Printing Interfaces 103
7.6 Printing from the Command Line 104
7.7 Special Features in openSUSE 104
7.8 Troubleshooting 108

8 The X Window System 117

8.1 Manually Configuring the X Window System 117
8.2 Installing and Configuring Fonts 123
8.3 For More Information 129

9 Xen Virtualization 131

9.1 Basic Components 131
9.2 Setting Up a Virtual Machine Host 135
9.3 Setting Up Virtual Machines 139
9.4 Managing a Virtualization Environment 146

10 System Monitoring Utilities 151

10.1 Debugging 152

10.2 Files and File Systems 154
10.3 Hardware Information 156
10.4 Networking 159
10.5 The /proc File System 160
10.6 Processes 163
10.7 System Information 167
10.8 User Information 171
10.9 Time and Date 171

Part III System 173

11 32-Bit and 64-Bit Applications in a 64-Bit System Environment 175

11.1 Runtime Support 175
11.2 Software Development 176
11.3 Software Compilation on Biarch Platforms 176
11.4 Kernel Specifications 178

12 Booting and Configuring a Linux System 179

12.1 The Linux Boot Process 179
12.2 The init Process 183
12.3 System Configuration via /etc/sysconfig 192

13 The Boot Loader 195

13.1 Selecting a Boot Loader 196
13.2 Booting with GRUB 196
13.3 Configuring the Boot Loader with YaST 205
13.4 Uninstalling the Linux Boot Loader 209
13.5 Creating Boot CDs 209
13.6 The Graphical SUSE Screen 210
13.7 Troubleshooting 211
13.8 For More Information 213

14 Special System Features 215

14.1 Information about Special Software Packages 215
14.2 Virtual Consoles 222
14.3 Keyboard Mapping 222
14.4 Language and Country-Specific Settings 223

15 Dynamic Kernel Device Management with udev 227

15.1 The /dev Directory . . . 227
15.2 Kernel uevents and udev . . . 228
15.3 Drivers, Kernel Modules, and Devices . . . 228
15.4 Booting and Initial Device Setup . . . 229
15.5 Monitoring the Running udev Daemon . . . 229
15.6 Influencing Kernel Device Event Handling with udev Rules . . . 231
15.7 Persistent Device Naming . . . 237
15.8 Files used by udev . . . 238
15.9 For More Information . . . 239

16 File Systems in Linux 241

16.1 Terminology . . . 241
16.2 Major File Systems in Linux . . . 242
16.3 Some Other Supported File Systems . . . 247
16.4 Large File Support in Linux . . . 248
16.5 For More Information . . . 249

17 Access Control Lists in Linux 251

17.1 Traditional File Permissions . . . 251
17.2 Advantages of ACLs . . . 253
17.3 Definitions . . . 253
17.4 Handling ACLs . . . 254
17.5 ACL Support in Applications . . . 262
17.6 For More Information . . . 262

18 Authentication with PAM 263

18.1 Structure of a PAM Configuration File . . . 264
18.2 The PAM Configuration of sshd . . . 266
18.3 Configuring PAM Using pam-config . . . 268
18.4 For More Information . . . 269

19 Working with the Shell 271

19.1 Using the Bash Shell . . . 271
19.2 Users and Access Permissions . . . 277
19.3 Important Linux Commands . . . 280
19.4 The vi Editor . . . 291

Part IV Services 297

20 Basic Networking 299

20.1 IP Addresses and Routing 302
20.2 IPv6—The Next Generation Internet 305
20.3 Name Resolution 314
20.4 Configuring a Network Connection with YaST 315
20.5 Configuring a Network Connection Manually 332
20.6 smpppd as Dial-up Assistant 345

21 SLP Services in the Network 349

21.1 Installation 349
21.2 Activating SLP 350
21.3 SLP Front-Ends in openSUSE 350
21.4 Installation over SLP 351
21.5 Providing Services via SLP 351
21.6 For More Information 352

22 The Domain Name System 353

22.1 DNS Terminology 353
22.2 Installation 354
22.3 Configuration with YaST 354
22.4 Starting the Name Server BIND 361
22.5 The Configuration File /etc/named.conf 363
22.6 Zone Files 367
22.7 Dynamic Update of Zone Data 372
22.8 Secure Transactions 372
22.9 DNS Security 373
22.10 For More Information 374

23 DHCP 375

23.1 Configuring a DHCP Server with YaST 376
23.2 DHCP Software Packages 387
23.3 The DHCP Server dhcpd 388
23.4 For More Information 391

24 Time Synchronization with NTP 393

24.1 Configuring an NTP Client with YaST 393
24.2 Configuring xntp in the Network 398
24.3 Setting Up a Local Reference Clock 398

25 Using NIS 401

25.1 Configuring NIS Servers . . . 401
25.2 Configuring NIS Clients . . . 407

26 LDAP—A Directory Service 409

26.1 LDAP versus NIS . . . 410
26.2 Structure of an LDAP Directory Tree . . . 411
26.3 Configuring an LDAP Server with YaST . . . 414
26.4 Configuring an LDAP Client with YaST . . . 419
26.5 Configuring LDAP Users and Groups in YaST . . . 426
26.6 Browsing the LDAP Directory Tree . . . 428
26.7 Manually Configuring an LDAP Server . . . 429
26.8 Manually Administering LDAP Data . . . 435
26.9 For More Information . . . 439

27 Active Directory Support 441

27.1 Integrating Linux and AD Environments . . . 441
27.2 Background Information for Linux AD Support . . . 442
27.3 Configuring a Linux Client for Active Directory . . . 447
27.4 Logging In to an AD Domain . . . 450
27.5 Changing Passwords . . . 452

28 Sharing File Systems with NFS 455

28.1 NFS Client Configuration . . . 455
28.2 Exporting File Systems over NFS . . . 459
28.3 Exporting File Systems Manually . . . 463
28.4 NFS with Kerberos . . . 466
28.5 For More Information . . . 466

29 Samba 469

29.1 Terminology . . . 469
29.2 Installing a Samba Server . . . 471
29.3 Starting and Stopping Samba . . . 471
29.4 Configuring a Samba Server . . . 471
29.5 Configuring Clients . . . 477
29.6 Samba as Login Server . . . 478
29.7 For More Information . . . 479

30 The Apache HTTP Server 481

30.1 Quick Start . . . 481
30.2 Configuring Apache . . . 483
30.3 Starting and Stopping Apache . . . 497
30.4 Installing, Activating, and Configuring Modules . . . 499
30.5 Getting CGI Scripts to Work . . . 507
30.6 Setting Up a Secure Web Server with SSL . . . 509
30.7 Avoiding Security Problems . . . 515
30.8 Troubleshooting . . . 517
30.9 For More Information . . . 518

31 Setting up a FTP server with YaST 521

31.1 Starting the FTP server . . . 521
31.2 FTP General Settings . . . 522
31.3 FTP Performance Settings . . . 523
31.4 Authentication . . . 523
31.5 Expert Settings . . . 524
31.6 For more information . . . 524

Part V Mobility 525

32 Power Management 527

32.1 Power Saving Functions . . . 527
32.2 ACPI . . . 528
32.3 Rest for the Hard Disk . . . 536
32.4 The powersave Package . . . 537

33 Wireless Communication 543

33.1 Wireless LAN . . . 543
33.2 Bluetooth . . . 553
33.3 Infrared Data Transmission . . . 560

34 Using Tablet PCs 565

34.1 Installing Tablet PC Packages . . . 566
34.2 Configuring Your Tablet Device . . . 567
34.3 Using the Virtual Keyboard . . . 567
34.4 Rotating Your Display . . . 568
34.5 Using Gesture Recognition . . . 568
34.6 Taking Notes and Sketching with the Pen . . . 569
34.7 Troubleshooting . . . 571

34.8 For More Information 573

Part VI Security 575

35 Masquerading and Firewalls 577

35.1 Packet Filtering with iptables 577
35.2 Masquerading Basics 580
35.3 Firewalling Basics 581
35.4 SuSEfirewall2 582
35.5 For More Information 587

36 SSH—Secure Network Operations 589

36.1 The OpenSSH Package 589
36.2 The ssh Program 590
36.3 scp—Secure Copy 590
36.4 sftp—Secure File Transfer 591
36.5 The SSH Daemon (sshd)—Server-Side 591
36.6 SSH Authentication Mechanisms 592
36.7 X, Authentication, and Forwarding Mechanisms 593
36.8 Configuring An SSH Daemon with YaST 594

37 Managing X.509 Certification 597

37.1 The Principles of Digital Certification 597
37.2 YaST Modules for CA Management 602

38 Network Authentication—Kerberos 613

38.1 Kerberos Terminology 613
38.2 How Kerberos Works 615
38.3 Users' View of Kerberos 618
38.4 For More Information 619

39 Installing and Administering Kerberos 621

39.1 Kerberos Network Topology 622
39.2 Choosing the Kerberos Realms 623
39.3 Setting Up the KDC Hardware 624
39.4 Configuring Time Synchronization 625
39.5 Configuring the KDC 626
39.6 Configuring Kerberos Clients 628
39.7 Configuring Remote Kerberos Administration 634

39.8 Creating Kerberos Service Principals 636
39.9 Enabling PAM Support for Kerberos 637
39.10 Configuring SSH for Kerberos Authentication 638
39.11 Using LDAP and Kerberos 639

40 Encrypting Partitions and Files 643

40.1 Setting Up an Encrypted File System with YaST 644
40.2 Using Encrypted Home Directories 647
40.3 Using vi to Encrypt Single Files 649

41 Confining Privileges with AppArmor 651

41.1 Installing Novell AppArmor 652
41.2 Enabling and Disabling Novell AppArmor 652
41.3 Getting Started with Profiling Applications 654

42 Security and Confidentiality 661

42.1 Local Security and Network Security 662
42.2 Some General Security Tips and Tricks 671
42.3 Using the Central Security Reporting Address 673

A An Example Network 675

B GNU Licenses 677

B.1 GNU General Public License 677
B.2 GNU Free Documentation License 680

Index 685

About This Guide

This manual gives you a general understanding of openSUSE®. It is intended mainly for system administrators and home users with basic system administration knowledge. Check out the various parts of this manuals for a selection of applications needed in everyday life and in-depth descriptions of advanced installation and configuration scenarios.

Advanced Deployment Scenarios
: Learn how to deploy openSUSE from a remote location and become acquainted with complex disk setup scenarios.

Administration
: Learn how to update and configure your openSUSE, how to administrate your system from a remote location, and get to know some important utilities for Linux administrators.

System
: Get an introduction to the components of your Linux system and a deeper understanding of their interaction.

Services
: Learn how to configure the various network and file services that come with openSUSE.

Mobility
: Get an introduction to mobile computing with openSUSE and learn how to configure the various options for wireless computing, power management, and profile management.

Security
: Become acquainted with openSUSE security features and learn how to setup and configure services that will make your system secure.

1 Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation and enter your comments there.

2 Additional Documentation

We provide HTML and PDF versions of our books in different languages. The following manuals are available on this product:

Start-Up
: Guides you through the installation and basic configuration of your system. For newcomers, the manual also introduces basic Linux concepts such as the file system, the user concept and access permissions and gives an overview of the features openSUSE offers to support mobile computing. Provides help and advice in troubleshooting.

KDE Quick Start
: Gives a short introduction to the KDE desktop and some key applications running on it.

KDE User Guide
: Introduces the KDE desktop of openSUSE and a variety of applications shipping with it. It guides you through using these applications and helps you perform key tasks. It is intended mainly for end users who want to make efficient use of KDE in everyday life.

GNOME Quick Start
: Gives a short introduction to the GNOME desktop and some key applications running on it.

GNOME User Guide
: Introduces the GNOME desktop of openSUSE and a variety of applications you will encounter when working with the GNOME desktop. It guides you through using these applications and helps you perform key tasks. It is intended mainly for end users who want to make efficient use of applications running on the GNOME desktop.

Reference
Gives you a general understanding of openSUSE and covers advanced system administration tasks. It is intended mainly for system administrators and home users with basic system administration knowledge. It provides detailed information about advanced deployment scenarios, administration of your system, the interaction of key system components and the set-up of various network and file services openSUSE offers.

Novell AppArmor Quick Start
Helps you understand the main concepts behind Novell® AppArmor.

Novell AppArmor Administration Guide
Contains in-depth information about the use of Novell AppArmor in your environment.

Lessons For Lizards
A community book project for the openSUSE distribution. A snapshot of the manual written by the open source community is released on an equal footing with the Novell/SUSE manuals. The lessons are written in a cook book style and cover more specific or exotic topics than the traditional manuals. For more information, see `http://developer.novell.com/wiki/index.php/Lessons_for_Lizards`.

Find HTML versions of the openSUSE manuals in your installed system under `/usr/share/doc/manual` or in the help centers of your KDE or GNOME desktop. You can also access the documentation on the Web at `http://www.novell.com/documentation/opensuse103/` where you can download PDF or HTML versions of the manuals. For information where to find the books on your installation media, refer to the Release Notes of this product, available from your installed system under `/usr/share/doc/release-notes/`.

3 Documentation Conventions

The following typographical conventions are used in this manual:

- `/etc/passwd`: filenames and directory names

- *`placeholder`*: replace *`placeholder`* with the actual value

- `PATH`: the environment variable PATH
- `ls`, `--help`: commands, options, and parameters
- `user`: users or groups
- Alt, Alt + F1: a key to press or a key combination; keys are shown in uppercase as on a keyboard
- *File*, *File* > *Save As*: menu items, buttons
- *Dancing Penguins* (Chapter *Penguins*, ↑Another Manual): This is a reference to a chapter in another manual.

4 About the Making of This Manual

This book is written in Novdoc, a subset of DocBook (see `http://www.docbook.org`). The XML source files were validated by `xmllint`, processed by `xsltproc`, and converted into HTML using a customized version of Norman Walsh's stylesheets.

5 Source Code

The source code of openSUSE is publicly available. To download the source code, proceed as outlined under `http://www.novell.com/products/suselinux/source_code.html`. If requested we send you the source code on a DVD. We need to charge a $15 or €15 fee for creation, handling and postage. To request a DVD of the source code, send an e-mail to sourcedvd@suse.de [`mailto:sourcedvd@suse.de`] or mail the request to:

```
SUSE Linux Products GmbH
Product Management openSUSE
Maxfeldstr. 5
D-90409 Nürnberg
Germany
```

6 Acknowledgments

With a lot of voluntary commitment, the developers of Linux cooperate on a global scale to promote the development of Linux. We thank them for their efforts—this distribution would not exist without them. Furthermore, we thank Frank Zappa and Pawar. Special thanks, of course, go to Linus Torvalds.

Have a lot of fun!

Your SUSE Team

Part I. Advanced Deployment Scenarios

Remote Installation 1

openSUSE® can be installed in several different ways. As well as the usual media installation covered in Chapter 1, *Installation with YaST* (↑Start-Up), you can choose from various network-based approaches or even take a completely hands-off approach to the installation of openSUSE.

Each method is introduced by means of two short check lists: one listing the prerequisites for this method and the other illustrating the basic procedure. More detail is then provided for all the techniques used in these installation scenarios.

NOTE

In the following sections, the system to hold your new openSUSE installation is referred to as *target system* or *installation target*. The term *installation source* is used for all sources of installation data. This includes physical media, such as CD and DVD, and network servers distributing the installation data in your network.

1.1 Installation Scenarios for Remote Installation

This section introduces the most common installation scenarios for remote installations. For each scenario, carefully check the list of prerequisites and follow the procedure outlined for this scenario. If in need of detailed instructions for a particular step, follow the links provided for each one of them.

IMPORTANT

The configuration of the X Window System is not part of any remote installation process. After the installation has finished, log in to the target system as `root`, enter `telinit 3`, and start SaX2 to configure the graphics hardware as described in Section "Setting Up Graphics Card and Monitor" (Chapter 2, *Setting Up Hardware Components with YaST*, ↑Start-Up).

1.1.1 Simple Remote Installation via VNC—Static Network Configuration

This type of installation still requires some degree of physical access to the target system to boot for installation. The installation itself is entirely controlled by a remote workstation using VNC to connect to the installation program. User interaction is required as with the manual installation in Chapter 1, *Installation with YaST* (↑Start-Up).

For this type of installation, make sure that the following requirements are met:

- Remote installation source: NFS, HTTP, FTP, or SMB with working network connection
- Target system with working network connection
- Controlling system with working network connection and VNC viewer software or Java-enabled browser (Firefox, Konqueror, Internet Explorer, or Opera)
- Physical boot medium (CD or DVD) for booting the target system
- Valid static IP addresses already assigned to the installation source and the controlling system
- Valid static IP address to assign to the target system

To perform this kind of installation, proceed as follows:

1. Set up the installation source as described in Section 1.2, "Setting Up the Server Holding the Installation Sources" (page 12). Choose an NFS, HTTP, or FTP network server. For an SMB installation source, refer to Section 1.2.5, "Managing an SMB Installation Source" (page 20).

2. Boot the target system using the first CD or DVD of the openSUSE media kit.

3. When the boot screen of the target system appears, use the boot options prompt to set the appropriate VNC options and the address of the installation source. This is described in detail in Section 1.4, "Booting the Target System for Installation" (page 31).

   The target system boots to a text-based environment, giving the network address and display number under which the graphical installation environment can be addressed by any VNC viewer application or browser. VNC installations announce themselves over OpenSLP and can be found using Konqueror in `service:/` or `slp:/` mode.

4. On the controlling workstation, open a VNC viewing application or Web browser and connect to the target system as described in Section 1.5.1, "VNC Installation" (page 36).

5. Perform the installation as described in Chapter 1, *Installation with YaST* (↑Start-Up). Reconnect to the target system after it reboots for the final part of the installation.

6. Finish the installation.

1.1.2 Simple Remote Installation via VNC—Dynamic Network Configuration

This type of installation still requires some degree of physical access to the target system to boot for installation. The network configuration is made with DHCP. The installation itself is entirely controlled from a remote workstation using VNC to connect to the installer, but still requires user interaction for the actual configuration efforts.

For this type of installation, make sure that the following requirements are met:

- Remote installation source: NFS, HTTP, FTP, or SMB with working network connection

- Target system with working network connection

- Controlling system with working network connection and VNC viewer software or Java-enabled browser (Firefox, Konqueror, Internet Explorer, or Opera)
- Physical boot medium (CD, DVD, or custom boot disk) for booting the target system
- Running DHCP server providing IP addresses

To perform this kind of installation, proceed as follows:

1. Set up the installation source as described in Section 1.2, “Setting Up the Server Holding the Installation Sources” (page 12). Choose an NFS, HTTP, or FTP network server. For an SMB installation source, refer to Section 1.2.5, “Managing an SMB Installation Source” (page 20).

2. Boot the target system using the first CD or DVD of the openSUSE media kit.

3. When the boot screen of the target system appears, use the boot options prompt to set the appropriate VNC options and the address of the installation source. This is described in detail in Section 1.4, “Booting the Target System for Installation” (page 31).

   The target system boots to a text-based environment, giving the network address and display number under which the graphical installation environment can be addressed by any VNC viewer application or browser. VNC installations announce themselves over OpenSLP and can be found using Konqueror in `service:/` or `slp:/` mode.

4. On the controlling workstation, open a VNC viewing application or Web browser and connect to the target system as described in Section 1.5.1, “VNC Installation” (page 36).

5. Perform the installation as described in Chapter 1, *Installation with YaST* (↑Start-Up). Reconnect to the target system after it reboots for the final part of the installation.

6. Finish the installation.

1.1.3 Remote Installation via VNC—PXE Boot and Wake on LAN

This type of installation is completely hands-off. The target machine is started and booted remotely. User interaction is only needed for the actual installation. This approach is suitable for cross-site deployments.

To perform this type of installation, make sure that the following requirements are met:

- Remote installation source: NFS, HTTP, FTP, or SMB with working network connection
- TFTP server
- Running DHCP server for your network
- Target system capable of PXE boot, networking, and Wake on LAN, plugged in and connected to the network
- Controlling system with working network connection and VNC viewer software or Java-enabled browser (Firefox, Konqueror, Internet Explorer, or Opera)

To perform this type of installation, proceed as follows:

1. Set up the installation source as described in Section 1.2, "Setting Up the Server Holding the Installation Sources" (page 12). Choose an NFS, HTTP, or FTP network server or configure an SMB installation source as described in Section 1.2.5, "Managing an SMB Installation Source" (page 20).
2. Set up a TFTP server to hold a boot image that can be pulled by the target system. This is described in Section 1.3.2, "Setting Up a TFTP Server" (page 24).
3. Set up a DHCP server to provide IP addresses to all machines and reveal the location of the TFTP server to the target system. This is described in Section 1.3.1, "Setting Up a DHCP Server" (page 22).
4. Prepare the target system for PXE boot. This is described in further detail in Section 1.3.5, "Preparing the Target System for PXE Boot" (page 30).

5 Initiate the boot process of the target system using Wake on LAN. This is described in Section 1.3.7, “Wake on LAN” (page 31).

6 On the controlling workstation, open a VNC viewing application or Web browser and connect to the target system as described in Section 1.5.1, “VNC Installation” (page 36).

7 Perform the installation as described in Chapter 1, *Installation with YaST* (↑Start-Up). Reconnect to the target system after it reboots for the final part of the installation.

8 Finish the installation.

1.1.4 Simple Remote Installation via SSH—Static Network Configuration

This type of installation still requires some degree of physical access to the target system to boot for installation and to determine the IP address of the installation target. The installation itself is entirely controlled from a remote workstation using SSH to connect to the installer. User interaction is required as with the regular installation described in Chapter 1, *Installation with YaST* (↑Start-Up).

For this type of installation, make sure that the following requirements are met:

- Remote installation source: NFS, HTTP, FTP, or SMB with working network connection

- Target system with working network connection

- Controlling system with working network connection and working SSH client software

- Physical boot medium (CD, DVD, or custom boot disk) for the target system

- Valid static IP addresses already assigned to the installation source and the controlling system

- Valid static IP address to assign to the target system

To perform this kind of installation, proceed as follows:

1 Set up the installation source as described in Section 1.2, “Setting Up the Server Holding the Installation Sources” (page 12). Choose an NFS, HTTP, or FTP network server. For an SMB installation source, refer to Section 1.2.5, “Managing an SMB Installation Source” (page 20).

2 Boot the target system using the first CD or DVD of the openSUSE media kit.

3 When the boot screen of the target system appears, use the boot options prompt to set the appropriate parameters for network connection, address of the installation source, and SSH enablement. This is described in detail in Section 1.4.3, “Using Custom Boot Options” (page 33).

   The target system boots to a text-based environment, giving the network address under which the graphical installation environment can be addressed by any SSH client.

4 On the controlling workstation, open a terminal window and connect to the target system as described in Section “Connecting to the Installation Program” (page 38).

5 Perform the installation as described in Chapter 1, *Installation with YaST* (↑Start-Up). Reconnect to the target system after it reboots for the final part of the installation.

6 Finish the installation.

1.1.5 Simple Remote Installation via SSH—Dynamic Network Configuration

This type of installation still requires some degree of physical access to the target system to boot for installation and determine the IP address of the installation target. The installation itself is entirely controlled from a remote workstation using VNC to connect to the installer, but still requires user interaction for the actual configuration efforts.

For this type of installation, make sure that the following requirements are met:

- Remote installation source: NFS, HTTP, FTP, or SMB with working network connection
- Target system with working network connection
- Controlling system with working network connection and working SSH client software
- Physical boot medium (CD or DVD) for booting the target system
- Running DHCP server providing IP addresses

To perform this kind of installation, proceed as follows:

1. Set up the installation source as described in Section 1.2, "Setting Up the Server Holding the Installation Sources" (page 12). Choose an NFS, HTTP, or FTP network server. For an SMB installation source, refer to Section 1.2.5, "Managing an SMB Installation Source" (page 20).

2. Boot the target system using the first CD or DVD of the openSUSE media kit.

3. When the boot screen of the target system appears, use the boot options prompt to pass the appropriate parameters for network connection, location of the installation source, and SSH enablement. See Section 1.4.3, "Using Custom Boot Options" (page 33) for detailed instructions on the use of these parameters.

   The target system boots to a text-based environment, giving you the network address under which the graphical installation environment can be addressed by any SSH client.

4. On the controlling workstation, open a terminal window and connect to the target system as described in Section "Connecting to the Installation Program" (page 38).

5. Perform the installation as described in Chapter 1, *Installation with YaST* (↑Start-Up). Reconnect to the target system after it reboots for the final part of the installation.

6. Finish the installation.

1.1.6 Remote Installation via SSH—PXE Boot and Wake on LAN

This type of installation is completely hands-off. The target machine is started and booted remotely.

To perform this type of installation, make sure that the following requirements are met:

- Remote installation source: NFS, HTTP, FTP, or SMB with working network connection

- TFTP server

- Running DHCP server for your network, providing a static IP to the host to install

- Target system capable of PXE boot, networking, and Wake on LAN, plugged in and connected to the network

- Controlling system with working network connection and SSH client software

To perform this type of installation, proceed as follows:

1. Set up the installation source as described in Section 1.2, "Setting Up the Server Holding the Installation Sources" (page 12). Choose an NFS, HTTP, or FTP network server. For the configuration of an SMB installation source, refer to Section 1.2.5, "Managing an SMB Installation Source" (page 20).

2. Set up a TFTP server to hold a boot image that can be pulled by the target system. This is described in Section 1.3.2, "Setting Up a TFTP Server" (page 24).

3. Set up a DHCP server to provide IP addresses to all machines and reveal the location of the TFTP server to the target system. This is described in Section 1.3.1, "Setting Up a DHCP Server" (page 22).

4. Prepare the target system for PXE boot. This is described in further detail in Section 1.3.5, "Preparing the Target System for PXE Boot" (page 30).

5. Initiate the boot process of the target system using Wake on LAN. This is described in Section 1.3.7, "Wake on LAN" (page 31).

6. On the controlling workstation, start an SSH client and connect to the target system as described in Section 1.5.2, “SSH Installation” (page 38).

7. Perform the installation as described in Chapter 1, *Installation with YaST* (↑Start-Up). Reconnect to the target system after it reboots for the final part of the installation.

8. Finish the installation.

1.2 Setting Up the Server Holding the Installation Sources

Depending on the operating system running on the machine to use as network installation source for openSUSE, there are several options for the server configuration. The easiest way to set up an installation server is to use YaST on SUSE Linux 9.3 and higher. On other versions of openSUSE, set up the installation source manually.

TIP

You can even use a Microsoft Windows machine as installation server for your Linux deployment. See Section 1.2.5, “Managing an SMB Installation Source” (page 20) for details.

1.2.1 Setting Up an Installation Server Using YaST

YaST offers a graphical tool for creating network installation sources. It supports HTTP, FTP, and NFS network installation servers.

1. Log in as `root` to the machine that should act as installation server.

2. Install the `yast2-instserver` package.

3. Start *YaST* > *Miscellaneous* > *Installation Server*.

4 Select the server type (HTTP, FTP, or NFS). The selected server service is started automatically every time the system starts. If a service of the selected type is already running on your system and you want to configure it manually for the server, deactivate the automatic configuration of the server service with *Do Not Configure Any Network Services*. In both cases, define the directory in which the installation data should be made available on the server.

5 Configure the required server type. This step relates to the automatic configuration of server services. It is skipped when automatic configuration is deactivated.

Define an alias for the root directory of the FTP or HTTP server on which the installation data should be found. The installation source will later be located under `ftp://`*`Server-IP/Alias/Name`* (FTP) or under `http://`*`Server-IP/Alias/Name`* (HTTP). *`Name`* stands for the name of the installation source, which is defined in the following step. If you selected NFS in the previous step, define wild cards and export options. The NFS server will be accessible under `nfs://`*`Server-IP/Name`*. Details of NFS and exports can be found in Chapter 28, *Sharing File Systems with NFS* (page 455).

TIP: Firewall Settings

Make sure that the firewall settings of your server system allow traffic on the ports for HTTP, NFS, and FTP. If they currently do not, start the YaST firewall module and open the respective ports.

6 Configure the installation source. Before the installation media are copied to their destination, define the name of the installation source (ideally, an easily remembered abbreviation of the product and version). YaST allows providing ISO images of the media instead of copies of the installation CDs. If you want this, activate the relevant check box and specify the directory path under which the ISO files can be found locally. Depending on the product to distribute using this installation server, it might be that more add-on CDs or service pack CDs are required and should be added as extra installation sources. To announce your installation server in the network via OpenSLP, activate the appropriate option.

TIP

Consider announcing your installation source via OpenSLP if your network setup supports this option. This saves you from entering the network installation path on every target machine. The target systems are just booted using the SLP boot option and find the network installation source without any further configuration. For details on this option, refer to Section 1.4, "Booting the Target System for Installation" (page 31).

7 Upload the installation data. The most lengthy step in configuring an installation server is copying the actual installation CDs. Insert the media in the sequence requested by YaST and wait for the copying procedure to end. When the sources have been fully copied, return to the overview of existing information sources and close the configuration by selecting *Finish*.

   Your installation server is now fully configured and ready for service. It is automatically started every time the system is started. No further intervention is required. You only need to configure and start this service correctly by hand if you have deactivated the automatic configuration of the selected network service with YaST as an initial step.

To deactivate an installation source, select the installation source to remove then select *Delete*. The installation data are removed from the system. To deactivate the network service, use the respective YaST module.

If your installation server should provide the installation data for more than one product of product version, start the YaST installation server module and select *Add* in the overview of existing installation sources to configure the new installation source.

1.2.2 Setting Up an NFS Installation Source Manually

Setting up an NFS source for installation is basically done in two steps. In the first step, create the directory structure holding the installation data and copy the installation media over to this structure. Second, export the directory holding the installation data to the network.

To create a directory holding the installation data, proceed as follows:

1 Log in as `root`.

2 Create a directory that should later hold all installation data and change into this directory. For example:

```
mkdir install/product/productversion

cd install/product/productversion
```

Replace *`product`* with an abbreviation of the product name and *`productversion`* with a string that contains the product name and version.

3 For each CD contained in the media kit execute the following commands:

3a Copy the entire content of the installation CD into the installation server directory:

```
cp -a /media/path_to_your_CD-ROM_drive .
```

Replace *`path_to_your_CD-ROM_drive`* with the actual path under which your CD or DVD drive is addressed. Depending on the type of drive used in your system, this can be `cdrom`, `cdrecorder`, `dvd`, or `dvdrecorder`.

3b Rename the directory to the CD number:

```
mv path_to_your_CD-ROM_drive CDx
```

Replace *`x`* with the actual number of your CD.

On openSUSE, you can export the installation sources with NFS using YaST. Proceed as follows:

1 Log in as `root`.

2 Start *YaST* > *Network Services* > *NFS Server*.

3 Select *Start* and *Open Port in Firewall* and click *Next*.

4 Select *Add Directory* and browse for the directory containing the installation sources, in this case, *`productversion`*.

5. Select *Add Host* and enter the hostnames of the machines to which to export the installation data. Instead of specifying hostnames here, you could also use wild cards, ranges of network addresses, or just the domain name of your network. Enter the appropriate export options or leave the default, which works fine in most setups. For more information about the syntax used in exporting NFS shares, read the `exports` man page.

6. Click *Finish*. The NFS server holding the openSUSE installation sources is automatically started and integrated into the boot process.

If you prefer manually exporting the installation sources via NFS instead of using the YaST NFS Server module, proceed as follows:

1. Log in as `root`.

2. Open the file `/etc/exports` and enter the following line:

   ```
   /productversion *(ro,root_squash,sync)
   ```

   This exports the directory `/productversion` to any host that is part of this network or to any host that can connect to this server. To limit the access to this server, use netmasks or domain names instead of the general wild card `*`. Refer to the `export` man page for details. Save and exit this configuration file.

3. To add the NFS service to the list of servers started during system boot, execute the following commands:

   ```
   insserv /etc/init.d/nfsserver
   insserv /etc/init.d/portmap
   ```

4. Start the NFS server with `rcnfsserver start`. If you need to change the configuration of your NFS server later, modify the configuration file and restart the NFS daemon with `rcnfsserver restart`.

Announcing the NFS server via OpenSLP makes its address known to all clients in your network.

1. Log in as `root`.

2. Enter the directory `/etc/slp.reg.d/`.

3. Create a configuration file called `install.suse.nfs.reg` containing the following lines:

   ```
   # Register the NFS Installation Server
   service:install.suse:nfs://$HOSTNAME/path_to_instsource/CD1,en,65535
   description=NFS Installation Source
   ```

   Replace *`path_to_instsource`* with the actual path to the installation source on your server.

4. Save this configuration file and start the OpenSLP daemon with `rcslpd start`.

For more information about OpenSLP, refer to the package documentation located under `/usr/share/doc/packages/openslp/` or refer to Chapter 21, *SLP Services in the Network* (page 349).

1.2.3 Setting Up an FTP Installation Source Manually

Creating an FTP installation source is very similar to creating an NFS installation source. FTP installation sources can be announced over the network using OpenSLP as well.

1. Create a directory holding the installation sources as described in Section 1.2.2, “Setting Up an NFS Installation Source Manually” (page 14).

2. Configure the FTP server to distribute the contents of your installation directory:

   2a Log in as `root` and install the package `vsftpd` using the YaST package manager.

   2b Enter the FTP server root directory:

   ```
   cd /srv/ftp
   ```

   2c Create a subdirectory holding the installation sources in the FTP root directory:

   ```
   mkdir instsource
   ```

   Replace *`instsource`* with the product name.

2d Mount the contents of the installation repository into the change root environment of the FTP server:

```
mount --bind path_to_instsource /srv/ftp/instsource
```

Replace *`path_to_instsource`* and *`instsource`* with values matching your setup. If you need to make this permanent, add it to `/etc/fstab`.

2e Start vsftpd with `vsftpd`.

3 Announce the installation source via OpenSLP, if this is supported by your network setup:

3a Create a configuration file called `install.suse.ftp.reg` under `/etc/slp/reg.d/` that contains the following lines:

```
# Register the FTP Installation Server
service:install.suse:ftp://$HOSTNAME/srv/ftp/instsource/CD1,en,65535
description=FTP Installation Source
```

Replace *`instsource`* with the actual name to the installation source directory on your server. The `service:` line should be entered as one continuous line.

3b Save this configuration file and start the OpenSLP daemon with `rcslpd start`.

TIP: Configuring an FTP Server with YaST

If you prefer using YaST over manually configuring the FTP installation server, refer to Chapter 31, *Setting up a FTP server with YaST* (page 521) for more information on how to use the YaST FTP server module.

1.2.4 Setting Up an HTTP Installation Source Manually

Creating an HTTP installation source is very similar to creating an NFS installation source. HTTP installation sources can be announced over the network using OpenSLP as well.

1 Create a directory holding the installation sources as described in Section 1.2.2, "Setting Up an NFS Installation Source Manually" (page 14).

2 Configure the HTTP server to distribute the contents of your installation directory:

2a Install the Web server Apache as described in Section 30.1.2, "Installation" (page 482).

2b Enter the root directory of the HTTP server (`/srv/www/htdocs`) and create a subdirectory that will hold the installation sources:

```
mkdir instsource
```

Replace *`instsource`* with the product name.

2c Create a symbolic link from the location of the installation sources to the root directory of the Web server (`/srv/www/htdocs`):

```
ln -s /path_instsource /srv/www/htdocs/instsource
```

2d Modify the configuration file of the HTTP server (`/etc/apache2/default-server.conf`) to make it follow symbolic links. Replace the following line:

```
Options None
```

with

```
Options Indexes FollowSymLinks
```

2e Reload the HTTP server configuration using `rcapache2 reload`.

3 Announce the installation source via OpenSLP, if this is supported by your network setup:

3a Create a configuration file called `install.suse.http.reg` under `/etc/slp/reg.d/` that contains the following lines:

```
# Register the HTTP Installation Server
service:install.suse:http://$HOSTNAME/srv/www/htdocs/instsource/CD1/,en,65535
description=HTTP Installation Source
```

Replace *`instsource`* with the actual path to the installation source on your server. The `service:` line should be entered as one continuous line.

3b Save this configuration file and start the OpenSLP daemon using `rcslpd restart`.

1.2.5 Managing an SMB Installation Source

Using SMB, you can import the installation sources from a Microsoft Windows server and start your Linux deployment even with no Linux machine around.

To set up an exported Windows Share holding your openSUSE installation sources, proceed as follows:

1 Log in to your Windows machine.

2 Start Explorer and create a new folder that will hold the entire installation tree and name it `INSTALL`, for example.

3 Export this share according the procedure outlined in your Windows documentation.

4 Enter this share and create a subfolder, called *`product`*. Replace *`product`* with the actual product name.

5 Enter the `INSTALL/`*`product`* folder and copy each CD or DVD to a separate folder, such as `CD1` and `CD2`.

To use a SMB mounted share as installation source, proceed as follows:

1. Boot the installation target.

2. Select *Installation*.

3. Press F4 for a selection of installation sources.

4. Choose SMB and enter the Windows machine's name or IP address, the share name (`INSTALL/`*`product`*`/CD1`, in this example), username, and password.

   After you hit Enter, YaST starts and you can perform the installation.

1.2.6 Using ISO Images of the Installation Media on the Server

Instead of copying physical media into your server directory manually, you can also mount the ISO images of the installation media into your installation server and use them as installation source. To set up an HTTP, NFS or FTP server that uses ISO images instead of media copies, proceed as follows:

1. Download the ISO images and save them to the machine to use as the installation server.

2. Log in as `root`.

3. Choose and create an appropriate location for the installation data, as described in Section 1.2.2, "Setting Up an NFS Installation Source Manually" (page 14), Section 1.2.3, "Setting Up an FTP Installation Source Manually" (page 17), or Section 1.2.4, "Setting Up an HTTP Installation Source Manually" (page 19).

4. Create subdirectories for each CD or DVD.

5. To mount and unpack each ISO image to the final location, issue the following command:

   ```
   mount -o loop path_to_iso path_to_instsource/product/mediumx
   ```

   Replace *`path_to_iso`* with the path to your local copy of the ISO image, *`path_to_instsource`* with the source directory of your server, *`product`* with the product name, and *mediumx* with the type (CD or DVD) and number of media you are using.

6 Repeat the previous step to mount all ISO images needed for your product.

7 Start your installation server as usual, as described in Section 1.2.2, “Setting Up an NFS Installation Source Manually” (page 14), Section 1.2.3, “Setting Up an FTP Installation Source Manually” (page 17), or Section 1.2.4, “Setting Up an HTTP Installation Source Manually” (page 19).

1.3 Preparing the Boot of the Target System

This section covers the configuration tasks needed in complex boot scenarios. It contains ready-to-apply configuration examples for DHCP, PXE boot, TFTP, and Wake on LAN.

1.3.1 Setting Up a DHCP Server

A DHCP server on openSUSE is set up by manually editing the appropriate configuration files. This section covers extending an existing DHCP server configuration to provide the data needed to serve in a TFTP, PXE, and WOL environment.

Setting Up a DHCP Server Manually

All the DHCP server needs to do, apart from providing automatic address allocation to your network clients, is to announce the IP address of the TFTP server and the file that should be pulled in by the installation routines on the target machine.

1 Log in as `root` to the machine hosting the DHCP server.

2 Append the following lines to your DHCP server's configuration file located under `/etc/dhcpd.conf`:

```
group {
  # PXE related stuff
```

```
      #
      # "next server" defines the tftp server that will be used
      next server ip_tftp_server:
      #
      # "filename" specifies the pxelinux image on the tftp server
      # the server runs in chroot under /srv/tftpboot
      filename  "pxelinux.0";
    }
```

Replace *`ip_of_the_tftp_server`* with the actual IP address of the TFTP server. For more information about the options available in `dhcpd.conf`, refer to the `dhcpd.conf` manual page.

3 Restart the DHCP server by executing `rcdhcpd restart`.

If you plan on using SSH for the remote control of a PXE and Wake on LAN installation, explicitly specify the IP address DHCP should provide to the installation target. To achieve this, modify the above-mentioned DHCP configuration according to the following example:

```
group {
  # PXE related stuff
  #
  # "next server" defines the tftp server that will be used
  next server ip_tftp_server:
  #
  # "filename" specifies the pxelinux image on the tftp server
  # the server runs in chroot under /srv/tftpboot
  filename "pxelinux.0";
  host test { hardware ethernet mac_address;
              fixed-address some_ip_address; }
       }
```

The host statement introduces the hostname of the installation target. To bind the hostname and IP address to a specific host, you must know and specify the system's hardware (MAC) address. Replace all the variables used in this example with the actual values that match your environment.

After restarting the DHCP server, it provides a static IP to the host specified, enabling you to connect to the system via SSH.

1.3.2 Setting Up a TFTP Server

Set up a TFTP server with YaST or set it up manually on any other Linux operating system that supports xinetd and tftp. The TFTP server delivers the boot image to the target system once it boots and sends a request for it.

Setting Up a TFTP Server Using YaST

1. Log in as `root`.

2. Install the `yast2-tftp-server` package.

3. Start *YaST* > *Network Services* > *TFTP Server* and install the requested package.

4. Click *Enable* to make sure that the server is started and included in the boot routines. No further action from your side is required to secure this. xinetd starts tftpd at boot time.

5. Click *Open Port in Firewall* to open the appropriate port in the firewall running on your machine. If there is no firewall running on your server, this option is not available.

6. Click *Browse* to browse for the boot image directory. The default directory `/tftpboot` is created and selected automatically.

7. Click *Finish* to apply your settings and start the server.

Setting Up a TFTP Server Manually

1. Log in as `root` and install the packages `tftp` and `xinetd`.

2. If unavailable, create `/srv/tftpboot` and `/srv/tftpboot/pxelinux.cfg` directories.

3. Add the appropriate files needed for the boot image as described in Section 1.3.3, "Using PXE Boot" (page 25).

4. Modify the configuration of xinetd located under `/etc/xinetd.d/` to make sure that the TFTP server is started on boot:

4a If it does not exist, create a file called `tftp` under this directory with `touch tftp`. Then run `chmod 755 tftp`.

4b Open the file `tftp` and add the following lines:

```
service tftp
{
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /srv/tftpboot
        disable                 = no
}
```

4c Save the file and restart xinetd with `rcxinetd restart`.

1.3.3 Using PXE Boot

Some technical background information as well as PXE's complete specifications are available in the Preboot Execution Environment (PXE) Specification (http://www.pix.net/software/pxeboot/archive/pxespec.pdf).

1 Change to the directory of your installation repository and copy the `linux`, `initrd`, `message`, and `memtest` files to the `/srv/tftpboot` directory by entering the following:

```
cp -a boot/loader/linux boot/loader/initrd
      boot/loader/message boot/loader/memtest /srv/tftpboot
```

2 Install the `syslinux` package directly from your installation CDs or DVDs with YaST.

3 Copy the `/usr/share/syslinux/pxelinux.0` file to the `/srv/tftpboot` directory by entering the following:

```
cp -a /usr/share/syslinux/pxelinux.0 /srv/tftpboot
```

4. Change to the directory of your installation repository and copy the `isolinux.cfg` file to `/srv/tftpboot/pxelinux.cfg/default` by entering the following:

   ```
   cp -a boot/loader/isolinux.cfg /srv/tftpboot/pxelinux.cfg/default
   ```

5. Edit the `/srv/tftpboot/pxelinux.cfg/default` file and remove the lines beginning with `gfxboot`, `readinfo`, and `framebuffer`.

6. Insert the following entries in the append lines of the default `failsafe` and `apic` labels:

   `insmod=`*`kernel module`*

   By means of this entry, enter the network kernel module needed to support network installation on the PXE client. Replace *`kernel module`* with the appropriate module name for your network device.

   `netdevice=`*`interface`*

   This entry defines the client's network interface that must be used for the network installation. It is only necessary if the client is equipped with several network cards and must be adapted accordingly. In case of a single network card, this entry can be omitted.

   `install=nfs://`*`ip_instserver`*`/`*`path_instsource`*`/CD1`

   This entry defines the NFS server and the installation source for the client installation. Replace *`ip_instserver`* with the actual IP address of your installation server. *`path_instsource`* should be replaced with the actual path to the installation sources. HTTP, FTP, or SMB sources are addressed in a similar manner, except for the protocol prefix, which should read `http`, `ftp`, or `smb`.

   IMPORTANT

   If you need to pass other boot options to the installation routines, such as SSH or VNC boot parameters, append them to the `install` entry. An overview of parameters and some examples are given in Section 1.4, “Booting the Target System for Installation” (page 31).

An example /srv/tftpboot/pxelinux.cfg/default file follows. Adjust the protocol prefix for the installation source to match your network setup and specify your preferred method of connecting to the installer by adding the vnc and vncpassword or the usessh and sshpassword options to the install entry. The lines separated by \ must be entered as one continuous line without a line break and without the \.

```
default linux

# default
label linux
  kernel linux
     append initrd=initrd ramdisk_size=65536 insmod=e100 \
     install=nfs://ip_instserver/path_instsource/product/CD1

# failsafe
label failsafe
  kernel linux
  append initrd=initrd ramdisk_size=65536 ide=nodma apm=off acpi=off \
  insmod=e100 install=nfs://ip_instserver/path_instsource/product/CD1

# apic
label apic
  kernel linux
  append initrd=initrd ramdisk_size=65536 apic insmod=e100 \
  install=nfs://ip_instserver/path_instsource/product/CD1

# manual
label manual
  kernel linux
  append initrd=initrd ramdisk_size=65536 manual=1

# rescue
label rescue
  kernel linux
  append initrd=initrd ramdisk_size=65536 rescue=1

#  memory test
label memtest
  kernel memtest

# hard disk
label harddisk
  localboot 0

implicit     0
display      message
prompt       1
timeout      100
```

Replace *`ip_instserver`* and *`path_instsource`* with the values used in your setup.

The following section serves as a short reference to the PXELINUX options used in this setup. Find more information about the options available in the documentation of the `syslinux` package located under `/usr/share/doc/packages/syslinux/`.

1.3.4 PXELINUX Configuration Options

The options listed here are a subset of all the options available for the PXELINUX configuration file.

`DEFAULT` *`kernel options`*`...`

Sets the default kernel command line. If PXELINUX boots automatically, it acts as if the entries after DEFAULT had been typed in at the boot prompt, except the auto option is automatically added, indicating an automatic boot.

If no configuration file is present or no DEFAULT entry is present in the configuration file, the default is the kernel name “linux” with no options.

`APPEND` *`options...`*

Add one or more options to the kernel command line. These are added for both automatic and manual boots. The options are added at the very beginning of the kernel command line, usually permitting explicitly entered kernel options to override them.

`LABEL` *`label`* `KERNEL` *`image`* `APPEND` *`options`*`...`

Indicates that if *`label`* is entered as the kernel to boot, PXELINUX should instead boot *`image`* and the specified `APPEND` options should be used instead of the ones specified in the global section of the file (before the first `LABEL` command). The default for *`image`* is the same as *`label`* and, if no `APPEND` is given, the default is to use the global entry (if any). Up to 128 `LABEL` entries are permitted.

Note that GRUB uses the following syntax:

```
title mytitle
  kernel my_kernel my_kernel_options
  initrd myinitrd
```

PXELINUX uses the following syntax:

```
label mylabel
  kernel mykernel
  append myoptions
```

Labels are mangled as if they were filenames and they must be unique after mangling. For example, the two labels “v2.1.30” and “v2.1.31” would not be distinguishable under PXELINUX because both mangle to the same DOS filename.

The kernel does not have to be a Linux kernel; it can be a boot sector or a COMBOOT file.

`APPEND -`

Append nothing. `APPEND` with a single hyphen as argument in a `LABEL` section can be used to override a global `APPEND`.

`LOCALBOOT` *`type`*

On PXELINUX, specifying `LOCALBOOT 0` instead of a `KERNEL` option means invoking this particular label and causes a local disk boot instead of a kernel boot.

Argument	Description
0	Perform a normal boot
4	Perform a local boot with the Universal Network Driver Interface (UNDI) driver still resident in memory
5	Perform a local boot with the entire PXE stack, including the UNDI driver, still resident in memory

All other values are undefined. If you do not know what the UNDI or PXE stacks are, specify `0`.

`TIMEOUT` *`time-out`*

Indicates how long to wait at the boot prompt until booting automatically, in units of 1/10 second. The time-out is canceled as soon as the user types anything on the keyboard, assuming the user will complete the command begun. A time-out of zero disables the time-out completely (this is also the default). The maximum possible time-out value is 35996 (just less than one hour).

`PROMPT` *`flag_val`*

If `flag_val` is 0, displays the boot prompt only if Shift or Alt is pressed or Caps Lock or Scroll Lock is set (this is the default). If `flag_val` is 1, always displays the boot prompt.

```
F2  filename
F1  filename
..etc...
F9  filename
F10 filename
```

Displays the indicated file on the screen when a function key is pressed at the boot prompt. This can be used to implement preboot online help (presumably for the kernel command line options). For backward compatibility with earlier releases, F10 can be also entered as F0. Note that there is currently no way to bind filenames to F11 and F12.

1.3.5 Preparing the Target System for PXE Boot

Prepare the system's BIOS for PXE boot by including the PXE option in the BIOS boot order.

WARNING: BIOS Boot Order

Do not place the PXE option ahead of the hard disk boot option in the BIOS. Otherwise this system would try to reinstall itself every time you boot it.

1.3.6 Preparing the Target System for Wake on LAN

Wake on LAN (WOL) requires the appropriate BIOS option to be enabled prior to the installation. Also, note down the MAC address of the target system. This data is needed to initiate Wake on LAN.

1.3.7 Wake on LAN

Wake on LAN allows a machine to be turned on by a special network packet containing the machine's MAC address. Because every machine in the world has a unique MAC identifier, you do not need to worry about accidentally turning on the wrong machine.

IMPORTANT: Wake on LAN across Different Network Segments

If the controlling machine is not located in the same network segment as the installation target that should be awakened, either configure the WOL requests to be sent as multicasts or remotely control a machine on that network segment to act as the sender of these requests.

1.3.8 Manual Wake on LAN

1. Log in as `root`.

2. Start *YaST* > *Software Management* and install the package `netdiag`.

3. Open a terminal and enter the following command as `root` to wake the target:

   ```
   ether-wake  mac_of_target
   ```

   Replace *`mac_of_target`* with the actual MAC address of the target.

1.4 Booting the Target System for Installation

Basically, there are two different ways to customize the boot process for installation apart from those mentioned under Section 1.3.7, "Wake on LAN" (page 31) and Section 1.3.3, "Using PXE Boot" (page 25). You can either use the default boot options and function keys or use the boot options prompt of the installation boot screen to pass any boot options that the installation kernel might need on this particular hardware.

1.4.1 Using the Default Boot Options

The boot options are described in detail in Chapter 1, *Installation with YaST* (↑Start-Up). Generally, just selecting *Installation* starts the installation boot process.

If problems occur, use *Installation—ACPI Disabled* or *Installation—Safe Settings*. For more information about troubleshooting the installation process, refer to Section “Installation Problems” (Chapter 13, *Common Problems and Their Solutions*, ↑Start-Up).

1.4.2 Using the F Keys

The menu bar at the bottom screen offers some advanced functionality needed in some setups. Using the F keys, you can specify additional options to pass to the installation routines without having to know the detailed syntax of these parameters (see Section 1.4.3, “Using Custom Boot Options” (page 33)).

See the table below for a complete set of the options available.

***Table 1.1** F Keys During Installation*

Key	Purpose	Available Options
F1	Provide help	None
F2	Select the installation language	All supported languages
F3	Change screen resolution for installation	• Text mode • VESA • resolution #1 • resolution #2 • ...

Key	Purpose	Available Options
F4	Select the installation source	• CD-ROM or DVD • SLP • FTP • HTTP • NFS • SMB • Hard Disk
F5	Choose appropriate kernel options	• Default • No ACPI • No local APIC • Installation—ACPI Disabled • Installation—Safe Settings
F6	Apply driver update disk	Driver

1.4.3 Using Custom Boot Options

Using the appropriate set of boot options helps facilitate your installation procedure. Many parameters can also be configured later using the linuxrc routines, but using the boot options is easier. In some automated setups, the boot options can be provided with `initrd` or an `info` file.

The following table lists all installation scenarios mentioned in this chapter with the required parameters for booting and the corresponding boot options. Just append all of them in the order they appear in this table to get one boot option string that is handed to the installation routines. For example (all in one line):

```
install=... netdevice=... hostip=...netmask=... vnc=... vncpassword=...
```

Replace all the values (...) in this string with the values appropriate for your setup.

***Table 1.2** Installation (Boot) Scenarios Used in This Chapter*

Installation Scenario	Parameters Needed for Booting	Boot Options
Chapter 1, *Installation with YaST* (↑Start-Up)	None: system boots automatically	None needed
Section 1.1.1, "Simple Remote Installation via VNC—Static Network Configuration" (page 4)	• Location of the installation server • Network device • IP address • Netmask • Gateway • VNC enablement • VNC password	• `install=(nfs,http,ftp,smb)://`*`path_to_instmedia`* • `netdevice=`*`some_netdevice`* (only needed if several network devices are available) • `hostip=`*`some_ip`* • `netmask=`*`some_netmask`* • `gateway=`*`ip_gateway`* • `vnc=1` • `vncpassword=`*`some_password`*
Section 1.1.2, "Simple Remote Installation via VNC—Dynamic Network Configuration" (page 5)	• Location of the installation server • VNC enablement • VNC password	• `install=(nfs,http,ftp,smb)://`*`path_to_instmedia`* • `vnc=1` • `vncpassword=`*`some_password`*

Installation Scenario	Parameters Needed for Booting	Boot Options
Section 1.1.3, “Remote Installation via VNC—PXE Boot and Wake on LAN” (page 7)	• Location of the installation server • Location of the TFTP server • VNC enablement • VNC password	Not applicable; process managed through PXE and DHCP
Section 1.1.4, “Simple Remote Installation via SSH—Static Network Configuration” (page 8)	• Location of the installation server • Network device • IP address • Netmask • Gateway • SSH enablement • SSH password	• `install=(nfs,http,ftp,smb)://`*`path_to_instmedia`* • `netdevice=`*`some_netdevice`* (only needed if several network devices are available) • `hostip=`*`some_ip`* • `netmask=`*`some_netmask`* • `gateway=`*`ip_gateway`* • `usessh=1` • `sshpassword=`*`some_password`*
Section 1.1.5, “Simple Remote Installation via SSH—Dynamic Network Configuration” (page 9)	• Location of the installation server • SSH enablement • SSH password	• `install=(nfs,http,ftp,smb)://`*`path_to_instmedia`* • `usessh=1` • `sshpassword=`*`some_password`*
Section 1.1.6, “Remote Installation via SSH—PXE Boot and	• Location of the installation server	Not applicable; process managed through PXE and DHCP

Installation Scenario	Parameters Needed for Booting	Boot Options
Wake on LAN" (page 11)	• Location of the TFTP server • SSH enablement • SSH password	

TIP: More Information about linuxrc Boot Options

Find more information about the linuxrc boot options used for booting a Linux system in `/usr/share/doc/packages/linuxrc/linuxrc.html`.

1.5 Monitoring the Installation Process

There are several options for remotely monitoring the installation process. If the proper boot options have been specified while booting for installation, either VNC or SSH can be used to control the installation and system configuration from a remote workstation.

1.5.1 VNC Installation

Using any VNC viewer software, you can remotely control the installation of openSUSE from virtually any operating system. This section introduces the setup using a VNC viewer application or a Web browser.

Preparing for VNC Installation

All you need to do on the installation target to prepare for a VNC installation is to provide the appropriate boot options at the initial boot for installation (see Section 1.4.3, "Using Custom Boot Options" (page 33)). The target system boots into a text-based environment and waits for a VNC client to connect to the installation program.

The installation program announces the IP address and display number needed to connect for installation. If you have physical access to the target system, this information is provided right after the system booted for installation. Enter this data when your VNC client software prompts for it and provide your VNC password.

Because the installation target announces itself via OpenSLP, you can retrieve the address information of the installation target via an SLP browser without the need for any physical contact to the installation itself provided your network setup and all machines support OpenSLP:

1. Start the KDE file and Web browser Konqueror.

2. Enter `service://yast.installation.suse` in the location bar. The target system then appears as an icon in the Konqueror screen. Clicking this icon launches the KDE VNC viewer in which to perform the installation. Alternatively, run your VNC viewer software with the IP address provided and add `:1` at the end of the IP address for the display the installation is running on.

Connecting to the Installation Program

Basically, there are two ways to connect to a VNC server (the installation target in this case). You can either start an independent VNC viewer application on any operating system or connect using a Java-enabled Web browser.

Using VNC, you can control the installation of a Linux system from any other operating system, including other Linux flavors, Windows, or Mac OS.

On a Linux machine, make sure that the package `tightvnc` is installed. On a Windows machine, install the Windows port of this application, which can be obtained at the TightVNC home page (`http://www.tightvnc.com/download.html`).

To connect to the installation program running on the target machine, proceed as follows:

1. Start the VNC viewer.

2. Enter the IP address and display number of the installation target as provided by the SLP browser or the installation program itself:

   ```
   ip_address:display_number
   ```

A window opens on your desktop displaying the YaST screens as in a normal local installation.

Using a Web browser to connect to the installation program makes you totally independent of any VNC software or the underlying operating system. As long as the browser application has Java support enabled, you can use any browser (Firefox, Internet Explorer, Konqueror, Opera, etc.) to perform the installation of your Linux system.

To perform a VNC installation, proceed as follows:

1. Launch your preferred Web browser.

2. Enter the following at the address prompt:

   ```
   http://ip_address_of_target:5801
   ```

3. Enter your VNC password when prompted to do so. The browser window now displays the YaST screens as in a normal local installation.

1.5.2 SSH Installation

Using SSH, you can remotely control the installation of your Linux machine using any SSH client software.

Preparing for SSH Installation

Apart from installing the appropriate software package (OpenSSH for Linux and PuTTY for Windows), you just need to pass the appropriate boot options to enable SSH for installation. See Section 1.4.3, "Using Custom Boot Options" (page 33) for details. OpenSSH is installed by default on any SUSE Linux–based operating system.

Connecting to the Installation Program

1. Retrieve the installation target's IP address. If you have physical access to the target machine, just take the IP address the installation routine provides at the console after the initial boot. Otherwise take the IP address that has been assigned to this particular host in the DHCP server configuration.

2. At a command line, enter the following command:

```
ssh -X root@ip_address_of_target
```

Replace `ip_address_of_target` with the actual IP address of the installation target.

3 When prompted for a username, enter `root`.

4 When prompted for the password, enter the password that has been set with the SSH boot option. After you have successfully authenticated, a command line prompt for the installation target appears.

5 Enter `yast` to launch the installation program. A window opens showing the normal YaST screens as described in Chapter 1, *Installation with YaST* (↑Start-Up).

Advanced Disk Setup

2

Sophisticated system configurations require particular disk setups. All common partitioning tasks may be done with YaST. To get persistent device naming with block devices, use the devices below `/dev/disk/by-id` or `/dev/disk/by-uuid`. Logical Volume Management (LVM) is a disk partitioning scheme that is designed to be much more flexible than the physical partitioning used in standard setups. Its snapshot functionality enables you to create data backups easily. Redundant Array of Independent Disks (RAID) offers increased data integrity, performance, and fault tolerance.

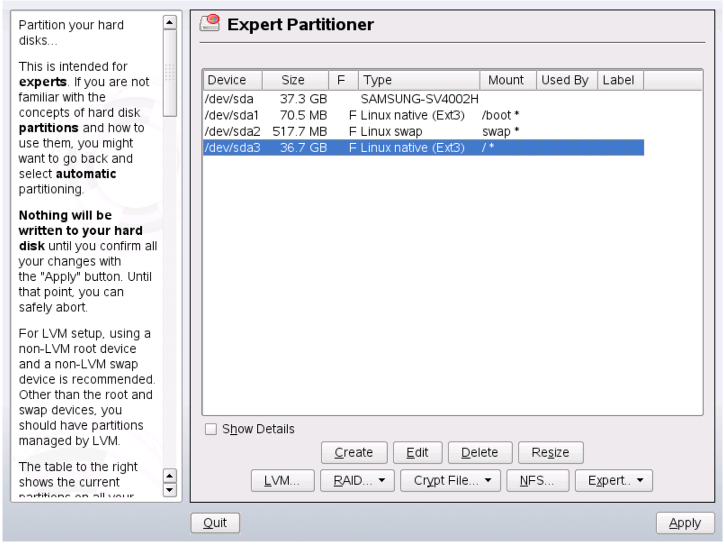
2.1 Using the YaST Partitioner

With the expert partitioner, shown in Figure 2.1, “The YaST Partitioner” (page 42), manually modify the partitioning of one or several hard disks. Partitions can be added, deleted, resized, and edited. Also access the soft RAID and LVM configuration from this YaST module.

WARNING: Repartitioning the Running System

Although it is possible to repartition your system while it is running, the risk of making a mistake that causes data loss is very high. Try to avoid repartitioning your installed system and always do a complete backup of your data before attempting to do so.

Figure 2.1 *The YaST Partitioner*



Partition your hard disks...

This is intended for **experts**. If you are not familiar with the concepts of hard disk **partitions** and how to use them, you might want to go back and select **automatic** partitioning.

Nothing will be written to your hard disk until you confirm all your changes with the "Apply" button. Until that point, you can safely abort.

For LVM setup, using a non-LVM root device and a non-LVM swap device is recommended. Other than the root and swap devices, you should have partitions managed by LVM.

The table to the right shows the current

Device	Size	F	Type	Mount	Used By	Label
/dev/sda	37.3 GB		SAMSUNG-SV4002H			
/dev/sda1	70.5 MB	F	Linux native (Ext3)	/boot *		
/dev/sda2	517.7 MB	F	Linux swap	swap *		
/dev/sda3	36.7 GB	F	Linux native (Ext3)	/ *		

All existing or suggested partitions on all connected hard disks are displayed in the list of the YaST *Expert Partitioner* dialog. Entire hard disks are listed as devices without numbers, such as `/dev/sda`. Partitions are listed as parts of these devices, such as `/dev/sda1`. The size, type, file system, and mount point of the hard disks and their partitions are also displayed. The mount point describes where the partition appears in the Linux file system tree.

If you run the expert dialog during installation, any free hard disk space is also listed and automatically selected. To provide more disk space to openSUSE®, free the needed space starting from the bottom toward the top of the list (starting from the last partition of a hard disk toward the first). For example, if you have three partitions, you cannot use the second exclusively for openSUSE and retain the third and first for other operating systems.

2.1.1 Partition Types

Every hard disk has a partition table with space for four entries. An entry in the partition table can correspond to a primary partition or an extended partition. Only one extended partition entry is allowed, however.

A primary partition simply consists of a continuous range of cylinders (physical disk areas) assigned to a particular operating system. With primary partitions only, you would be limited to four partitions per hard disk, because more do not fit in the partition table. This is why extended partitions are used. Extended partitions are also continuous ranges of disk cylinders, but an extended partition may itself be subdivided into *logical partitions*. Logical partitions do not require entries in the partition table. In other words, an extended partition is a container for logical partitions.

If you need more than four partitions, create an extended partition as the fourth partition or earlier. This extended partition should span the entire remaining free cylinder range. Then create multiple logical partitions within the extended partition. The maximum number of logical partitions is 15 on SCSI, SATA, and Firewire disks and 63 on (E)IDE disks. It does not matter which types of partitions are used for Linux. Primary and logical partitions both work fine.

2.1.2 Creating a Partition

To create a partition from scratch, proceed as follows:

1. Select *Create*. If several hard disks are connected, a selection dialog appears in which to select a hard disk for the new partition.

2. Specify the partition type (primary or extended). Create up to four primary partitions or up to three primary partitions and one extended partition. Within the extended partition, create several logical partitions (see Section 2.1.1, “Partition Types” (page 42)).

3. Select the file system to use and a mount point. YaST suggests a mount point for each partition created. Refer to Chapter 16, *File Systems in Linux* (page 241) for details on the various file systems.

4. Specify additional file system options if your setup requires them. This is necessary, for example, if you need persistent device names. For details on the available options, refer to Section 2.1.3, “Editing a Partition” (page 44).

5. Click *OK* > *Apply* to apply your partitioning setup and leave the partitioning module.

   If you created the partition during installation, you are returned to the installation overview screen.

2.1.3 Editing a Partition

When you create a new partition or modify an existing partition, set various parameters. For new partitions, suitable parameters are set by YaST and usually do not require any modification. To edit your partition setup manually, proceed as follows:

1 Select the partition.

2 Click *Edit* to edit the partition and set the parameters:

File System ID
: Even if you do not want to format the partition at this stage, assign it a file system ID to ensure that the partition is registered correctly. Possible values include *Linux*, *Linux swap*, *Linux LVM*, and *Linux RAID*. For LVM and RAID details, refer to Section 2.2, “LVM Configuration” (page 49) and Section 2.3, “Soft RAID Configuration” (page 55).

File System
: Change the file system or format the partition here. Changing the file system or reformatting partitions irreversibly deletes all data from the partition . For details on the various file systems, refer to Chapter 16, *File Systems in Linux* (page 241).

  Swap is a special format that allows the partition to be used as virtual memory. Create a swap partition of at least 256 MB. However, if you use up your swap space, consider adding more memory to your system instead of adding more swap space.

  Ext3 is the default file system for the Linux partitions. ReiserFS, JFS, and Ext3 are journaling file systems. These file systems are able to restore the system very quickly after a system crash, because write processes are logged during the operation. Furthermore, ReiserFS is very fast in handling lots of small files. Ext2 is not a journaling file system. However, it is rock solid and good for smaller partitions, because it does not require much disk space for management.

Encrypt File System
: If you activate the encryption, all data is written to the hard disk in encrypted form. This increases the security of sensitive data, but slightly reduces the system speed, because the encryption takes some time. More information

about the encryption of file systems is provided in Chapter 40, *Encrypting Partitions and Files* (page 643).

Fstab Options

Specify various parameters contained in the global file system administration file (`/etc/fstab`). The default settings should suffice for most setups. You can, for example, change the file system identification from the device name to a volume label. In the volume label, use all characters except / and space.

To get persistent devices names, use the mount option *Device ID* or *UUID*. In openSUSE, persistent device names are enabled by default.

If you intend to use quota on the file system, use the mount option *Enable Quota Support*. This must be done before you can define quotas for users in the YaST *User Management* module. For further information on how to configure user quota, refer to Section “Managing Quotas” (Chapter 5, *Managing Users with YaST*, ↑Start-Up).

Mount Point

Specify the directory at which the partition should be mounted in the file system tree. Select from various YaST proposals or enter any other name.

3 Select *OK* > *Apply* to activate the partition.

2.1.4 Expert Options

Expert opens a menu containing the following commands:

Reread Partition Table

Rereads the partitioning from disk. For example, you need this after manual partitioning in the text console.

Delete Partition Table and Disk Label

This completely overwrites the old partition table. For example, this can be helpful if you have problems with unconventional disk labels. Using this method, all data on the hard disk is lost.

Call iSCSI configuration

To access SCSI over IP block devices, you first have to configure iSCSI. This results in additionally available devices in the main partition list.

2.1.5 More Partitioning Tips

The following section comprises a few hints and tips on partitioning that should help you in taking the right decisions while setting up your system.

TIP: Cylinder Numbers

Note, that different partitioning tools may start counting the cylinders of a partition with `0` or with `1`. When calculating the number of cylinders, you should always use the difference between the last and the first cylinder number and add one.

Foreign Partitions and `fstab`

If the partitioning is performed by YaST and other partitions are detected in the system, these partitions are also added to the `/etc/fstab` file to enable easy access to this data. This file contains all partitions in the system with their properties, such as the file system, mount point, and user permissions.

Example 2.1 */etc/fstab: Partition Data*

```
/dev/disk/by-id/scsi-SATA_SAMSUNG_SV4002H0413J1FR917614-part1 /data1 auto noauto,user 0 0
/dev/disk/by-id/scsi-SATA_SAMSUNG_SV4002H0413J1FR917614-part5 /data2 auto noauto,user 0 0
/dev/disk/by-id/scsi-SATA_SAMSUNG_SV4002H0413J1FR917614-part6 /data3 auto noauto,user 0 0
```

The partitions, regardless of whether they are Linux or FAT partitions, are specified with the options `noauto` and `user`. This allows any user to mount or unmount these partitions as needed. For security reasons, YaST does not automatically enter the `exec` option here, which is needed for executing programs from the location. However, to run programs from there, you can enter this option manually. This measure is necessary if you encounter system messages such as “bad interpreter” or “Permission denied”.

Using `swap`

Swap is used to extend the physically available memory. This way it is possible to use more memory than physical ram available. The memory management system of kernels before 2.4.10 needed swap as a safety measure. In those times, if you did not have twice the size of your ram in swap, the performance of the system suffered. This does not hold true anymore as these limitations no longer exist.

When the kernel runs out of memory, it swaps out pages of memory that are not used frequently. Therefore, the running applications have more memory available and even their caching works more smoothly.

If an application tries to allocate as much memory as it can possibly get, there are some problems with swap. There are three major cases to look at:

System with no swap

> The application gets all memory that can be freed by any means. All caches are freed, and thus all other applications are slowed down. After several minutes, the out of memory killer mechanism of the kernel will become active and kill the process.

System with medium sized swap (128 MB–256 MB)

> At first, the system is slowed down like a system without swap. After all physical ram has been used up, swap space is used as well. At this point, the system becomes very slow and it becomes impossible to run commands from remote. Depending on the speed of the hard disks that run the swap space, the system stays in this condition for about 10 to 15 minutes until the out of memory killer of the kernel resolves the issue.

System with lots of swap (several GB)

> You better do not have an application that is running wild and swapping frantically, in this case. If you do have this problem, the system will need many hours to recover. In the process, it is likely that other processes get timeouts and faults, leaving the system in an undefined state, even if the faulty process is killed. In this case, you better just reboot the machine hard and try to get it running again. Lots of swap is only useful if you have an application that relies on this feature. Such applications (like databases or graphics manipulation programs) have an option to directly use hard disk space for their needs. It is advisable to use this option instead of using lots of swap space.

If your system does not run wild, but needs more swap after some time, it is possible to extend the swap space online. If you prepared a partition for swap space, just add this partition with YaST. If you do not have a partition available, you may also just use a swap file to extend the swap. Swap files are generally slower than partitions, but compared to physical ram, both are extremely slow and the actual speed difference is not as important as one would think in the first place.

Procedure 2.1 *Adding a Swap File Manually*

To add a swap file in the running system, proceed as follows:

1 Create an empty file in your system. For example, if you want to add a swap file with 128 MB swap at `/var/lib/swap/swapfile`, use the commands:

```
mkdir -p /var/lib/swap
dd if=/dev/zero of=/var/lib/swap/swapfile bs=1M count=128
```

2 Initialize this swap file with the command

```
mkswap /var/lib/swap/swapfile
```

3 Activate the swap with the command

```
swapon /var/lib/swap/swapfile
```

To disable this swap file, you use the command

```
swapoff /var/lib/swap/swapfile
```

4 Check the current available swap spaces with the command

```
cat /proc/swaps
```

Note, that at this point this is only temporary swap space. After the next reboot, it is not used anymore.

5 To enable this swap file permanently, add the following line to `/etc/fstab`:

```
/var/lib/swap/swapfile swap swap defaults 0 0
```

2.1.6 Partitioning and LVM

From the expert partitioner, access the LVM configuration with *LVM* (see Section 2.2, "LVM Configuration" (page 49)). However, if a working LVM configuration already exists on your system, it is automatically activated as soon as you enter the LVM configuration for the first time in a session. In this case, any disks containing a partition belonging to an activated volume group cannot be repartitioned because the Linux kernel cannot reread the modified partition table of a hard disk when any partition on this disk is in use. However, if you already have a functioning LVM configuration on your system, physical repartitioning should not be necessary. Instead, change the configuration of the logical volumes.

At the beginning of the physical volumes (PVs), information about the volume is written to the partition. To reuse such a partition for other non-LVM purposes, it is advisable to delete the beginning of this volume. For example, in the VG `system` and PV `/dev/sda2`, do this with the command `dd if=/dev/zero of=/dev/sda2 bs=512 count=1`.

WARNING: File System for Booting

The file system used for booting (the root file system or `/boot`) must not be stored on an LVM logical volume. Instead, store it on a normal physical partition.

2.2 LVM Configuration

This section briefly describes the principles behind LVM and its basic features that make it useful under many circumstances. In Section 2.2.2, "LVM Configuration with YaST" (page 51), learn how to set up LVM with YaST.

WARNING

Using LVM might be associated with increased risk, such as data loss. Risks also include application crashes, power failures, and faulty commands. Save your data before implementing LVM or reconfiguring volumes. Never work without a backup.

2.2.1 The Logical Volume Manager

The Logical Volume Manager (LVM) enables flexible distribution of hard disk space over several file systems. It was developed because sometimes the need to change the segmentation of hard disk space arises only after the initial partitioning during installation has already been done. Because it is difficult to modify partitions on a running system, LVM provides a virtual pool (volume group, VG for short) of memory space from which logical volumes (LVs) can be created as needed. The operating system accesses these LVs instead of the physical partitions. Volume groups can span more than only one disk so that several disks or parts of them may constitute one single VG. This way, LVM provides a kind of abstraction from the physical disk space that allows its segmentation to be changed in a much easier and safer way than physical repartitioning does. Background information regarding physical partitioning can be found in Section 2.1.1, "Partition Types" (page 42) and Section 2.1, "Using the YaST Partitioner" (page 41).

Figure 2.2 *Physical Partitioning versus LVM*

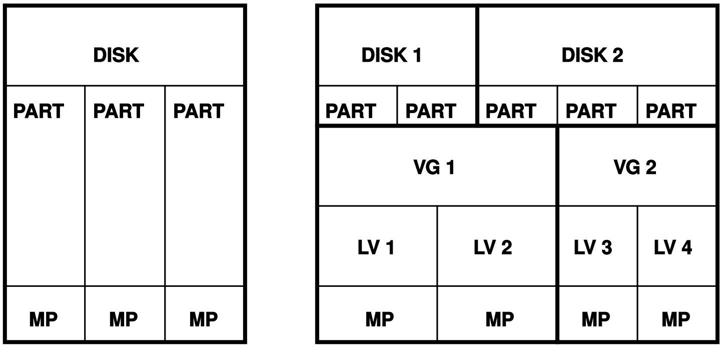


Figure 2.2, "Physical Partitioning versus LVM" (page 50) compares physical partitioning (left) with LVM segmentation (right). On the left side, one single disk has been divided into three physical partitions (PART), each with a mount point (MP) assigned so that the operating system can access them. On the right side, two disks have been divided into two and three physical partitions each. Two LVM volume groups (VG 1 and VG 2) have been defined. VG 1 contains two partitions from DISK 1 and one from DISK 2. VG 2 contains the remaining two partitions from DISK 2. In LVM, the physical disk partitions that are incorporated in a volume group are called physical volumes (PVs). Within the volume groups, four logical volumes (LV 1 through LV 4) have been defined, which can be used by the operating system via the associated mount points. The border

between different logical volumes need not be aligned with any partition border. See the border between LV 1 and LV 2 in this example.

LVM features:

- Several hard disks or partitions can be combined in a large logical volume.
- Provided the configuration is suitable, an LV (such as `/usr`) can be enlarged when the free space is exhausted.
- Using LVM, it is possible to add hard disks or LVs in a running system. However, this requires hot-swappable hardware that is capable of such actions.
- It is possible to activate a "striping mode" that distributes the data stream of a logical volume over several physical volumes. If these physical volumes reside on different disks, this can improve the reading and writing performance just like RAID 0.
- The snapshot feature enables consistent backups (especially for servers) in the running system.

With these features, using LVM already makes sense for heavily used home PCs or small servers. If you have a growing data stock, as in the case of databases, music archives, or user directories, LVM is just the right thing for you. This would allow file systems that are larger than the physical hard disk. Another advantage of LVM is that up to 256 LVs can be added. However, keep in mind that working with LVM is different from working with conventional partitions. Instructions and further information about configuring LVM is available in the official LVM HOWTO at `http://tldp.org/HOWTO/LVM-HOWTO/`.

Starting from kernel version 2.6, LVM version 2 is available, which is downward-compatible with the previous LVM and enables the continued management of old volume groups. When creating new volume groups, decide whether to use the new format or the downward-compatible version. LVM 2 does not require any kernel patches. It makes use of the device mapper integrated in kernel 2.6. This kernel only supports LVM version 2. Therefore, when talking about LVM, this section always refers to LVM version 2.

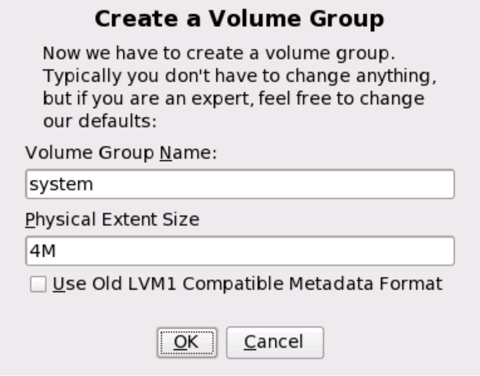
2.2.2 LVM Configuration with YaST

The YaST LVM configuration can be reached from the YaST Expert Partitioner (see Section 2.1, "Using the YaST Partitioner" (page 41)). This partitioning tool enables you to edit and delete existing partitions and create new ones that should be used with LVM. There, create an LVM partition by first clicking *Create* > *Do not format* then selecting *0x8E Linux LVM* as the partition identifier. After creating all the partitions to use with LVM, click *LVM* to start the LVM configuration.

Creating Volume Groups

If no volume group exists on your system yet, you are prompted to add one (see Figure 2.3, "Creating a Volume Group" (page 52)). It is possible to create additional groups with *Add group*, but usually one single volume group is sufficient. `system` is suggested as a name for the volume group in which the openSUSE® system files are located. The physical extent size defines the size of a physical block in the volume group. All the disk space in a volume group is handled in chunks of this size. This value is normally set to 4 MB and allows for a maximum size of 256 GB for physical and logical volumes. The physical extent size should only be increased, for example, to 8, 16, or 32 MB, if you need logical volumes larger than 256 GB.

Figure 2.3 *Creating a Volume Group*



Create a Volume Group

Now we have to create a volume group.
Typically you don't have to change anything,
but if you are an expert, feel free to change
our defaults:

Volume Group Name: system

Physical Extent Size: 4M

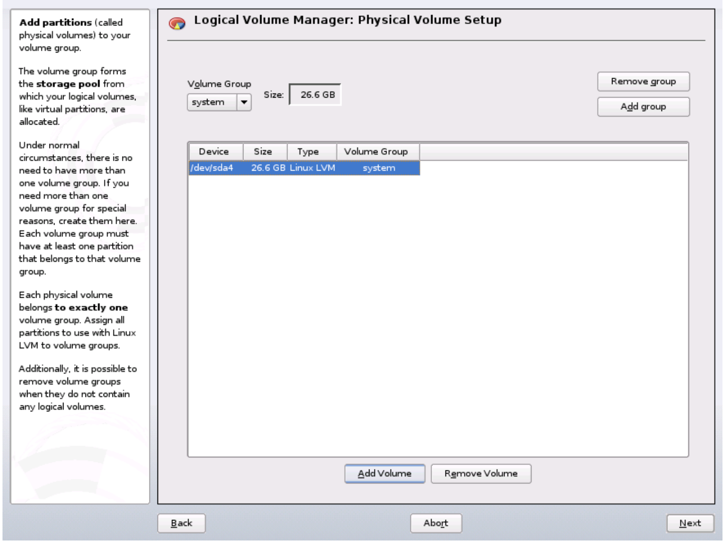
□ Use Old LVM1 Compatible Metadata Format

Configuring Physical Volumes

Once a volume group has been created, the following dialog lists all partitions with either the "Linux LVM" or "Linux native" type. No swap or DOS partitions are shown. If a partition is already assigned to a volume group, the name of the volume group is shown in the list. Unassigned partitions are indicated with "--".

If there are several volume groups, set the current volume group in the selection box to the upper left. The buttons in the upper right enable creation of additional volume groups and deletion of existing volume groups. Only volume groups that do not have any partitions assigned can be deleted. All partitions that are assigned to a volume group are also referred to as a physical volumes (PV).

Figure 2.4 *Physical Volume Setup*



Add partitions (called physical volumes) to your volume group.

The volume group forms the **storage pool** from which your logical volumes, like virtual partitions, are allocated.

Under normal circumstances, there is no need to have more than one volume group. If you need more than one volume group for special reasons, create them here. Each volume group must have at least one partition that belongs to that volume group.

Each physical volume belongs **to exactly one** volume group. Assign all partitions to use with Linux LVM to volume groups.

Additionally, it is possible to remove volume groups when they do not contain any logical volumes.

Logical Volume Manager: Physical Volume Setup

Volume Group: system

Size: 26.6 GB

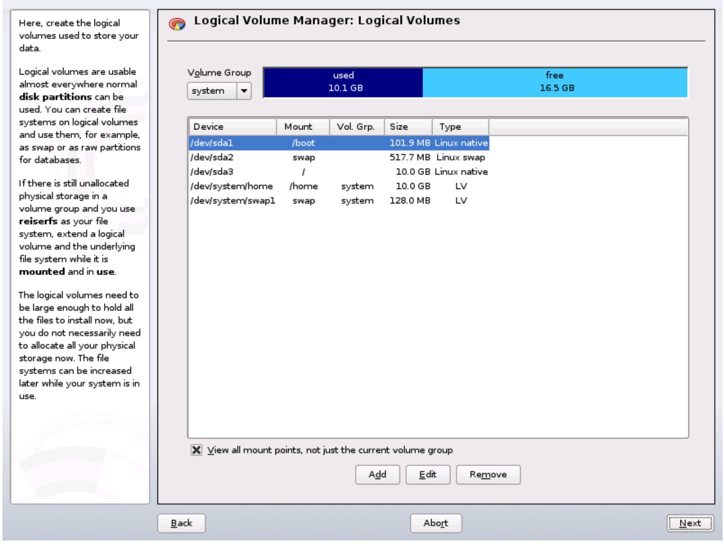
Device	Size	Type	Volume Group
/dev/sda4	26.6 GB	Linux LVM	system

To add a previously unassigned partition to the selected volume group, first click the partition then *Add Volume*. At this point, the name of the volume group is entered next to the selected partition. Assign all partitions reserved for LVM to a volume group. Otherwise, the space on the partition remains unused. Before exiting the dialog, every volume group must be assigned at least one physical volume. After assigning all physical volumes, click *Next* to proceed to the configuration of logical volumes.

Configuring Logical Volumes

After the volume group has been filled with physical volumes, define the logical volumes the operating system should use in the next dialog. Set the current volume group in a selection box to the upper left. Next to it, the free space in the current volume group is shown. The list below contains all logical volumes in that volume group. All normal Linux partitions to which a mount point is assigned, all swap partitions, and all already existing logical volumes are listed here. *Add*, *Edit*, and *Remove* logical volumes as needed until all space in the volume group has been exhausted. Assign at least one logical volume to each volume group.

Figure 2.5 *Logical Volume Management*



Here, create the logical volumes used to store your data.

Logical volumes are usable almost everywhere normal **disk partitions** can be used. You can create file systems on logical volumes and use them, for example, as swap or as raw partitions for databases.

If there is still unallocated physical storage in a volume group and you use **reiserfs** as your file system, extend a logical volume and the underlying file system while it is **mounted** and in **use**.

The logical volumes need to be large enough to hold all the files to install now, but you do not necessarily need to allocate all your physical storage now. The file systems can be increased later while your system is in use.

Logical Volume Manager: Logical Volumes

Volume Group: system

Device	Mount	Vol. Grp.	Size	Type
/dev/sda1	/boot		101.9 MB	Linux native
/dev/sda2	swap		517.7 MB	Linux swap
/dev/sda3	/		10.0 GB	Linux native
/dev/system/home	/home	system	10.0 GB	LV
/dev/system/swap1	swap	system	128.0 MB	LV

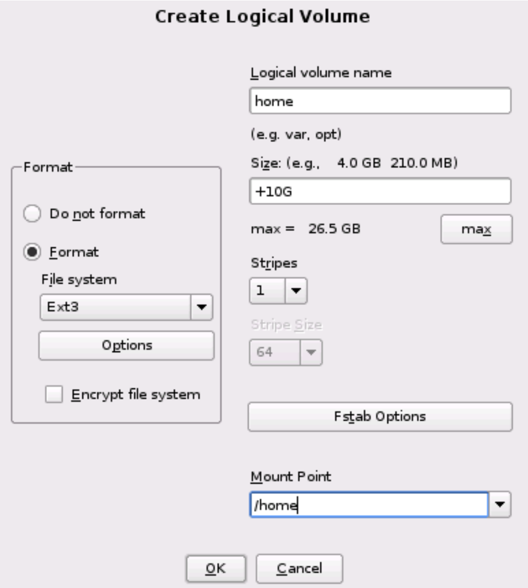
☒ View all mount points, not just the current volume group

To create a new logical volume, click *Add* and fill out the pop-up that opens. As for partitioning, enter the size, file system, and mount point. Normally, a file system, such as reiserfs or ext2, is created on a logical volume and is then designated a mount point. The files stored on this logical volume can be found at this mount point on the installed system. Additionally it is possible to distribute the data stream in the logical volume among several physical volumes (striping). If these physical volumes reside on different hard disks, this generally results in a better reading and writing performance (like RAID 0). However, a striping LV with `n` stripes can only be created correctly if the hard disk space required by the LV can be distributed evenly to `n` physical volumes. If, for example, only two physical volumes are available, a logical volume with three stripes is impossible.

WARNING: Striping

YaST has no chance at this point to verify the correctness of your entries concerning striping. Any mistake made here is apparent only later when the LVM is implemented on disk.

Figure 2.6 Creating Logical Volumes



Create Logical Volume

Format

○ Do not format

● Format

File system: Ext3

□ Encrypt file system

Logical volume name: home

(e.g. var, opt)

Size: (e.g., 4.0 GB 210.0 MB): +10G

max = 26.5 GB

Stripes: 1

Stripe Size: 64

Mount Point: /home

If you have already configured LVM on your system, the existing logical volumes can be entered now. Before continuing, assign appropriate mount points to these logical volumes too. With *Next*, return to the YaST Expert Partitioner and finish your work there.

Direct LVM Management

If you already have configured LVM and only want to change something, there is an alternative way to do that. In the YaST Control Center, select *System* > *LVM*. Basically this dialog allows the same actions as described above with the exception of physical partitioning. It shows the existing physical volumes and logical volumes in two lists and you can manage your LVM system using the methods already described.

2.3 Soft RAID Configuration

The purpose of RAID (redundant array of independent disks) is to combine several hard disk partitions into one large *virtual* hard disk to optimize performance, data security, or both. Most RAID controllers use the SCSI protocol because it can address a larger number of hard disks in a more effective way than the IDE protocol and is more suitable for parallel processing of commands. There are some RAID controllers that support IDE or SATA hard disks. Soft RAID provides the advantages of RAID systems without the additional cost of hardware RAID controllers. However, this requires some CPU time and has memory requirements that make it unsuitable for real high performance computers.

2.3.1 RAID Levels

openSUSE® offers the option of combining several hard disks into one soft RAID system with the help. RAID implies several strategies for combining several hard disks in a RAID system, each with different goals, advantages, and characteristics. These variations are commonly known as *RAID levels*.

Common RAID levels are:

RAID 0

This level improves the performance of your data access by spreading out blocks of each file across multiple disk drives. Actually, this is not really a RAID, because it does not provide data backup, but the name *RAID 0* for this type of system has become the norm. With RAID 0, two or more hard disks are pooled together. The performance is very good, but the RAID system is destroyed and your data lost if even one hard disk fails.

RAID 1

This level provides adequate security for your data, because the data is copied to another hard disk 1:1. This is known as *hard disk mirroring*. If a disk is destroyed, a copy of its contents is available on another one. All of them except one could be damaged without endangering your data. However, if damage is not detected, it also may happen that damaged data is mirrored to the correct disk and data corruption happens that way. The writing performance suffers a little in the copying process compared to when using single disk access (10 to 20 % slower), but read access is significantly faster in comparison to any one of the normal physical hard disks, because the data is duplicated so can be parallel scanned. Generally it can be said that Level 1 provides nearly twice the read transaction rate of single disks and almost the same write transaction rate as single disks.

RAID 2 and RAID 3

These are not typical RAID implementations. Level 2 stripes data at the bit level rather than the block level. Level 3 provides byte-level striping with a dedicated parity disk and cannot service simultaneous multiple requests. Both levels are only rarely used.

RAID 4

Level 4 provides block-level striping just like Level 0 combined with a dedicated parity disk. In the case of a data disk failure, the parity data is used to create a replacement disk. However, the parity disk may create a bottleneck for write access. Nevertheless, Level 4 is sometimes used.

RAID 5

RAID 5 is an optimized compromise between Level 0 and Level 1 in terms of performance and redundancy. The hard disk space equals the number of disks used minus one. The data is distributed over the hard disks as with RAID 0. *Parity blocks*, created on one of the partitions, are there for security reasons. They are linked to each other with XOR, enabling the contents to be reconstructed by the corresponding parity block in case of system failure. With RAID 5, no more than one hard disk can fail at the same time. If one hard disk fails, it must be replaced as soon as possible to avoid the risk of losing data.

RAID 6

To further increase the reliability of the RAID system, it is possible to use RAID 6. In this level, even if two disks fail, the array still can be reconstructed. With RAID 6, at least 4 hard disks are needed to run the array. Note, that when running as software raid, this configuration needs a considerable amount of CPU time and memory.

Other RAID Levels

Several other RAID levels have been developed (RAIDn, RAID 10, RAID 0+1, RAID 30, RAID 50, etc.), some of them being proprietary implementations created by hardware vendors. These levels are not very widespread, so are not explained here.

2.3.2 Soft RAID Configuration with YaST

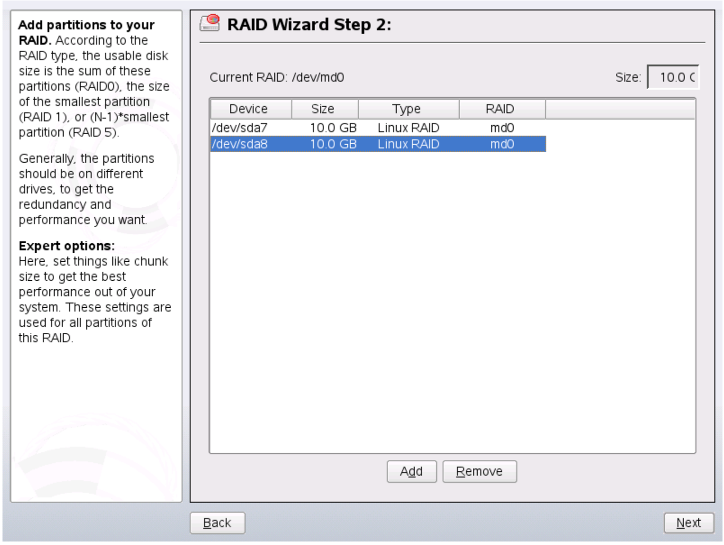
The YaST soft RAID configuration can be reached from the YaST Expert Partitioner, described in Section 2.1, “Using the YaST Partitioner” (page 41). This partitioning tool enables you to edit and delete existing partitions and create new ones that should be used with soft RAID. There, create RAID partitions by first clicking *Create* > *Do not format* then selecting *0xFD Linux RAID* as the partition identifier. For RAID 0 and RAID 1, at least two partitions are needed—for RAID 1, usually exactly two and no more. If RAID 5 is used, at least three partitions are required. It is recommended to take only partitions of the same size. The RAID partitions should be stored on different hard disks to decrease the risk of losing data if one is defective (RAID 1 and 5) and to optimize the performance of RAID 0. After creating all the partitions to use with RAID, click *RAID* > *Create RAID* to start the RAID configuration.

TIP

Starting with openSUSE 10.2, the system detects the settings of pseudo RAID adapters found on many mainboards. These are used to setup the software RAID without additional interaction.

In the next dialog, choose between RAID levels 0, 1, and 5 (see Section 2.3.1, "RAID Levels" (page 56) for details). After *Next* is clicked, the following dialog lists all partitions with either the "Linux RAID" or "Linux native" type (see Figure 2.7, "RAID Partitions" (page 58)). No swap or DOS partitions are shown. If a partition is already assigned to a RAID volume, the name of the RAID device (for example, `/dev/md0`) is shown in the list. Unassigned partitions are indicated with "--".

Figure 2.7 *RAID Partitions*



Add partitions to your RAID. According to the RAID type, the usable disk size is the sum of these partitions (RAID0), the size of the smallest partition (RAID 1), or (N-1)*smallest partition (RAID 5).

Generally, the partitions should be on different drives, to get the redundancy and performance you want.

Expert options: Here, set things like chunk size to get the best performance out of your system. These settings are used for all partitions of this RAID.

RAID Wizard Step 2:

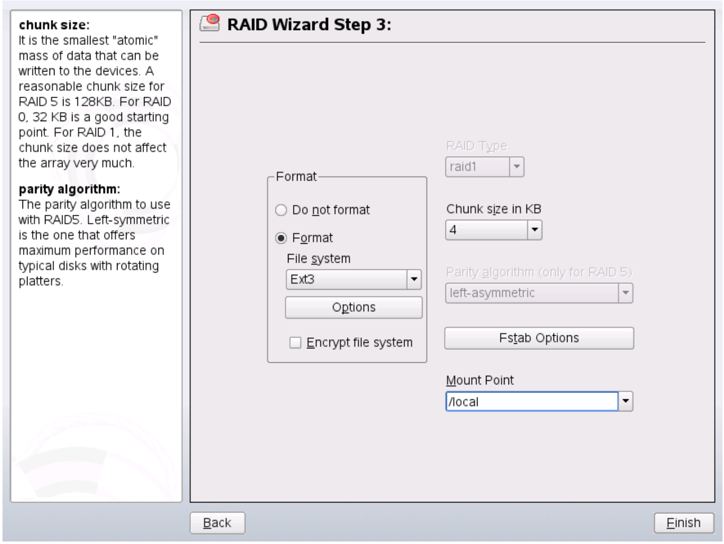
Current RAID: /dev/md0

Size: 10.0 C

Device	Size	Type	RAID
/dev/sda7	10.0 GB	Linux RAID	md0
/dev/sda8	10.0 GB	Linux RAID	md0

To add a previously unassigned partition to the selected RAID volume, first click the partition then *Add*. At this point, the name of the RAID device is entered next to the selected partition. Assign all partitions reserved for RAID. Otherwise, the space on the partition remains unused. After assigning all partitions, click *Next* to proceed to the settings dialog where you can fine-tune the performance (see Figure 2.8, "File System Settings" (page 59)).

***Figure 2.8** File System Settings*



chunk size:
It is the smallest "atomic" mass of data that can be written to the devices. A reasonable chunk size for RAID 5 is 128KB. For RAID 0, 32 KB is a good starting point. For RAID 1, the chunk size does not affect the array very much.

parity algorithm:
The parity algorithm to use with RAID5. Left-symmetric is the one that offers maximum performance on typical disks with rotating platters.

RAID Wizard Step 3:

Format
○ Do not format
● Format
File system: Ext3
□ Encrypt file system

RAID Type: raid1
Chunk size in KB: 4
Parity algorithm (only for RAID 5): left-asymmetric
Mount Point: /local

As with conventional partitioning, set the file system to use as well as encryption and the mount point for the RAID volume. Checking *Persistent Superblock* ensures that the RAID partitions are recognized as such when booting. After completing the configuration with *Finish*, see the `/dev/md0` device and others indicated with *RAID* in the expert partitioner.

2.3.3 Troubleshooting

Check the file `/proc/mdstats` to find out whether a RAID partition has been destroyed. In the event of a system failure, shut down your Linux system and replace the defective hard disk with a new one partitioned the same way. Then restart your system and enter the command `mdadm /dev/mdX --add /dev/sdX`. Replace 'X' with your particular device identifiers. This integrates the hard disk automatically into the RAID system and fully reconstructs it.

2.3.4 For More Information

Configuration instructions and more details for soft RAID can be found in the HOWTOs at `http://en.tldp.org/HOWTO/Software-RAID-HOWTO.html`.

Linux RAID mailing lists are also available, such as `http://marc.theaimsgroup.com/?l=linux-raid`.

Part II. Administration

Online Update 3

openSUSE offers a continuous stream of software security updates for your product. By default openSUSE Updater is used to keep your system up-to-date. Refer to Section “Keeping the System Up-to-date” (Chapter 3, *Installing or Removing Software*, ↑Start-Up) for further information on openSUSE Updater. This chapter covers alternative graphical tools and command line utilities for updating software packages.

The current patches for openSUSE® are available from an update software repository. If you have registered your product during the installation, an update repository is already configured. If you have not registered openSUSE, you can do so by running *Software* > *Online Update Configuration* in YaST. Alternatively, you can manually add an update repository from a source you trust with each update tool. Please refer to the respective application described below for instructions.

openSUSE provides updates with different relevance levels. `Security` updates fix severe security hazards and should definitely be installed. `Recommended` updates fix issues that could compromise your computer, whereas `Optional` updates fix non-security relevant issues or provide enhancements.

3.1 YaST Online Update

To install updates and improvements with YaST, run *Software* > *Online Update*. All new patches (except the optional ones) that are currently available for your system are already marked for installation. Clicking *Accept* automatically installs these patches. After the installation has completed, confirm with *Finish*. Your system is now up-to-date.

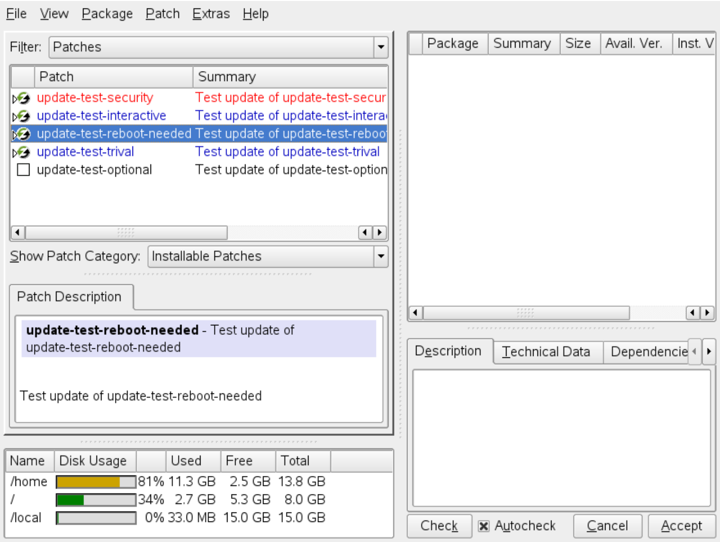
TIP

YaST Online Update has been integrated into the YaST software management module. This ensures that the newest version of a package is always installed. It is no longer necessary to run an online update after installing new packages.

3.1.1 Installing Patches Manually

The *Online Update* window consists of five sections. The list of all patches available is on the left. Find the description of the selected patch displayed below the list of patches. The disk usage is displayed at the bottom of the left column. The right column lists the packages included in the selected patch (a patch can consist of several packages) and, below, a detailed description of the selected package.

Figure 3.1 *YaST Online Update*



Filter: Patches

	Patch	Summary
	update-test-security	Test update of update-test-secur
	update-test-interactive	Test update of update-test-intera
	update-test-reboot-needed	Test update of update-test-reboo
	update-test-trival	Test update of update-test-trival
□	update-test-optional	Test update of update-test-option

Show Patch Category: Installable Patches

Patch Description

update-test-reboot-needed - Test update of update-test-reboot-needed

Test update of update-test-reboot-needed

Name	Disk Usage	Used	Free	Total
/home	81%	11.3 GB	2.5 GB	13.8 GB
/	34%	2.7 GB	5.3 GB	8.0 GB
/local	0%	33.0 MB	15.0 GB	15.0 GB

The patch display lists patches available for openSUSE. The patches are sorted by relevance, security-wise. The color of the patch name as well as a pop-up window under the mouse cursor indicate the security status of the patch: `Security` (red), `Recommended` (blue), or `Optional` (black). There are three different views on patches. Use *Show Patch Category* to toggle the views:

Installable Patches (default view)
Patches currently not installed that apply to packages installed on your system.

Installable and Installed Patches
All patches that apply to packages installed on your system.

All Patches
All patches available for openSUSE.

A list entry consists of a symbol and the patch name. For a list of possible symbols, press Shift + F1. Actions required by `Security` and `Recommended` patches are automatically preset. These actions are *Autoinstall*, *Autoupdate*, or *Autodelete*. Actions for `Optional` patches are not preset—right-click on a patch and choose an action from the list.

Most patches include updates for several packages. If you want to change actions for single packages, right-click on a package in the package window and choose an action. Once you have marked all patches and packages as desired, proceed with *Accept*.

3.1.2 Automatic Online Update

YaST also offers the possibility to set up an automatic update. Open *Software* > *Automatic Online Update* for the configuration screen. You can either configure a *Daily* or a *Weekly* update. Some patches, such as kernel updates, require user interaction, which would cause the automatic update procedure to stop. Therefore you should check *Skip Interactive Patches*, if you want the update procedure to proceed fully automatically. Having done so, you should run a manual *Online Update* from time to time in order to install patches that require interaction.

3.1.3 Adding an Update Repository

To add or remove repositoreis, use the *Software* > *Software Repositories* module, described in Section “Adding Software Repositories” (Chapter 3, *Installing or Removing Software*, ↑Start-Up).

3.2 Update from the Command Line with zypper

openSUSE comes with a command line tool for installing and updating packages—zypper. It is especially useful to accomplish remote software management tasks or to manage software from shell scripts.

3.2.1 Installing and Removing Software with Zypper

To install a package from registered installation repositories, use `zypper in` *`package_name`*. To remove an installed package, use `zypper rm` *`package_name`*.

By default, zypper asks for confirmation before installing or removing a selected package. Override this behavior using the `--non-interactive` option. Note that this option must be given before the actual mode (install, remove, and update) as in `zypper --non-interactive in` *`package_name`*. This option allows the use of zypper in scripts and cron jobs.

3.2.2 Updating Software with Zypper

There are two different ways to update software using zypper. To integrate all officially released patches into your system, just run the `zypper up` command. In this case, all patches that are available in your installation repositories are checked for relevance, and installed if necessary.

If an installation source just has new packages, but does not provide a patch for this package, `zypper up` does not show any effect. To update one of these packages, you must reinstall the respective package with the `zypper in` *`package_name`* command.

3.2.3 Managing Installation Repositories

All installation or update commands of zypper rely on a list of installation repositories known to zypper. To list all installation repositories known to the system, use the command `zypper sl`.

If you want to remove an installation repository, first list all the repositories and then use the command `zypper sd` *`#number`*. For example, the installation repository you want to remove could be represented by the following line (line break added for layout reasons):

```
2 | No | Yes | YaST | openSUSE FACTORY Extra 10.3 | \
http://download.opensuse.org/distribution/SL-Factory-non-oss/inst-source-extra/
```

In this case run the command `zypper sd 2` to remove this installation repository from the list.

To add an installation repository, you can use the command `zypper sa` *`installation_repository`*. Information about additional installation sources is provided at http://en.opensuse.org/Installation_Sources.

3.2.4 Using the Zypper Shell

Sometimes, several different zypper commands must be run in a sequence. To prevent zypper from rereading all the databases for each zypper command, it is possible to run zypper in shell mode: `zypper sh`.

When the shell is running, just issue the zypper commands with the respective parameters:

```
zypper sh
zypper> in zsh
...
zypper> exit
```

Using the zypper shell is usually faster, because all the relevant data stays in memory.

Zypper supports the readline library. This means that you can use all the command line editing functions in the zypper shell that are also available in the Bash shell. Zypper maintains its command history in file `~/.zypper_history`.

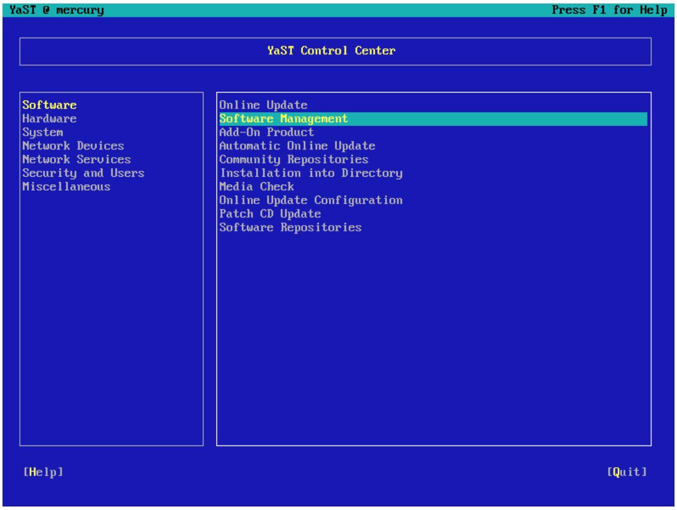
3.2.5 For More Information

For more information about updating from the command line, enter `zypper --help` or see the `zypper(8)` man page. For examples and detailed information, visit `http://en.opensuse.org/Using_zypper`.

4 YaST in Text Mode

This section is intended for system administrators and experts who do not run an X server on their systems and depend on the text-based installation tool. It provides basic information about starting and operating YaST in text mode.

Figure 4.1 Main Window of YaST in Text Mode



```
YaST @ mercury                                                Press F1 for Help

                              YaST Control Center

Software                 Online Update
Hardware                 Software Management
System                   Add-On Product
Network Devices          Automatic Online Update
Network Services         Community Repositories
Security and Users       Installation into Directory
Miscellaneous            Media Check
                         Online Update Configuration
                         Patch CD Update
                         Software Repositories

[Help]                                                                   [Quit]
```

When YaST is started in text mode, the YaST Control Center appears first. See Figure 4.1 . The main window consists of three areas. The left frame, which is surrounded by a thick white border, features the categories to which the various modules belong. The active category is indicated by a colored background. The right frame, which is surrounded by a thin white border, provides an overview of the modules available in the active category. The bottom frame contains the buttons for *Help* and *Quit*.

When the YaST Control Center is started, the category *Software* is selected automatically. Use ↓ and ↑ to change the category. To start a module from the selected category, press →. The module selection now appears with a thick border. Use ↓ and ↑ to select the desired module. Keep the arrow keys pressed to scroll through the list of available modules. When a module is selected, the module title appears with a colored background and a brief description is displayed in the bottom frame.

Press Enter to start the desired module. Various buttons or selection fields in the module contain a letter with a different color (yellow by default). Use Alt + yellow_letter to select a button directly instead of navigating there with Tab. Exit the YaST Control Center by pressing Alt + Q or by selecting *Quit* and pressing Enter.

4.1 Navigation in Modules

The following description of the control elements in the YaST modules assumes that all function keys and Alt key combinations work and are not assigned different global functions. Read Section 4.2, "Restriction of Key Combinations" (page 71) for information about possible exceptions.

Navigation among Buttons and Selection Lists

Use Tab and Alt + Tab or Shift + Tab to navigate among the buttons and the frames containing selection lists.

Navigation in Selection Lists

Use the arrow keys (↑ and ↓) to navigate among the individual elements in an active frame containing a selection list. If individual entries within a frame exceed its width, use Shift + → or Shift + ← to scroll horizontally to the right and left. Alternatively, use Ctrl + E or Ctrl + A. This combination can also be used if using → or ← would result in changing the active frame or the current selection list, as in the Control Center.

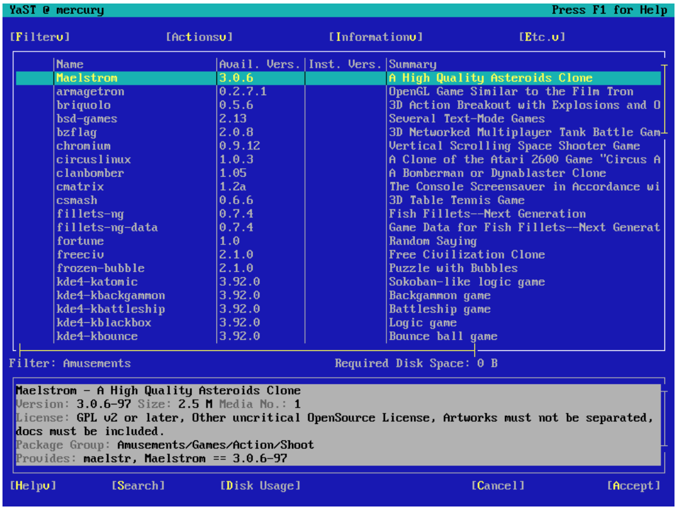
Buttons, Radio Buttons, and Check Boxes

To select buttons with empty square brackets (check boxes) or empty parentheses (radio buttons), press Space or Enter. Alternatively, radio buttons and check boxes can be selected directly with Alt + yellow_letter. In this case, you do not need to confirm with Enter. If you navigate to an item with Tab, press Enter to execute the selected action or activate the respective menu item.

Function Keys

The F keys (F1 to F12) enable quick access to the various buttons. Which function keys are actually mapped to which buttons depends on the active YaST module, because the different modules offer different buttons (Details, Info, Add, Delete, etc.). Use F10 for *OK*, *Next*, and *Finish*. Press F1 to access the YaST help, which shows the functions mapped to the individual F keys.

Figure 4.2 *The Software Installation Module*



Name	Avail. Vers.	Inst. Vers.	Summary
Maelstrom	3.0.6		A High Quality Asteroids Clone
armagetron	0.2.7.1		OpenGL Game Similar to the Film Tron
briquolo	0.5.6		3D Action Breakout with Explosions and O
bsd-games	2.13		Several Text-Mode Games
bzflag	2.0.8		3D Networked Multiplayer Tank Battle Gam
chromium	0.9.12		Vertical Scrolling Space Shooter Game
circuslinux	1.0.3		A Clone of the Atari 2600 Game "Circus A
clanbomber	1.05		A Bomberman or Dynablaster Clone
cmatrix	1.2a		The Console Screensaver in Accordance wi
csmash	0.6.6		3D Table Tennis Game
fillets-ng	0.7.4		Fish Fillets--Next Generation
fillets-ng-data	0.7.4		Game Data for Fish Fillets--Next Generat
fortune	1.0		Random Saying
freeciv	2.1.0		Free Civilization Clone
frozen-bubble	2.1.0		Puzzle with Bubbles
kde4-katomic	3.92.0		Sokoban-like logic game
kde4-kbackgammon	3.92.0		Backgammon game
kde4-kbattleship	3.92.0		Battleship game
kde4-kblackbox	3.92.0		Logic game
kde4-kbounce	3.92.0		Bounce ball game

Filter: Amusements Required Disk Space: 0 B

Maelstrom - A High Quality Asteroids Clone
Version: 3.0.6-97 Size: 2.5 M Media No.: 1
License: GPL v2 or later, Other uncritical OpenSource License, Artworks must not be separated, docs must be included.
Package Group: Amusements/Games/Action/Shoot
Provides: maelstr, Maelstrom == 3.0.6-97

4.2 Restriction of Key Combinations

If your window manager uses global Alt combinations, the Alt combinations in YaST might not work. Keys like Alt or Shift can also be occupied by the settings of the terminal.

Replacing Alt with Esc

Alt shortcuts can be executed with Esc instead of Alt. For example, Esc – H replaces Alt + H. (First press Esc, *then* press H.)

Backward and Forward Navigation with Ctrl + F and Ctrl + B

If the Alt and Shift combinations are occupied by the window manager or the terminal, use the combinations Ctrl + F (forward) and Ctrl + B (backward) instead.

Restriction of Function Keys

The F keys are also used for functions. Certain function keys might be occupied by the terminal and may not be available for YaST. However, the Alt key combinations and function keys should always be fully available on a pure text console.

4.3 YaST Command Line Options

Besides the text mode interface, YaST provides a pure command line interface. To get a list of YaST command line options, enter:

```
yast -h
```

4.3.1 Starting the Individual Modules

To save time, the individual YaST modules can be started directly. To start a module, enter:

```
yast <module_name>
```

View a list of all module names available on your system with `yast -l` or `yast --list`. Start the network module, for example, with `yast lan`.

4.3.2 Installing Packages from the Command Line

If you know a package name and the package is provided by any of your active installation repositories, you can use command line option `-i` to install the package:

```
yast -i <package_name>
```

or

```
yast --install <package_name>
```

package_name can be a single short package name, for example gvim which is installed with dependency checking or the full path to an rpm package, which is installed without dependency checking.

If you need a command-line based software management utility with functionality beyond what YaST provides, consider using zypper. This new utility uses the same software management library that is also the foundation for the YaST package manager. The basic usage of zypper is covered in Section 3.2, “Update from the Command Line with zypper” (page 66).

4.3.3 Command Line Parameters of the YaST Modules

To use YaST functionality in scripts, YaST provides command line support for individual modules. Not all modules have a command line support. To display the available options of a module, enter:

```
yast <module_name> help
```

If a module does not provide command line support, the module is started in text mode and the following message appears:

```
This YaST2 module does not support the command line interface.
```

Updating the System and System Changes

5

You can update an existing system without completely reinstalling it. There are two types of updates: *updating individual software packages* and *updating the entire system*.

5.1 Updating the System

Software tends to “grow” from version to version. Therefore, take a look at the available partition space with df before updating. If you suspect you are running short of disk space, secure your data before you update and repartition your system. There is no general rule of thumb regarding how much space each partition should have. Space requirements depend on your particular partitioning profile, the software selected, and the version numbers of the system.

5.1.1 Preparations

Before updating, copy the old configuration files to a separate medium, such as streamer, removable hard disk, USB stick, or ZIP drive, to secure the data. This primarily applies to files stored in /etc as well as some of the directories and files in /var and /opt. You may also want to write the user data in /home (the HOME directories) to a backup medium. Back up this data as root. Only root has read permission for all local files.

Before starting your update, make note of the root partition. The command df / lists the device name of the root partition. In Example 5.1, “List with df -h” (page 76), the root partition to write down is /dev/sda3 (mounted as /).

Example 5.1 *List with df -h*

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        74G   22G   53G  29% /
udev            252M  124K  252M   1% /dev
/dev/sda5       116G  5.8G  111G   5% /home
/dev/sda1        39G  1.6G   37G   4% /windows/C
/dev/sda2       4.6G  2.6G  2.1G  57% /windows/D
```

5.1.2 Possible Problems

If you update a default system from the previous version to this version, YaST works out necessary changes and performs them. Depending on your customizations, some steps or the entire update procedure may fail and you must resort to copying back your backup data. Here, we point out more issues to check before starting the system update.

Checking passwd and group in /etc

Before updating the system, make sure that `/etc/passwd` and `/etc/group` do not contain any syntax errors. For this purpose, start the verification utilities `pwck` and `grpck` as `root` and eliminate any reported errors.

PostgreSQL

Before updating PostgreSQL (`postgres`), dump the databases. See the manual page of `pg_dump`. This is only necessary if you actually used PostgreSQL prior to your update.

5.1.3 Updating with YaST

Following the preparation procedure outlined in Section 5.1.1, “Preparations” (page 75), you can now update your system:

1 Boot the system as for the installation, described in Section “System Start-Up for Installation” (Chapter 1, *Installation with YaST*, ↑Start-Up). In YaST, choose a language and select *Update* in the *Installation Mode* dialog. Do not select *New Installation*. Also add repositories to make sure to get all available software updated whenever possible. Find more information about installation repositories in Section “Add-On Products and Repositories” (Chapter 1, *Installation with YaST*, ↑Start-Up).

2 YaST determines whether there are multiple root partitions. If there is only one, continue with the next step. If there are several, select the right partition and confirm with *Next* (`/dev/sda3` was selected in the example in Section 5.1.1, “Preparations” (page 75)). YaST reads the old `fstab` on this partition to analyze and mount the file systems listed there.

3 Check the previously used repositories, if there are any. Enable all the repositories you still want to use and update third-party software from. Click the *Toggle Status* for every list item, if appropriate.

4 In case you added repositories during the update procedure as recommended above, you now can activate those you are actually interested in.

5 In the *Installation Settings* dialog, adjust the settings according to your requirements. Normally, you can leave the default settings untouched, but if you intend to enhance your system, check the packages offered in the *Software Selection* submenus or add support for additional languages.

You also have the possibility to make backups of various system components. Selecting backups slows down the update process. Use this option if you do not have a recent system backup.

6 Confirm the update by clicking *Start Update*.

Once the basic update installation is finished, test the Internet connection as offered by the YaST dialog. Finally, YaST updates the remaining software, offers the Novell Customer Center Configuration, and displays the release notes. Click *Finish* to write the YaST configuration.

5.1.4 Updating Individual Packages

Regardless of your overall updated environment, you can always update individual packages. From this point on, however, it is your responsibility to ensure that your system remains consistent. Update advice can be found at `http://www.novell.com/linux/download/updates/`.

Select components from the YaST package selection list according to your needs. If you select a package essential for the overall operation of the system, YaST issues a warning. Such packages should be updated only in the update mode. For example, many packages contain *shared libraries*. If you update these programs and applications in the running system, things might malfunction.

5.2 Software Changes from Version to Version

The individual aspects changed from version to version are outlined in the following in detail. This summary indicates, for example, whether basic settings have been completely reconfigured, whether configuration files have been moved to other places, or whether common applications have been significantly changed. Significant modifications that affect the daily use of the system at either the user level or the administrator level are mentioned here.

Problems and special issues of the respective versions are published online as they are identified. See the links listed below. Important updates of individual packages can be accessed at `http://www.novell.com/products/linuxprofessional/downloads/` using the YaST Online Update. For more information, see Chapter 3, *Online Update* (page 63).

5.2.1 From 9.3 to 10.0

Refer to the article “Known Problems and Special Features in SUSE Linux 10” in the SUSE Support Database at `http://portal.suse.com` under the keyword *special features*.

Becoming the Superuser Using `su`

By default, calling `su` to become `root` does not set the `PATH` for `root`. Either call `su -` to start a login shell with the complete environment for `root` or set `ALWAYS_SET_PATH` to `yes` in `/etc/default/su` if you want to change the default behavior of `su`.

Powersave Configuration Variables

Names of the powersave configuration variables are changed for consistency, but the sysconfig files are still the same. Find more information in Section 32.4.1, “Configuring the powersave Package” (page 537).

PCMCIA

cardmgr no longer manages PC cards. Instead, as with Cardbus cards and other subsystems, a kernel module manages them. All necessary actions are executed by hotplug. The pcmcia start script has been removed and cardctl is replaced by pccardctl. For more information, see /usr/share/doc/packages/pcmciautils/README.SUSE.

Setting Up D-BUS for Interprocess Communication in .xinitrc

Many applications now rely on D-BUS for interprocess communication (IPC). Calling dbus-launch starts dbus-daemon. The systemwide /etc/X11/xinit/xinitrc uses dbus-launch to start the window manager.

If you have a local ~/.xinitrc file, you must change it accordingly. Otherwise applications like f-spot, banshee, tomboy, or Network Manager banshee might fail. Save your old ~/.xinitrc. Then copy the new template file into your home directory with:

```
cp /etc/skel/.xinitrc.template ~/.xinitrc
```

Finally, add your customizations from the saved .xinitrc.

NTP-Related Files Renamed

For reasons of compatibility with LSB (Linux Standard Base), most configuration files and the init script were renamed from xntp to ntp. The new filenames are:

```
/etc/slp.reg.d/ntp.reg
/etc/init.d/ntp
/etc/logrotate.d/ntp
```

```
/usr/sbin/rcntp
/etc/sysconfig/ntp
```

Hotplug Events Handled by the udev Daemon

Hotplug events are now completely handled by the udev daemon (`udevd`). We do not use the event multiplexer system in `/etc/hotplug.d` and `/etc/dev.d` anymore. Instead `udevd` calls all hotplug helper tools directly, according to its rules. Udev rules and helper tools are provided by udev and various other packages.

TEI XSL Stylesheets

Find the TEI XSL stylesheets (`tei-xsl-stylesheets`) with a new directory layout at `/usr/share/xml/tei/stylesheet/rahtz/current`. From there, for example, use `base/p4/html/tei.xsl` to produce HTML output. For more information, see http://www.tei-c.org/Stylesheets/teic/

File System Change Notification for GNOME Applications

For proper functionality, GNOME applications depend on file system change notification support. For local-only file systems, install the gamin package (preferred) or run the FAM daemon. For remote file systems, run FAM on both the server and client and open the firewall for RPC calls by FAM.

GNOME (gnome-vfs2 and libgda) contains a wrapper that picks gamin or fam to provide file system change notification:

- If the FAM daemon is not running, gamin is preferred. (Rationale: Inotify is supported only by gamin and it is more efficient for local file systems).

- If the FAM daemon is running, FAM is preferred (Rationale: If FAM is running, you probably want remote notification, which is supported only by FAM).

5.2.2 From 10.0 to 10.1

Refer to the article “Known Problems and Special Features in SUSE Linux 10” in the SUSE Support Database at `http://www.novell.com/suselinuxportal` under the keyword *special features*.

Apache 2.2

For Apache version 2.2, Chapter 30, *The Apache HTTP Server* (page 481) was completely reworked. In addition, find generic upgrade information at `http://httpd.apache.org/docs/2.2/upgrading.html` and the description of new features at `http://httpd.apache.org/docs/2.2/new_features_2_2.html`.

Starting an FTP Server (vsftpd)

By default, `xinetd` no longer starts the `vsftpd` FTP server. It is now a stand-alone daemon and you must configure it with the YaST runtime editor.

Firefox 1.5: The URL Open Command

With Firefox 1.5, the method for applications to open a Firefox instance or window has changed. The new method was already partly available in former versions where the behavior was implemented in the wrapper script.

If your application does not use `mozilla-xremote-client` or `firefox -remote`, you do not have to change anything. Otherwise the new command to open a URL is `firefox` *`url`* and it does not matter whether Firefox is already running or not. If it is already running, it follows the preference configured in *Open links from other applications in*.

From the command line, you can influence the behavior by using `firefox -new-window` *`url`* or `firefox -new-tab` *`url`*.

Firefox with Pango Support

On some computers, Firefox with Pango support enabled is very slow. The performance seems to depend on the X server. Set `MOZ_DISABLE_PANGO=0` if you want to switch on font rendering for your environment anyway:

```
export MOZ_DISABLE_PANGO=0
firefox
```

Updating to MySQL 5.0

As with every major release update, it is strongly recommended to perform a backup of the MySQL table files and create an SQL dump beforehand. After the update, `/etc/init.d/mysql` automatically executes `mysql_fix_privilege_tables`. Refer to `http://dev.mysql.com/doc/refman/5.0/en/upgrade.html` for more information and detailed instructions.

Local and IO APIC

The local and IO APIC for the 32-bit x86 architecture has changed. A local and IO APIC (I/O Advanced Programmable Interrupt Controller) is an SMP-capable replacement for PC-style interrupt controllers. SMP systems and all recent uniprocessor systems have such a controller.

Until now, local and IO APIC was disabled on uniprocessor systems by default and had to be manually activated by using the "apic" kernel parameter. Now it runs by default and can be manually deactivated. For 64-bit systems, APIC is always enabled by default.

- Any system with a BIOS version newer than 2001 gets local and IO APIC activated by default unless local and IO APIC is disabled in the BIOS or by the user.

- Any BIOS from Intel newer than 1998 gets local and IO APIC activated by default.

- Any system with more than one CPU gets local and IO APIC activated by default.

If you experience problems with devices not working properly, you can manually apply the following configuration options:

- To disable local APIC, use `nolapic` (this implies disabling IO APICs).

- To disable IO APIC, use `noapic`.

- To get the same default as earlier releases, use `nolapic`.

ulimit Settings

The ulimit settings can be configured in `/etc/sysconfig/ulimit`. By default, only two limits are changed from the kernel defaults:

- `SOFTVIRTUALLIMIT=80` limits a single process so that it does not allocate more than 80% of the available virtual memory (RAM and swap).

- `SOFTRESIDENTLIMIT=85` limits a single process so that it does not occupy more than 85% of the physical memory (RAM).

These soft limits can be overridden with the ulimit command by the user. Hard limits could only be overridden by root.

The values have been chosen conservatively to avoid breaking large processes that have worked before. If there are no legitimate processes with huge memory consumption, set the limits lower to provide more effective protection against run-away processes. The limits are per process and thus not an effective protection against malicious users. The limits are meant to protect against accidental excessive memory usage.

To configure different limits depending on the user, use the pam_limits functionality and configure `/etc/security/limits.conf`. The ulimit package is not required for that, but both mechanisms can be used in parallel. The limits configured in `limits.conf` override the global defaults from the ulimit package.

Unlocking CD and DVD Drives and Ejecting Media

A new mounting mechanism replaces the submount system used earlier. This new mechanism does not unmount media automatically, but on hardware request. Some devices, most notably older CD drives but also some new drives with broken firmware, do not send this signal. To eject the media on such devices, select Eject in the context menu (opened by right-clicking) of the device in "My Computer" or select Eject in the context menu of the device icon on the desktop.

5.2.3 From 10.1 to 10.2

Refer to the “Bugs” article in the openSUSE wiki at `http://en.opensuse.org/Bugs`.

The Standard Kernel

The `kernel-default` package contains the standard kernel for both uniprocessor and multiprocessor systems. The kernel comes with SMP support and runs with minimal overhead on uniprocessor systems. There is no `kernel-smp` package anymore.

Add-On Medium with Additional Languages

Include the language add-on medium in your list of installation sources, if you want better support for one of our tier 2 languages. Tier 2 languages are all but the tier 1 languages (English, French, German, Italian, Spanish, Brazilian Portuguese, simplified and traditional Chinese, Japanese, and Czech). Support for tier 1 languages is available on the standard media set.

5.2.4 From 10.2 to 10.3

Refer to the “Bugs” article in the openSUSE wiki at `http://en.opensuse.org/Bugs`.

6

RPM—the Package Manager

RPM (RPM Package Manager) is used for managing software packages. Its main commands are `rpm` and `rpmbuild`. The powerful RPM database can be queried by the users, system administrators, and package builders for detailed information about the installed software.

Essentially, `rpm` has five modes: installing, uninstalling, or updating software packages; rebuilding the RPM database; querying RPM bases or individual RPM archives; integrity checking of packages; and signing packages. `rpmbuild` can be used to build installable packages from pristine sources.

Installable RPM archives are packed in a special binary format. These archives consist of the program files to install and certain meta information used during the installation by `rpm` to configure the software package or stored in the RPM database for documentation purposes. RPM archives normally have the extension `.rpm`.

TIP: Software Development Packages

For a number of packages, the components needed for software development (libraries, headers, include files, etc.) have been put into separate packages. These development packages are only needed if you want to compile software yourself, for example, the most recent GNOME packages. They can be identified by the name extension `-devel`, such as the packages `alsa-devel`, `gimp-devel`, and `kdelibs3-devel`.

6.1 Verifying Package Authenticity

RPM packages have a GnuPG signature. The key including the fingerprint is:

```
1024D/9C800ACA 2000-10-19 SuSE Package Signing Key <build@suse.de>
Key fingerprint = 79C1 79B2 E1C8 20C1 890F  9994 A84E DAE8 9C80 0ACA
```

The command `rpm --checksig` *`package`*`-1.2.3.rpm` can be used to verify the signature of an RPM package to determine whether it really originates from SUSE or from another trustworthy facility. This is especially recommended for update packages from the Internet. The SUSE public package signature key normally resides in `/root/.gnupg/`. The key is additionally located in the directory `/usr/lib/rpm/gnupg/` to enable normal users to verify the signature of RPM packages.

6.2 Managing Packages: Install, Update, and Uninstall

Normally, the installation of an RPM archive is quite simple: `rpm -i` *`package`*`.rpm`. With this command, the package is installed, but only if its dependencies are fulfilled and there are no conflicts with other packages. With an error message, `rpm` requests those packages that need to be installed to meet dependency requirements. In the background, the RPM database ensures that no conflicts arise—a specific file can only belong to one package. By choosing different options, you can force `rpm` to ignore these defaults, but this is only for experts. Otherwise, risk compromising the integrity of the system and possibly jeopardize the ability to update the system.

The options `-U` or `--upgrade` and `-F` or `--freshen` can be used to update a package, for example, `rpm -F` *`package`*`.rpm`. This command removes the files of the old version and immediately installs the new files. The difference between the two versions is that `-U` installs packages that previously did not exist in the system, but `-F` merely updates previously installed packages. When updating, `rpm` updates configuration files carefully using the following strategy:

- If a configuration file was not changed by the system administrator, `rpm` installs the new version of the appropriate file. No action by the system administrator is required.

- If a configuration file was changed by the system administrator before the update, `rpm` saves the changed file with the extension `.rpmorig` or `.rpmsave` (backup file) and installs the version from the new package, but only if the originally installed file and the newer version are different. If this is the case, compare the backup file (`.rpmorig` or `.rpmsave`) with the newly installed file and make your changes again in the new file. Afterwards, be sure to delete all `.rpmorig` and `.rpmsave` files to avoid problems with future updates.
- `.rpmnew` files appear if the configuration file already exists *and* if the `noreplace` label was specified in the `.spec` file.

Following an update, `.rpmsave` and `.rpmnew` files should be removed after comparing them, so they do not obstruct future updates. The `.rpmorig` extension is assigned if the file has not previously been recognized by the RPM database.

Otherwise, `.rpmsave` is used. In other words, `.rpmorig` results from updating from a foreign format to RPM. `.rpmsave` results from updating from an older RPM to a newer RPM. `.rpmnew` does not disclose any information as to whether the system administrator has made any changes to the configuration file. A list of these files is available in `/var/adm/rpmconfigcheck`. Some configuration files (like `/etc/httpd/httpd.conf`) are not overwritten to allow continued operation.

The `-U` switch is *not* just an equivalent to uninstalling with the `-e` option and installing with the `-i` option. Use `-U` whenever possible.

To remove a package, enter `rpm -e` *`package`*. `rpm` only deletes the package if there are no unresolved dependencies. It is theoretically impossible to delete Tcl/Tk, for example, as long as another application requires it. Even in this case, RPM calls for assistance from the database. If such a deletion is—for whatever reason and under unusual circumstances—impossible, even if *no* additional dependencies exist, it may be helpful to rebuild the RPM database using the option `--rebuilddb`.

6.3 RPM and Patches

To guarantee the operational security of a system, update packages must be installed in the system from time to time. Previously, a bug in a package could only be eliminated by replacing the entire package. Large packages with bugs in small files could easily result in large amounts of data. However the SUSE RPM offers a feature enabling the installation of patches in packages.

The most important considerations are demonstrated using pine as an example:

Is the patch RPM suitable for my system?
: To check this, first query the installed version of the package. For pine, this can be done with

```
rpm -q pine
pine-4.44-188
```

Then check if the patch RPM is suitable for this version of pine:

```
rpm -qp --basedon pine-4.44-224.i586.patch.rpm
pine = 4.44-188
pine = 4.44-195
pine = 4.44-207
```

This patch is suitable for three different versions of pine. The installed version in the example is also listed, so the patch can be installed.

Which files are replaced by the patch?
: The files affected by a patch can easily be seen in the patch RPM. The `rpm` parameter `-P` allows selection of special patch features. Display the list of files with the following command:

```
rpm -qpPl pine-4.44-224.i586.patch.rpm
/etc/pine.conf
/etc/pine.conf.fixed
/usr/bin/pine
```

or, if the patch is already installed, with the following command:

```
rpm -qPl pine
/etc/pine.conf
/etc/pine.conf.fixed
/usr/bin/pine
```

How can a patch RPM be installed in the system?
: Patch RPMs are used just like normal RPMs. The only difference is that a suitable RPM must already be installed.

Which patches are already installed in the system and for which package versions?

A list of all patches installed in the system can be displayed with the command `rpm -qPa`. If only one patch is installed in a new system (as in this example), the list appears as follows:

```
rpm -qPa
pine-4.44-224
```

If, at a later date, you want to know which package version was originally installed, this information is also available in the RPM database. For `pine`, this information can be displayed with the following command:

```
rpm -q --basedon pine
pine = 4.44-188
```

More information, including information about the patch feature of RPM, is available in the man pages of `rpm` and `rpmbuild`.

6.4 Delta RPM Packages

Delta RPM packages contain the difference between an old and a new version of an RPM package. Applying a delta RPM on an old RPM results in the complete new RPM. It is not necessary to have a copy of the old RPM, because a delta RPM can also work with an installed RPM. The delta RPM packages are even smaller in size than patch RPMs, which is an advantage when transferring update packages over the Internet. The drawback is that update operations with delta RPMs involved consume considerably more CPU cycles than plain or patch RPMs.

The `prepdeltarpm`, `writedeltarpm`, and `applydeltarpm` binaries are part of the delta RPM suite (package `deltarpm`) and help you create and apply delta RPM packages. With the following commands, create a delta RPM called `new.delta.rpm`. The following command assumes that `old.rpm` and `new.rpm` are present:

```
prepdeltarpm -s seq -i info old.rpm > old.cpio
prepdeltarpm -f new.rpm > new.cpio
xdelta delta -0 old.cpio new.cpio delta
writedeltarpm new.rpm delta info new.delta.rpm
```

Finally, remove the temporary working files `old.cpio`, `new.cpio`, and `delta`.

Using `applydeltarpm`, you can reconstruct the new RPM from the file system if the old package is already installed:

```
applydeltarpm new.delta.rpm new.rpm
```

To derive it from the old RPM without accessing the file system, use the `-r` option:

```
applydeltarpm -r old.rpm new.delta.rpm new.rpm
```

See `/usr/share/doc/packages/deltarpm/README"` for technical details.

6.5 RPM Queries

With the `-q` option, `rpm` initiates queries, making it possible to inspect an RPM archive (by adding the option `-p`) and also to query the RPM database of installed packages. Several switches are available to specify the type of information required. See Table 6.1, "The Most Important RPM Query Options" (page 90).

Table 6.1 *The Most Important RPM Query Options*

`-i`	Package information
`-l`	File list
`-f FILE`	Query the package that contains the file *`FILE`* (the full path must be specified with *`FILE`*)
`-s`	File list with status information (implies `-l`)
`-d`	List only documentation files (implies `-l`)
`-c`	List only configuration files (implies `-l`)
`--dump`	File list with complete details (to be used with `-l`, `-c`, or `-d`)
`--provides`	List features of the package that another package can request with `--requires`
`--requires, -R`	Capabilities the package requires
`--scripts`	Installation scripts (preinstall, postinstall, uninstall)

For example, the command `rpm -q -i wget` displays the information shown in Example 6.1, “rpm -q -i wget” (page 91).

Example 6.1 *rpm -q -i wget*

```
Name        : wget                         Relocations: (not relocatable)
Version     : 1.9.1                             Vendor: SUSE LINUX AG,
Nuernberg, Germany
Release     : 50                            Build Date: Sat 02 Oct 2004
03:49:13 AM CEST
Install date: Mon 11 Oct 2004 10:24:56 AM CEST      Build Host: f53.suse.de
Group       : Productivity/Networking/Web/Utilities   Source RPM:
wget-1.9.1-50.src.rpm
Size        : 1637514                          License: GPL
Signature   : DSA/SHA1, Sat 02 Oct 2004 03:59:56 AM CEST, Key ID
a84edae89c800aca
Packager    : http://www.suse.de/feedback
URL         : http://wget.sunsite.dk/
Summary     : A tool for mirroring FTP and HTTP servers
Description :
Wget enables you to retrieve WWW documents or FTP files from a server.
This can be done in script files or via the command line.
[...]
```

The option `-f` only works if you specify the complete filename with its full path. Provide as many filenames as desired. For example, the following command

```
rpm -q -f /bin/rpm /usr/bin/wget
```

results in:

```
rpm-4.1.1-191
wget-1.9.1-50
```

If only part of the filename is known, use a shell script as shown in Example 6.2, “Script to Search for Packages” (page 91). Pass the partial filename to the script shown as a parameter when running it.

Example 6.2 *Script to Search for Packages*

```
#! /bin/sh
for i in $(rpm -q -a -l | grep  $1); do
    echo "\"$i\" is in package:"
    rpm -q -f $i
    echo ""
done
```

The command `rpm -q --changelog rpm` displays a detailed list of change information about a specific package, sorted by date. This example shows information about the package `rpm`.

With the help of the installed RPM database, verification checks can be made. Initiate these with `-V`, `-y`, or `--verify`. With this option, `rpm` shows all files in a package that have been changed since installation. `rpm` uses eight character symbols to give some hints about the following changes:

***Table 6.2** RPM Verify Options*

`5`	MD5 check sum
`S`	File size
`L`	Symbolic link
`T`	Modification time
`D`	Major and minor device numbers
`U`	Owner
`G`	Group
`M`	Mode (permissions and file type)

In the case of configuration files, the letter `c` is printed. For example, for changes to `/etc/wgetrc` (`wget`):

```
rpm -V wget
S.5....T c /etc/wgetrc
```

The files of the RPM database are placed in `/var/lib/rpm`. If the partition `/usr` has a size of 1 GB, this database can occupy nearly 30 MB, especially after a complete update. If the database is much larger than expected, it is useful to rebuild the database with the option `--rebuilddb`. Before doing this, make a backup of the old database. The `cron` script `cron.daily` makes daily copies of the database (packed with gzip) and stores them in `/var/adm/backup/rpmdb`. The number of copies is controlled by the variable MAX_RPMDB_BACKUPS (default: 5) in /etc/sysconfig/backup. The size of a single backup is approximately 1 MB for 1 GB in /usr.

6.6 Installing and Compiling Source Packages

All source packages carry a .src.rpm extension (source RPM).

TIP

Source packages can be copied from the installation medium to the hard disk and unpacked with YaST. They are not, however, marked as installed ([i]) in the package manager. This is because the source packages are not entered in the RPM database. Only *installed* operating system software is listed in the RPM database. When you "install" a source package, only the source code is added to the system.

The following directories must be available for rpm and rpmbuild in /usr/src/packages (unless you specified custom settings in a file like /etc/rpmrc):

SOURCES
: for the original sources (.tar.bz2 or .tar.gz files, etc.) and for distribution-specific adjustments (mostly .diff or .patch files)

SPECS
: for the *.spec* files, similar to a meta Makefile, which control the *build* process

BUILD
: all the sources are unpacked, patched, and compiled in this directory

RPMS
: where the completed binary packages are stored

SRPMS
: here are the source RPMs

When you install a source package with YaST, all the necessary components are installed in `/usr/src/packages`: the sources and the adjustments in `SOURCES` and the relevant `.spec` file in `SPECS`.

WARNING

Do not experiment with system components (`glibc`, `rpm`, `sysvinit`, etc.), because this endangers the operability of your system.

The following example uses the `wget.src.rpm` package. After installing the package with YaST, you should have files similar to the following listing:

```
/usr/src/packages/SOURCES/nops_doc.diff
/usr/src/packages/SOURCES/toplev_destdir.diff
/usr/src/packages/SOURCES/wget-1.9.1+ipvmisc.patch
/usr/src/packages/SOURCES/wget-1.9.1-brokentime.patch
/usr/src/packages/SOURCES/wget-1.9.1-passive_ftp.diff
/usr/src/packages/SOURCES/wget-LFS-20040909.tar.bz2
/usr/src/packages/SOURCES/wget-wrong_charset.patch
/usr/src/packages/SPECS/wget.spec
```

`rpmbuild -b` *`X`* `/usr/src/packages/SPECS/wget.spec` starts the compilation. *X* is a wild card for various stages of the build process (see the output of `--help` or the RPM documentation for details). The following is merely a brief explanation:

`-bp`
: Prepare sources in `/usr/src/packages/BUILD`: unpack and patch.

`-bc`
: Do the same as `-bp`, but with additional compilation.

`-bi`
: Do the same as `-bp`, but with additional installation of the built software. Caution: if the package does not support the BuildRoot feature, you might overwrite configuration files.

`-bb`
: Do the same as `-bi`, but with the additional creation of the binary package. If the compile was successful, the binary should be in `/usr/src/packages/RPMS`.

-ba
: Do the same as -bb, but with the additional creation of the source RPM. If the compilation was successful, the binary should be in /usr/src/packages/SRPMS.

--short-circuit
: Skip some steps.

The binary RPM created can now be installed with rpm -i or, preferably, with rpm -U. Installation with rpm makes it appear in the RPM database.

6.7 Compiling RPM Packages with build

The danger with many packages is that unwanted files are added to the running system during the build process. To prevent this, use build, which creates a defined environment in which the package is built. To establish this chroot environment, the build script must be provided with a complete package tree. This tree can be made available on the hard disk, via NFS, or from DVD. Set the position with build --rpms *directory*. Unlike rpm, the build command looks for the SPEC file in the source directory. To build wget (like in the above example) with the DVD mounted in the system under /media/dvd, use the following commands as root:

```
cd /usr/src/packages/SOURCES/
mv ../SPECS/wget.spec .
build --rpms /media/dvd/suse/ wget.spec
```

Subsequently, a minimum environment is established at /var/tmp/build-root. The package is built in this environment. Upon completion, the resulting packages are located in /var/tmp/build-root/usr/src/packages/RPMS.

The build script offers a number of additional options. For example, cause the script to prefer your own RPMs, omit the initialization of the build environment, or limit the rpm command to one of the above-mentioned stages. Access additional information with build --help and by reading the build man page.

6.8 Tools for RPM Archives and the RPM Database

Midnight Commander (mc) can display the contents of RPM archives and copy parts of them. It represents archives as virtual file systems, offering all usual menu options of Midnight Commander. Display the HEADER with F3. View the archive structure with the cursor keys and Enter. Copy archive components with F5.

KDE offers the kpackage tool as a front-end for rpm. A full-featured package manager is available as a YaST module (see Chapter 3, *Installing or Removing Software* (↑Start-Up)).

7 Printer Operation

openSUSE® supports printing with many types of printers, including remote network printers. Printers can be configured with YaST or manually. For configuration instructions, refer to Section “Setting Up a Printer” (Chapter 2, *Setting Up Hardware Components with YaST*, ↑Start-Up). Both graphical and command line utilities are available for starting and managing print jobs. If your printer does not work as expected, refer to Section 7.8, “Troubleshooting” (page 108).

CUPS is the standard print system in openSUSE. CUPS is highly user-oriented. In many cases, it is compatible with LPRng or can be adapted with relatively little effort. LPRng is included in openSUSE only for reasons of compatibility.

Printers can be distinguished by interface, such as USB or network, and printer language. When buying a printer, make sure that the printer has an interface (like USB or parallel port) that is available on your hardware and a suitable printer language. Printers can be categorized on the basis of the following three classes of printer languages:

PostScript Printers

> PostScript is the printer language in which most print jobs in Linux and Unix are generated and processed by the internal print system. This language is already quite old and very efficient. If PostScript documents can be processed directly by the printer and do not need to be converted in additional stages in the print system, the number of potential error sources is reduced. Because PostScript printers are subject to substantial license costs, these printers usually cost more than printers without a PostScript interpreter.

Standard Printer (Languages Like PCL and ESC/P)

Although these printer languages are quite old, they are still undergoing expansion to address new features in printers. In the case of known printer languages, the print system can convert PostScript jobs to the respective printer language with the help of Ghostscript. This processing stage is referred to as interpreting. The best-known languages are PCL, which is mostly used by HP printers and their clones, and ESC/P, which is used by Epson printers. These printer languages are usually supported by Linux and produce a decent print result. Linux may not be able to address some functions of extremely new and fancy printers, because the open source developers may still be working on these features. Except for HP developing `hpijs` drivers, there are currently no printer manufacturers who develop Linux drivers and make them available to Linux distributors under an open source license. Most of these printers are in the medium price range.

Proprietary Printers (Also Called GDI Printers)

These printers do not support any of the common printer languages. They use their own undocumented printer languages, which are subject to change when a new edition of a model is released. Usually only Windows drivers are available for these printers. See Section 7.8.1, “Printers without Standard Printer Language Support” (page 108) for more information.

Before you buy a new printer, refer to the following sources to check how well the printer you intend to buy is supported:

`http://www.linuxprinting.org/`

The LinuxPrinting.org printer database.

`http://www.cs.wisc.edu/~ghost/`

The Ghostscript Web page.

`/usr/share/doc/packages/ghostscript/catalog.devices`

List of included drivers.

The online databases always show the latest Linux support status. However, a Linux distribution can only integrate the drivers available at production time. Accordingly, a printer currently rated as “perfectly supported” may not have had this status when the latest openSUSE version was released. Thus, the databases may not necessarily indicate the correct status, but only provide an approximation.

7.1 The Workflow of the Printing System

The user creates a print job. The print job consists of the data to print plus information for the spooler, such as the name of the printer or the name of the printer queue, and, optionally, information for the filter, such as printer-specific options.

At least one dedicated printer queue exists for every printer. The spooler holds the print job in the queue until the desired printer is ready to receive data. When the printer is ready, the spooler sends the data through the filter and back-end to the printer.

The filter converts the data generated by the application that is printing (usually PostScript or PDF, but also ASCII, JPEG, etc.) into printer-specific data (PostScript, PCL, ESC/P, etc.). The features of the printer are described in the PPD files. A PPD file contains printer-specific options with the parameters needed to enable them on the printer. The filter system makes sure that options selected by the user are enabled.

If you use a PostScript printer, the filter system converts the data into printer-specific PostScript. This does not require a printer driver. If you use a non-PostScript printer, the filter system converts the data into printer-specific data using Ghostscript. This requires a Ghostscript printer driver suitable for your printer. The back-end receives the printer-specific data from the filter then passes it to the printer.

7.2 Methods and Protocols for Connecting Printers

There are various possibilities for connecting a printer to the system. The configuration of the CUPS print system does not distinguish between a local printer and a printer connected to the system over the network. In Linux, local printers must be connected as described in the manual of the printer manufacturer. CUPS supports serial, USB, parallel, and SCSI connections. For more information about the printer connection, read the article *CUPS in a Nutshell* in the Support Database at `http://en.opensuse.org/SDB:CUPS_in_a_Nutshell`.

WARNING: Changing Cable Connections in a Running System

When connecting the printer to the machine, do not forget that only USB devices can be plugged in or unplugged during operation. To avoid damaging your system or printer, shut down the system before changing any connections that are not USB.

7.3 Installing the Software

PPD (PostScript printer description) is the computer language that describes the properties, like resolution, and options, such as the availability of a duplex unit. These descriptions are required for using various printer options in CUPS. Without a PPD file, the print data would be forwarded to the printer in a “raw” state, which is usually not desired. During the installation of openSUSE, many PPD files are preinstalled to enable even printers without PostScript support to be used.

To configure a PostScript printer, the best approach is to get a suitable PPD file. Many PPD files are available in the package `manufacturer-PPDs`, which is automatically installed within the scope of the standard installation. See Section 7.7.3, “PPD Files in Various Packages” (page 106) and Section 7.8.2, “No Suitable PPD File Available for a PostScript Printer” (page 109).

New PPD files can be stored in the directory `/usr/share/cups/model/` or added to the print system with YaST (as described in Section “Adding PPD Files with YaST” (Chapter 2, *Setting Up Hardware Components with YaST*, ↑Start-Up)). Subsequently, the PPD file can be selected during the installation.

Be careful if a printer manufacturer wants you to install entire software packages in addition to modifying configuration files. First, this kind of installation would result in the loss of the support provided by openSUSE and, second, print commands may work differently and the system may no longer be able to address devices of other manufacturers. For this reason, the installation of manufacturer software is not recommended.

7.4 Network Printers

A network printer can support various protocols, some of them even concurrently. Although most of the supported protocols are standardized, some manufacturers expand (modify) the standard because they test systems that have not implemented the standard correctly or because they want to provide certain functions that are not available in the standard. Manufacturers then provide drivers for only a few operating systems, eliminating difficulties with those systems. Unfortunately, Linux drivers are rarely provided. The current situation is such that you cannot act on the assumption that every protocol works smoothly in Linux. Therefore, you may have to experiment with various options to achieve a functional configuration.

IMPORTANT: Remote Access Settings

By default, the cupsd only listens on internal network interfaces (`localhost`). When setting up a CUPS network server you need to adjust the `Listen` directive in `/etc/cups/cupsd.conf` to listen to the outer network.

CUPS supports the `socket`, `LPD`, `IPP`, and `smb` protocols.

socket
: *Socket* refers to a connection in which the data is sent to an Internet socket without first performing a data handshake. Some of the socket port numbers that are commonly used are `9100` or `35`. The device URI (uniform resource identifier) syntax is socket://*`IP.of.the.printer`*:*port*, for example, `socket://192.168.2.202:9100/`.

LPD (Line Printer Daemon)
: The proven LPD protocol is described in RFC 1179. Under this protocol, some job-related data, such as the ID of the printer queue, is sent before the actual print data is sent. Therefore, a printer queue must be specified when configuring the LPD protocol for the data transmission. The implementations of diverse printer manufacturers are flexible enough to accept any name as the printer queue. If necessary, the printer manual should indicate what name to use. LPT, LPT1, LP1, or similar names are often used. An LPD queue can also be configured on a different Linux or Unix host in the CUPS system. The port number for an LPD service is `515`. An example device URI is `lpd://192.168.2.202/LPT1`.

IPP (Internet Printing Protocol)
: IPP is a relatively new (1999) protocol based on the HTTP protocol. With IPP, more job-related data is transmitted than with the other protocols. CUPS uses IPP for internal data transmission. This is the preferred protocol for a forwarding queue between two CUPS servers. The name of the print queue is necessary to configure

IPP correctly. The port number for IPP is 631. Example device URIs are ipp://192.168.2.202/ps and ipp://192.168.2.202/printers/ps.

SMB (Windows Share)
CUPS also supports printing on printers connected to Windows shares. The protocol used for this purpose is SMB. SMB uses the port numbers 137, 138, and 139. Example device URIs are smb://user:password@workgroup/smb.example.com/printer, smb://user:password@smb.example.com/printer, and smb://smb.example.com/printer.

The protocol supported by the printer must be determined before configuration. If the manufacturer does not provide the needed information, the command nmap, which comes with the nmap package, can be used to guess the protocol. nmap checks a host for open ports. For example:

```
nmap -p 35,137-139,515,631,9100-10000 printerIP
```

7.4.1 Configuring CUPS with Command Line Tools

Apart from setting CUPS options with YaST when configuring a network printer, CUPS can be configured with command line tools like lpadmin and lpoptions. You need a device URI consisting of a back-end, such as USB, and parameters, like /dev/usb/lp0. For example, the full URI could be parallel:/dev/lp0 (printer connected to the first parallel port) or usb:/dev/usb/lp0 (first detected printer connected to the USB port).

With lpadmin, the CUPS server administrator can add, remove, or manage class and print queues. To add a print queue, use the following syntax:

```
lpadmin -p queue -v device-URI -P PPD-file -E
```

Then the device (-v) is available as *queue* (-p), using the specified PPD file (-P). This means that you must know the PPD file and the name of the device to configure the printer manually.

Do not use -E as the first option. For all CUPS commands, -E as the first argument sets use of an encrypted connection. To enable the printer, -E must be used as shown in the following example:

```
lpadmin -p ps -v parallel:/dev/lp0 -P \
/usr/share/cups/model/Postscript.ppd.gz -E
```

The following example configures a network printer:

```
lpadmin -p ps -v socket://192.168.2.202:9100/ -P \
/usr/share/cups/model/Postscript-level1.ppd.gz -E
```

For more options of `lpadmin`, see the man page of `lpadmin(1)`.

During printer setup, certain options are set as default. These options can be modified for every print job (depending on the print tool used). Changing these default options with YaST is also possible. Using command line tools, set default options as follows:

1. First, list all options:

   ```
   lpoptions -p queue -l
   ```

   Example:

   ```
   Resolution/Output Resolution: 150dpi *300dpi 600dpi
   ```

   The activated default option is identified by a preceding asterisk (`*`).

2. Change the option with `lpadmin`:

   ```
   lpadmin -p queue -o Resolution=600dpi
   ```

3. Check the new setting:

   ```
   lpoptions -p queue -l

   Resolution/Output Resolution: 150dpi 300dpi *600dpi
   ```

When a normal user runs `lpoptions`, the settings are written to `~/.lpoptions`. However, `root` settings are written to `/etc/cups/lpoptions`.

7.5 Graphical Printing Interfaces

Tools such as xpp and the KDE program KPrinter provide a graphical interface for choosing queues and setting both CUPS standard options and printer-specific options made available through the PPD file. You can even use KPrinter as the standard printing interface of non-KDE applications. In the print dialog of these applications, specify either `kprinter` or `kprinter --stdin` as the print command. The command to use depends on how the application transmits the data—just try which one results in starting KPrinter. If set up correctly, the application should open the KPrinter dialog whenever a print job is issued from it, so you can use the dialog to select a queue and set other printing options. This requires that the application's own print setup does not conflict with that of KPrinter and that printing options are only changed through KPrinter after it has been enabled. More information on KPrinter is available in Section “Printing” (Chapter 1, *Getting Started with the KDE Desktop*, ↑KDE User Guide).

7.6 Printing from the Command Line

To print from the command line, enter `lp -d` *`queuename filename`*, substituting the corresponding names for *`queuename`* and *`filename`*.

Some applications rely on the `lp` command for printing. In this case, enter the correct command in the application's print dialog, usually without specifying *`filename`*, for example, `lp -d` *`queuename`*.

7.7 Special Features in openSUSE

A number of CUPS features have been adapted for openSUSE. Some of the most important changes are covered here.

7.7.1 CUPS and Firewall

After having performed a default installation of openSUSE, SuSEfirewall2 is active and the external network devices are configured to be in the `External Zone` which blocks incoming traffic. These default settings have to be adjusted when using CUPS. More information about the SUSEfirewall2 configuration is available in Section 35.4, “SuSEfirewall2” (page 582).

CUPS Client

Normally a CUPS client runs on a regular workstation located in a network behind a firewall. In this case it is recommended to configure the external network devices to be in the `Internal Zone`, so the workstation is reachable from within the network.

CUPS Server

If the CUPS server is part of network protected by a firewall, the external network device should be configured to be in the Internal Zone of the firewall. When being part of the external zone, the TCP and UDP port 631 needs to be opened in order to make the CUPS server available in the network.

7.7.2 Changes in the CUPS Print Service

Generalized Functionality for BrowseAllow and BrowseDeny

The access permissions set for BrowseAllow and BrowseDeny apply to all kinds of packages sent to cupsd. The default settings in /etc/cups/cupsd.conf are as follows:

```
BrowseAllow @LOCAL
BrowseDeny All
```

and

```
<Location />
  Order Deny,Allow
  Deny From All
  Allow From 127.0.0.1
  Allow From 127.0.0.2
  Allow From @LOCAL
</Location>
```

In this way, only LOCAL hosts can access cupsd on a CUPS server. LOCAL hosts are hosts whose IP addresses belong to a non-PPP interface (interfaces whose IFF_POINTOPOINT flags are not set) and whose IP addresses belong to the same network as the CUPS server. Packets from all other hosts are rejected immediately.

cupsd Activated by Default

In a standard installation, cupsd is activated automatically, enabling comfortable access to the queues of CUPS network servers without any additional manual actions. The items in Section "Generalized Functionality for BrowseAllow and BrowseDeny" (page 105) are vital preconditions for this feature, because otherwise the security would not be sufficient for an automatic activation of cupsd.

7.7.3 PPD Files in Various Packages

The YaST printer configuration sets up the queues for CUPS using only the PPD files installed in /usr/share/cups/model. To find the suitable PPD files for the printer model, YaST compares the vendor and model determined during hardware detection with the vendors and models in all PPD files available in /usr/share/cups/model on the system. For this purpose, the YaST printer configuration generates a database from the vendor and model information extracted from the PPD files. When you select a printer from the list of vendors and models, receive the PPD files matching the vendor and model.

The configuration using only PPD files and no other information sources has the advantage that the PPD files in /usr/share/cups/model can be modified freely. The YaST printer configuration recognizes changes and regenerates the vendor and model database. For example, if you only have PostScript printers, normally you do not need the Foomatic PPD files in the cups-drivers package or the Gimp-Print PPD files in the cups-drivers-stp package. Instead, the PPD files for your PostScript printers can be copied directly to /usr/share/cups/model (if they do not already exist in the manufacturer-PPDs package) to achieve an optimum configuration for your printers.

CUPS PPD Files in the cups Package

The generic PPD files in the cups package have been complemented with adapted Foomatic PPD files for PostScript level 1 and level 2 printers:

- /usr/share/cups/model/Postscript-level1.ppd.gz

- /usr/share/cups/model/Postscript-level2.ppd.gz

PPD Files in the cups-drivers Package

Normally, the Foomatic printer filter foomatic-rip is used together with Ghostscript for non-PostScript printers. Suitable Foomatic PPD files have the entries *NickName:

`... Foomatic/Ghostscript driver` and `*cupsFilter: ... foomatic-rip`. These PPD files are located in the `cups-drivers` package.

YaST prefers a Foomatic PPD file if a Foomatic PPD file with the entry `*NickName: ... Foomatic ... (recommended)` matches the printer model and the `manufacturer-PPDs` package does not contain a more suitable PPD file.

Gimp-Print PPD Files in the `cups-drivers-stp` Package

Instead of `foomatic-rip`, the CUPS filter `rastertoprinter` from Gimp-Print can be used for many non-PostScript printers. This filter and suitable Gimp-Print PPD files are available in the `cups-drivers-stp` package. The Gimp-Print PPD files are located in `/usr/share/cups/model/stp/` and have the entries `*NickName: ... CUPS+Gimp-Print` and `*cupsFilter: ... rastertoprinter`.

PPD Files from Printer Manufacturers in the `manufacturer-PPDs` Package

The `manufacturer-PPDs` package contains PPD files from printer manufacturers that are released under a sufficiently liberal license. PostScript printers should be configured with the suitable PPD file of the printer manufacturer, because this file enables the use of all functions of the PostScript printer. YaST prefers a PPD file from the `manufacturer-PPDs` package if the following conditions are met:

- The vendor and model determined during the hardware detection match the vendor and model in a PPD file from the `manufacturer-PPDs` package.

- The PPD file from the `manufacturer-PPDs` package is the only suitable PPD file for the printer model or a there is a Foomatic PPD file with a `*NickName: ... Foomatic/Postscript (recommended)` entry that also matches the printer model.

Accordingly, YaST does not use any PPD file from the `manufacturer-PPDs` package in the following cases:

- The PPD file from the the `manufacturer-PPDs` package does not match the vendor and model. This may happen if the `manufacturer-PPDs` package contains only one PPD file for similar models, for example, if there is no separate PPD file for the individual models of a model series, but the model name is specified in a form like `Funprinter 1000 series` in the PPD file.

- The Foomatic PostScript PPD file is not recommended. This may be because the printer model does not operate efficiently enough in PostScript mode, for example, the printer may be unreliable in this mode because it has too little memory or the printer is too slow because its processor is too weak. Furthermore, the printer may not support PostScript by default, for example, because PostScript support is only available as an optional module.

If a PPD file from the `manufacturer-PPDs` package is suitable for a PostScript printer, but YaST cannot configure it for these reasons, select the respective printer model manually in YaST.

7.8 Troubleshooting

The following sections cover some of the most frequently encountered printer hardware and software problems and ways to solve or circumvent these problems. Among the topics covered are GDI printers, PPD files, and port configuration. Common network printer problems, defective printouts, and queue handling are also addressed.

7.8.1 Printers without Standard Printer Language Support

These printers do not support any common printer language and can only be addressed with special proprietary control sequences. Therefore they can only work with the operating system versions for which the manufacturer delivers a driver. GDI is a programming interface developed by Microsoft* for graphics devices. Usually the manufacturer delivers drivers only for Windows and because the Windows driver uses the GDI interface, these printers are also called *GDI printers*. The actual problem is not the programming interface, but the fact that these printers can only be addressed with the proprietary printer language of the respective printer model.

Some GDI printers can be switched to operate either in GDI mode or one of the standard printer languages. See the manual of the printer whether it is possible. Some models require a special Windows software to do the switch (note that the Windows printer driver may always switch the printer back into GDI mode when printing from Windows). For other GDI printers there are extension modules for a standard printer language available.

Some manufacturers provide proprietary drivers for their printers. The disadvantage of proprietary printer drivers is that there is no guarantee that these work with the installed print system and that they are suitable for the various hardware platforms. In contrast, printers that support a standard printer language do not depend on a special print system version or a special hardware platform.

Instead of spending time trying to make a proprietary Linux driver work, it may be more cost-effective to purchase a supported printer. This would solve the driver problem once and for all, eliminating the need to install and configure special driver software and obtain driver updates that may be required due to new developments in the print system.

7.8.2 No Suitable PPD File Available for a PostScript Printer

If the `manufacturer-PPDs` package does not contain any suitable PPD file for a PostScript printer, it should be possible to use the PPD file from the driver CD of the printer manufacturer or download a suitable PPD file from the Web page of the printer manufacturer.

If the PPD file is provided as a zip archive (.zip) or a self-extracting zip archive (`.exe`), unpack it with `unzip`. First, review the license terms of the PPD file. Then use the `cupstestppd` utility to check if the PPD file complies with “Adobe PostScript Printer Description File Format Specification, version 4.3.” If the utility returns “FAIL,” the errors in the PPD files are serious and are likely to cause major problems. The problem spots reported by `cupstestppd` should be eliminated. If necessary, ask the printer manufacturer for a suitable PPD file.

7.8.3 Parallel Ports

The safest approach is to connect the printer directly to the first parallel port and to select the following parallel port settings in the BIOS:

- I/O address: 378 (hexadecimal)

- Interrupt: irrelevant

- Mode: Normal, SPP, or Output Only

- DMA: disabled

If the printer cannot be addressed on the parallel port despite these settings, enter the I/O address explicitly in accordance with the setting in the BIOS in the form 0x378 in /etc/modprobe.conf. If there are two parallel ports that are set to the I/O addresses 378 and 278 (hexadecimal), enter these in the form 0x378,0x278.

If interrupt 7 is free, it can be activated with the entry shown in Example 7.1, “ /etc/modprobe.conf: Interrupt Mode for the First Parallel Port ” (page 110). Before activating the interrupt mode, check the file /proc/interrupts to see which interrupts are already in use. Only the interrupts currently being used are displayed. This may change depending on which hardware components are active. The interrupt for the parallel port must not be used by any other device. If you are not sure, use the polling mode with irq=none.

Example 7.1 */etc/modprobe.conf: Interrupt Mode for the First Parallel Port*

```
alias parport_lowlevel parport_pc
options parport_pc io=0x378 irq=7
```

7.8.4 Network Printer Connections

Identifying Network Problems

  Connect the printer directly to the computer. For test purposes, configure the printer as a local printer. If this works, the problems are related to the network.

Checking the TCP/IP Network

  The TCP/IP network and name resolution must be functional.

Checking the Remote Accessibility

  By default, the cupsd only listens on internal network interfaces (localhost). Check whether the Listen directive(s) in /etc/cups/cupsd.conf allow access from the outer network:

```
Listen 192.168.2,*:631
```

Checking the Firewall Settings

A CUPS server either needs to be in the internal firewall zone or, when being in the external zone, must be able to send and receive data on the UDP and TCP port 631.

Checking a Remote `lpd`

Use the following command to test if a TCP connection can be established to `lpd` (port `515`) on *`host`*:

```
netcat -z host 515 && echo ok || echo failed
```

If the connection to `lpd` cannot be established, `lpd` may not be active or there may be basic network problems.

As the user `root`, use the following command to query a (possibly very long) status report for *`queue`* on remote *`host`*, provided the respective `lpd` is active and the host accepts queries:

```
echo -e "\004queue" \
| netcat -w 2 -p 722 host 515
```

If `lpd` does not respond, it may not be active or there may be basic network problems. If `lpd` responds, the response should show why printing is not possible on the `queue` on `host`. If you receive a response like that in Example 7.2, "Error Message from `lpd`" (page 111), the problem is caused by the remote `lpd`.

***Example 7.2** Error Message from lpd*

```
lpd: your host does not have line printer access
lpd: queue does not exist
printer: spooling disabled
printer: printing disabled
```

Checking a Remote `cupsd`

By default, the CUPS network server should broadcast its queues every 30 seconds on UDP port `631`. Accordingly, the following command can be used to test whether there is a CUPS network server in the network.

```
netcat -u -l -p 631 & PID=$! ; sleep 40 ; kill $PID
```

If a broadcasting CUPS network server exists, the output appears as shown in Example 7.3, "Broadcast from the CUPS Network Server" (page 112).

Example 7.3 *Broadcast from the CUPS Network Server*

```
ipp://192.168.2.202:631/printers/queue
```

The following command can be used to test if a TCP connection can be established to `cupsd` (port `631`) on *`host`*:

```
netcat -z host 631 && echo ok || echo failed
```

If the connection to `cupsd` cannot be established, `cupsd` may not be active or there may be basic network problems. `lpstat -h` *`host`* `-l -t` returns a (possibly very long) status report for all queues on *`host`*, provided the respective `cupsd` is active and the host accepts queries.

The next command can be used to test if the *`queue`* on *`host`* accepts a print job consisting of a single carriage-return character. Nothing should be printed. Possibly, a blank page may be ejected.

```
echo -en "\r" \
| lp -d queue -h host
```

Troubleshooting a Network Printer or Print Server Box

Spoolers running in a print server box sometimes cause problems when they have to deal with a lot of print jobs. Because this is caused by the spooler in the print server box, there is nothing you can do about it. As a work-around, circumvent the spooler in the print server box by addressing the printer connected to the print server box directly with TCP socket. See Section 7.4, "Network Printers" (page 100).

In this way, the print server box is reduced to a converter between the various forms of data transfer (TCP/IP network and local printer connection). To use this method, you need to know the TCP port on the print server box. If the printer is connected to the print server box and powered on, this TCP port can usually be determined with the `nmap` utility from the `nmap` package some time after the print server box is powered on. For example, `nmap` *`IP-address`* may deliver the following output for a print server box:

```
Port        State        Service
23/tcp      open         telnet
80/tcp      open         http
515/tcp     open         printer
631/tcp     open         cups
9100/tcp    open         jetdirect
```

This output indicates that the printer connected to the print server box can be addressed via TCP socket on port `9100`. By default, `nmap` only checks a number of commonly known ports listed in `/usr/share/nmap/nmap-services`. To check all possible ports, use the command `nmap -p` *`from_port-to_port IP-address`*. This may take some time. For further information, refer to the man page of `nmap`.

Enter a command like

```
echo -en "\rHello\r\f" | netcat -w 1 IP-address port
cat file | netcat -w 1 IP-address port
```

to send character strings or files directly to the respective port to test if the printer can be addressed on this port.

7.8.5 Defective Printouts without Error Message

For the print system, the print job is completed when the CUPS back-end completes the data transfer to the recipient (printer). If the further processing on the recipient fails, for example, if the printer is not able to print the printer-specific data, the print system does not notice this. If the printer is not able to print the printer-specific data, select a different PPD file that is more suitable for the printer.

7.8.6 Disabled Queues

If the data transfer to the recipient fails entirely after several attempts, the CUPS back-end, such as `USB` or `socket`, reports an error to the print system (to `cupsd`). The back-end decides whether and how many attempts make sense until the data transfer is reported as impossible. Because further attempts would be in vain, `cupsd` disables printing for the respective queue. After eliminating the cause of the problem, the system administrator must reenable printing with the command `/usr/bin/enable`.

7.8.7 CUPS Browsing: Deleting Print Jobs

If a CUPS network server broadcasts its queues to the client hosts via browsing and a suitable local `cupsd` is active on the client hosts, the client `cupsd` accepts print jobs from applications and forwards them to the `cupsd` on the server. When `cupsd` accepts a print job, it is assigned a new job number. Therefore, the job number on the client host is different from the job number on the server. Because a print job is usually forwarded immediately, it cannot be deleted with the job number on the client host, because the client `cupsd` regards the print job as completed as soon as it has been forwarded to the server `cupsd`.

To delete the print job on the server, use a command such as `lpstat -h cups.example.com -o` to determine the job number on the server, provided the server has not already completed the print job (that is, sent it completely to the printer). Using this job number, the print job on the server can be deleted:

```
cancel -h cups.example.com queue-jobnnumber
```

7.8.8 Defective Print Jobs and Data Transfer Errors

Print jobs remain in the queues and printing resumes if you switch the printer off and on or shut down and reboot the computer during the printing process. Defective print jobs must be removed from the queue with `cancel`.

If a print job is defective or an error occurs in the communication between the host and the printer, the printer prints numerous sheets of paper with unintelligible characters, because it is unable to process the data correctly. To deal with this, follow these steps:

1. To stop printing, remove all paper from ink jet printers or open the paper trays of laser printers. High-quality printers have a button for canceling the current printout.

2. The print job may still be in the queue, because jobs are only removed after they are sent completely to the printer. Use `lpstat -o` or `lpstat -h cups.example.com -o` to check which queue is currently printing. Delete the print job with `cancel` *`queue-jobnumber`* or `cancel -h cups.example.com` *`queue-jobnumber`*.

3. Some data may still be transferred to the printer even though the print job has been deleted from the queue. Check if a CUPS back-end process is still running for the respective queue and terminate it. For example, for a printer connected to the parallel port, the command `fuser -k /dev/lp0` can be used to terminate all processes that are still accessing the printer (more precisely: the parallel port).

4. Reset the printer completely by switching it off for some time. Then insert the paper and turn on the printer.

7.8.9 Debugging the CUPS Print System

Use the following generic procedure to locate problems in the CUPS print system:

1. Set `LogLevel debug` in `/etc/cups/cupsd.conf`.

2. Stop `cupsd`.

3. Remove `/var/log/cups/error_log*` to avoid having to search through very large log files.

4. Start `cupsd`.

5. Repeat the action that led to the problem.

6. Check the messages in `/var/log/cups/error_log*` to identify the cause of the problem.

7.8.10 For More Information

Solutions to many specific problems are presented in the SUSE Support Database (`http://en.opensuse.org/SDB:SDB`). Locate the relevant articles with a text search for `SDB:CUPS`.

The X Window System

8

The X Window System (X11) is the de facto standard for graphical user interfaces in UNIX. X is network-based, enabling applications started on one host to be displayed on another host connected over any kind of network (LAN or Internet). This chapter describes the setup and optimization of the X Window System environment, and provides background information about the use of fonts in openSUSE®.

8.1 Manually Configuring the X Window System

By default, the X Window System is configured with the SaX2 interface, described in Section “Setting Up Graphics Card and Monitor” (Chapter 2, *Setting Up Hardware Components with YaST*, ↑Start-Up). Alternatively it can be configured manually by editing the its configuration files.

WARNING: Faulty X Configurations can Damage Your Hardware

Be very careful when configuring your X Window System. Never start the X Window System until the configuration is finished. A misconfigured system can cause irreparable damage to your hardware (this applies especially to fixed-frequency monitors). The creators of this book and openSUSE cannot be held responsible for any resulting damage. This information has been carefully researched, but this does not guarantee that all methods presented here are correct and cannot damage your hardware.

The command sax2 creates the /etc/X11/xorg.conf file. This is the primary configuration file of the X Window System. Find all the settings here concerning your graphics card, mouse, and monitor.

The following sections describe the structure of the configuration file /etc/X11/xorg.conf. It consists of several sections, each one dealing with a certain aspect of the configuration. Each section starts with the keyword Section <designation> and ends with EndSection. The following convention applies to all sections:

```
Section "designation"
  entry 1
  entry 2
  entry n
EndSection
```

The section types available are listed in Table 8.1, "Sections in /etc/X11/xorg.conf" (page 118).

***Table 8.1** Sections in /etc/X11/xorg.conf*

Type	Meaning
Files	The paths used for fonts and the RGB color table.
ServerFlags	General switches for the server behavior.
Module	A list of modules the server should load
InputDevice	Input devices, like keyboards and special input devices (touchpads, joysticks, etc.), are configured in this section. Important parameters in this section are Driver and the options defining the Protocol and Device. You normally have one InputDevice section per device attached to the computer.
Monitor	The monitor used. Important elements of this section are the Identifier, which is referred to later in the Screen definition, the refresh rate VertRefresh, and the synchronization frequency limits (HorizSync and VertRefresh). Settings are given in MHz, kHz, and Hz. Normally, the server refuses any modeline that does not correspond with the specification of

Type	Meaning
	the monitor. This prevents too high frequencies from being sent to the monitor by accident.
`Modes`	The modeline parameters for the specific screen resolutions. These parameters can be calculated by SaX2 on the basis of the values given by the user and normally do not need to be changed. Intervene manually at this point if, for example, you want to connect a fixed frequency monitor. Find details of the meaning of individual number values in the HOWTO files in `/usr/share/doc/howto/en/html/XFree86-Video-Timings-HOWTO` (available in the `howtoenh` package).
`Device`	A specific graphics card. It is referenced by its descriptive name.
`Screen`	Combines a `Monitor` and a `Device` to form all the necessary settings for X.Org. In the `Display` subsection, specify the size of the virtual screen (`Virtual`), the `ViewPort`, and the `Modes` used with this screen.
`ServerLayout`	The layout of a single or multihead configuration. This section binds the input devices `InputDevice` and the display devices `Screen`.
`DRI`	Provides information for the Direct Rendering Infrastructure (DRI).

`Monitor`, `Device`, and `Screen` are explained in more detail. Further information about the other sections can be found in the manual pages of `X.Org` and `xorg.conf`.

There can be several different `Monitor` and `Device` sections in `xorg.conf`. Even multiple `Screen` sections are possible. The `ServerLayout` section determines which of these sections is used.

8.1.1 Screen Section

The screen section combines a monitor with a device section and determines the resolution and color depth to use. A screen section might resemble Example 8.1, "Screen Section of the File /etc/X11/xorg.conf" (page 120).

Example 8.1 *Screen Section of the File /etc/X11/xorg.conf*

```
Section "Screen"❶
  DefaultDepth  16❷
  SubSection "Display"❸
    Depth       16❹
    Modes       "1152x864" "1024x768" "800x600"❺
    Virtual     1152x864❻
  EndSubSection
  SubSection "Display"
    Depth       24
    Modes       "1280x1024"
  EndSubSection
  SubSection "Display"
    Depth       32
    Modes "640x480"
  EndSubSection
  SubSection "Display"
    Depth        8
    Modes       "1280x1024"
  EndSubSection
  Device        "Device[0]"
  Identifier    "Screen[0]"❼
  Monitor       "Monitor[0]"
EndSection
```

❶ `Section` determines the section type, in this case `Screen`.

❷ `DefaultDepth` determines the color depth to use by default unless another color depth is explicitly specified.

❸ For each color depth, different `Display` subsections are specified.

❹ `Depth` determines the color depth to be used with this set of `Display` settings. Possible values are `8`, `15`, `16`, `24`, and `32`, though not all of these might be supported by all X server modules or resolutions.

❺ The `Modes` section comprises a list of possible screen resolutions. The list is checked by the X server from left to right. For each resolution, the X server searches for a suitable `Modeline` in the `Modes` section. The `Modeline` depends on the capability of both the monitor and the graphics card. The `Monitor` settings determine the resulting `Modeline`.

The first resolution found is the `Default mode`. With Ctrl + Alt + + (on the number pad), switch to the next resolution in the list to the right. With Ctrl + Alt + – (on the number pad), switch to the previous. This enables you to vary the resolution while X is running.

❻ The last line of the `Display` subsection with `Depth 16` refers to the size of the virtual screen. The maximum possible size of a virtual screen depends on the amount of memory installed on the graphics card and the desired color depth, not on the maximum resolution of the monitor. If you omit this line, the virtual resolution is just the physical resolution. Because modern graphics cards have a large amount of video memory, you can create very large virtual desktops. However, you may no longer be able to use 3D functionality if you fill most of the video memory with a virtual desktop. If, for example, the card has 16 MB of video RAM, the virtual screen can take up to 4096x4096 pixels in size at 8-bit color depth. Especially for accelerated cards, however, it is not recommended to use all your memory for the virtual screen, because the card's memory is also used for several font and graphics caches.

❼ The `Identifier` line (here `Screen[0]`) gives this section a defined name with which it can be uniquely referenced in the following `ServerLayout` section. The lines `Device` and `Monitor` specify the graphics card and the monitor that belong to this definition. These are just links to the `Device` and `Monitor` sections with their corresponding names or *identifiers*. These sections are discussed in detail below.

8.1.2 Device Section

A device section describes a specific graphics card. You can have as many device entries in `xorg.conf` as you like, provided their names are differentiated using the keyword `Identifier`. If you have more than one graphics card installed, the sections are simply numbered in order. The first one is called `Device[0]`, the second one `Device[1]`, and so on. The following file shows an excerpt from the `Device` section of a computer with a Matrox Millennium PCI graphics card (as configured by SaX2):

```
Section "Device"
  BoardName      "MGA2064W"
  BusID          "0:19:0"❶
  Driver         "mga"❷
  Identifier     "Device[0]"
  VendorName     "Matrox"
  Option         "sw_cursor"
EndSection
```

❶ The BusID refers to the PCI or AGP slot in which the graphics card is installed. This matches the ID displayed by the command lspci. The X server needs details in decimal form, but lspci displays these in hexadecimal form. The value of BusID is automatically detected by SaX2.

❷ The value of Driver is automatically set by SaX2 and specifies which driver to use for your graphics card. If the card is a Matrox Millennium, the driver module is called mga. The X server then searches through the ModulePath defined in the Files section in the drivers subdirectory. In a standard installation, this is the /usr/lib/xorg/modules/drivers directory. _drv.o is added to the name, so, in the case of the mga driver, the driver file mga_drv.o is loaded.

The behavior of the X server or of the driver can also be influenced through additional options. An example of this is the option sw_cursor, which is set in the device section. This deactivates the hardware mouse cursor and depicts the mouse cursor using software. Depending on the driver module, there are various options available, which can be found in the description files of the driver modules in the directory /usr/share/doc/*package_name*. Generally valid options can also be found in the manual pages (man xorg.conf, man X.Org, and man 4 chips).

If the graphics card has multiple video connectors, it is possible to configure the different devices of this single card as one single view. Use SaX2 to set up your graphics interface this way.

8.1.3 Monitor and Modes Section

Like the Device sections, the Monitor and Modes sections describe one monitor each. The configuration file /etc/X11/xorg.conf can contain as many Monitor sections as desired. Each Monitor section references a Modes section with the line UseModes if available. If no Modes section is available for the Monitor section, the X server calculates appropriate values from the general synchronization values. The server layout section specifies which Monitor section is relevant.

Monitor definitions should only be set by experienced users. The modelines are an important part of the Monitor sections. Modelines set horizontal and vertical timings for the respective resolution. The monitor properties, especially the allowed frequencies, are stored in the `Monitor` section.

WARNING

Unless you have in-depth knowledge of monitor and graphics card functions, do not change the modelines, because this could severely damage your monitor.

Those who try to develop their own monitor descriptions should be very familiar with the documentation in `/usr/share/X11/doc`.

Manual specification of modelines is rarely required today. If you are using a modern multisync monitor, the allowed frequencies and optimal resolutions can, as a rule, be read directly from the monitor by the X server via DDC, as described in the SaX2 configuration section. If this is not possible for some reason, use one of the VESA modes included in the X server. This will work with almost all graphics card and monitor combinations.

8.2 Installing and Configuring Fonts

The installation of additional fonts in openSUSE is very easy. Simply copy the fonts to any directory located in the X11 font path (see Section 8.2.1, “X11 Core Fonts” (page 124)). To the enable use of the fonts, the installation directory should be a subdirectory of the directories configured in `/etc/fonts/fonts.conf` (see Section 8.2.2, “Xft” (page 125)) or included into this file with `/etc/fonts/suse-font-dirs.conf`.

The following is an excerpt from `/etc/fonts/suse-font-dirs.conf`. This file is included into the configuration, because it is linked into the directory `/etc/fonts/conf.d` which is included by `/etc/fonts/fonts.conf`. In this directory, all files or symbolic links starting with a two digit number are loaded by fontconfig. For a more detailed explanation of this functionality, have a look at `/etc/fonts/conf.d/README`.

```
<!-- Font directory list -->
<dir>/usr/share/fonts</dir>
<dir>/usr/X11R6/lib/X11/fonts</dir>
<dir>/opt/kde3/share/fonts</dir>
<dir>/usr/local/share/fonts</dir>
```

```
<dir>~/.fonts</dir>
<include ignore_missing="yes">conf.d</include>
```

/etc/fonts/suse-font-dirs.conf is automatically generated to pull in fonts that ship with (mostly third party) applications like OpenOffice.org, Java or Adobe Acrobat Reader. Some typical entries of /etc/fonts/suse-font-dirs.conf would look like the following:

```
<dir>/usr/lib64/ooo-2.0/share/fonts</dir>
<dir>/usr/lib/jvm/java-1_4_2-sun-1.4.2.11/jre/lib/fonts</dir>
<dir>/usr/lib64/jvm/java-1.5.0-sun-1.5.0_07/jre/lib/fonts</dir>
<dir>/usr/X11R6/lib/Acrobat7/Resource/Font</dir>
<dir>/usr/X11R6/lib/Acrobat7/Resource/Font/PFM</dir>
```

To install additional fonts systemwide, manually copy the font files to a suitable directory (as root), such as /usr/share/fonts/truetype. Alternatively, the task can be performed with the KDE font installer in the KDE Control Center. The result is the same.

Instead of copying the actual fonts, you can also create symbolic links. For example, you may want to do this if you have licensed fonts on a mounted Windows partition and want to use them. Subsequently, run SuSEconfig --module fonts .

SuSEconfig --module fonts executes the script /usr/sbin/fonts-config, which handles the font configuration. For more information on this script, refer to its manual page (man fonts-config).

The procedure is the same for bitmap fonts, TrueType and OpenType fonts, and Type1 (PostScript) fonts. All these font types can be installed into any directory.

X.Org contains two completely different font systems: the old *X11 core font system* and the newly designed *Xft and fontconfig* system. The following sections briefly describe these two systems.

8.2.1 X11 Core Fonts

Today, the X11 core font system supports not only bitmap fonts but also scalable fonts, like Type1 fonts, TrueType, and OpenType fonts. Scalable fonts are only supported without antialiasing and subpixel rendering and the loading of large scalable fonts with glyphs for many languages may take a long time. Unicode fonts are also supported, but their use may be slow and require more memory.

The X11 core font system has a few inherent weaknesses. It is outdated and can no longer be extended in a meaningful way. Although it must be retained for reasons of backward compatibility, the more modern Xft and fontconfig system should be used if at all possible.

For its operation, the X server needs to know which fonts are available and where in the system it can find them. This is handled by a `FontPath` variable, which contains the path to all valid system font directories. In each of these directories, a file named `fonts.dir` lists the available fonts in this directory. The `FontPath` is generated by the X server at start-up. It searches for a valid `fonts.dir` file in each of the `FontPath` entries in the configuration file `/etc/X11/xorg.conf`. These entries are found in the `Files` section. Display the actual `FontPath` with `xset q`. This path may also be changed at runtime with `xset`. To add an additional path, use `xset +fp <path>`. To remove an unwanted path, use `xset -fp <path>`.

If the X server is already active, newly installed fonts in mounted directories can be made available with the command `xset fp rehash`. This command is executed by `SuSEconfig --module fonts`. Because the command `xset` needs access to the running X server, this only works if `SuSEconfig --module fonts` is started from a shell that has access to the running X server. The easiest way to achieve this is to assume `root` permissions by entering `su` and the `root` password. `su` transfers the access permissions of the user who started the X server to the `root` shell. To check if the fonts were installed correctly and are available by way of the X11 core font system, use the command `xlsfonts` to list all available fonts.

By default, openSUSE uses UTF-8 locales. Therefore, Unicode fonts should be preferred (font names ending with `iso10646-1` in `xlsfonts` output). All available Unicode fonts can be listed with `xlsfonts | grep iso10646-1`. Nearly all Unicode fonts available in openSUSE contain at least the glyphs needed for European languages (formerly encoded as `iso-8859-*`).

8.2.2 Xft

From the outset, the programmers of Xft made sure that scalable fonts including antialiasing are supported well. If Xft is used, the fonts are rendered by the application using the fonts, not by the X server as in the X11 core font system. In this way, the respective application has access to the actual font files and full control of how the glyphs are rendered. This constitutes the basis for the correct display of text in a number of languages. Direct access to the font files is very useful for embedding fonts for printing to make sure that the printout looks the same as the screen output.

In openSUSE, the two desktop environments KDE and GNOME, Mozilla, and many other applications already use Xft by default. Xft is already used by more applications than the old X11 core font system.

Xft uses the fontconfig library for finding fonts and influencing how they are rendered. The properties of fontconfig are controlled by the global configuration file `/etc/fonts/fonts.conf` and the user-specific configuration file `~/.fonts.conf`. Each of these fontconfig configuration files must begin with

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
```

and end with

```
</fontconfig>
```

To add directories to search for fonts, append lines such as the following:

```
<dir>/usr/local/share/fonts/</dir>
```

However, this is usually not necessary. By default, the user-specific directory `~/.fonts` is already entered in `/etc/fonts/fonts.conf`. Accordingly, all you need to do to install additional fonts is to copy them to `~/.fonts`.

You can also insert rules that influence the appearance of the fonts. For example, enter

```
<match target="font">
 <edit name="antialias" mode="assign">
  <bool>false</bool>
 </edit>
</match>
```

to disable antialiasing for all fonts or

```
<match target="font">
 <test name="family">
  <string>Luxi Mono</string>
  <string>Luxi Sans</string>
 </test>
 <edit name="antialias" mode="assign">
 <bool>false</bool>
 </edit>
</match>
```

to disable antialiasing for specific fonts.

By default, most applications use the font names sans-serif (or the equivalent sans), serif, or monospace. These are not real fonts but only aliases that are resolved to a suitable font, depending on the language setting.

Users can easily add rules to ~/.fonts.conf to resolve these aliases to their favorite fonts:

```
<alias>
 <family>sans-serif</family>
 <prefer>
  <family>FreeSans</family>
 </prefer>
</alias>
<alias>
 <family>serif</family>
 <prefer>
  <family>FreeSerif</family>
 </prefer>
</alias>
<alias>
 <family>monospace</family>
 <prefer>
  <family>FreeMono</family>
 </prefer>
</alias>
```

Because nearly all applications use these aliases by default, this affects almost the entire system. Thus, you can easily use your favorite fonts almost everywhere without having to modify the font settings in the individual applications.

Use the command fc-list to find out which fonts are installed and available for use. For instance, the command fc-list returns a list of all fonts. To find out which of the available scalable fonts (:scalable=true) contain all glyphs required for Hebrew (:lang=he), their font names (family), their style (style), their weight (weight), and the name of the files containing the fonts, enter the following command:

```
fc-list ":lang=he:scalable=true" family style weight
```

The output of this command could look like the following:

```
FreeSansBold.ttf: FreeSans:style=Bold:weight=200
FreeMonoBoldOblique.ttf: FreeMono:style=BoldOblique:weight=200
FreeSerif.ttf: FreeSerif:style=Medium:weight=80
FreeSerifBoldItalic.ttf: FreeSerif:style=BoldItalic:weight=200
FreeSansOblique.ttf: FreeSans:style=Oblique:weight=80
FreeSerifItalic.ttf: FreeSerif:style=Italic:weight=80
```

```
FreeMonoOblique.ttf: FreeMono:style=Oblique:weight=80
FreeMono.ttf: FreeMono:style=Medium:weight=80
FreeSans.ttf: FreeSans:style=Medium:weight=80
FreeSerifBold.ttf: FreeSerif:style=Bold:weight=200
FreeSansBoldOblique.ttf: FreeSans:style=BoldOblique:weight=200
FreeMonoBold.ttf: FreeMono:style=Bold:weight=200
```

Important parameters that can be queried with `fc-list`:

Table 8.2 *Parameters of fc-list*

Parameter	Meaning and Possible Values
`family`	Name of the font family, for example, `FreeSans`.
`foundry`	The manufacturer of the font, for example, `urw`.
`style`	The font style, such as `Medium`, `Regular`, `Bold`, `Italic`, or `Heavy`.
`lang`	The language that the font supports, for example, `de` for German, `ja` for Japanese, `zh-TW` for traditional Chinese, or `zh-CN` for simplified Chinese.
`weight`	The font weight, such as `80` for regular or `200` for bold.
`slant`	The slant, usually `0` for none and `100` for italic.
`file`	The name of the file containing the font.
`outline`	`true` for outline fonts or `false` for other fonts.
`scalable`	`true` for scalable fonts or `false` for other fonts.
`bitmap`	`true` for bitmap fonts or `false` for other fonts.
`pixelsize`	Font size in pixels. In connection with fc-list, this option only makes sense for bitmap fonts.

8.3 For More Information

Install the packages `xorg-x11-doc` and `howtoenh` to get more in-depth information on X11. More information on the X11 development can be found on the project's home page at `http://www.x.org`.

9 Xen Virtualization

This chapter describes and explains the components and technologies you need to understand to set up and manage a Xen-based virtualization environment. It contains the following sections:

For the latest Novell virtualization documentation, see http://www.novell.com/documentation/vmserver/.

9.1 Basic Components

The basic components of a Xen-based virtualization environment are the *Xen hypervisor*, the *host desktop environment*, any number of other *Xen-based virtual machines*, and the tools, commands, and configuration files that let you manage virtualization. Collectively, the physical computer running all these components is referred to as a *virtual machine host* because together these components form a platform for hosting virtual machines.

The Xen Hypervisor

The Xen hypervisor, sometimes referred to generically as a virtual machine monitor, is a program that coordinates the low-level interaction between virtual machines and physical hardware.

The Host Desktop Environment

The virtual machine host's desktop environment, also referred to as *domain 0*, is comprised of several components, such as:

- openSUSE, which gives the administrator a graphical desktop and command line environment to manage the virtual machine host components and its virtual machines.

  NOTE

  The term "host desktop" is used to refer to the `domain 0` management environment, even though openSUSE might be running in command line interface mode with no graphical desktop.

- The xend daemon (xend), which stores configuration information about each virtual machine and controls how virtual machines are created and managed.

- A modified version of QEMU, which is a program that emulates a full computer system, including a processor and various peripherals. It provides the ability to host operating systems in full virtualization mode.

Xen-Based Virtual Machines

A Xen-based virtual machine, also referred to as a *`domain`*, consists of the following components:

- At least one virtual disk that contains a bootable operating system. The virtual disk can be based on a file, partition, volume, or other type of block device.

- Virtual machine configuration information, which can be modified by exporting a text-based configuration file from xend or through Virtual Machine Manager.

Management Tools, Commands, and Configuration Files

There are a combination of GUI tools, commands, and configuration files to help you manage and customize your virtualization environment.

9.1.1 Understanding Virtualization Modes

Guest operating systems are hosted on virtual machines in either full virtualization mode or paravirtual mode. Each virtualization mode has advantages and disadvantages.

- Full virtualization mode lets virtual machines run unmodified operating systems, such as Windows* Server 2003 and Windows XP, but requires the computer running as the virtual machine host to support hardware-assisted virtualization technology, such as AMD* Virtualization or Intel* Virtualization Technology.

  Some guest operating systems hosted in full virtualization mode, can be configured to run openSUSE Virtual Machine Drivers instead of operating-system-specific drivers. Running virtual machine drivers improves performance dramatically in guest operating systems, such as Windows XP and Windows Server 2003. For more information, see “ Virtual Machine Driver Pack for Windows [http://www.novell.com/documentation/vmserver/susedriversforwindows/data/b7cf074.html#b7cf074] ”.

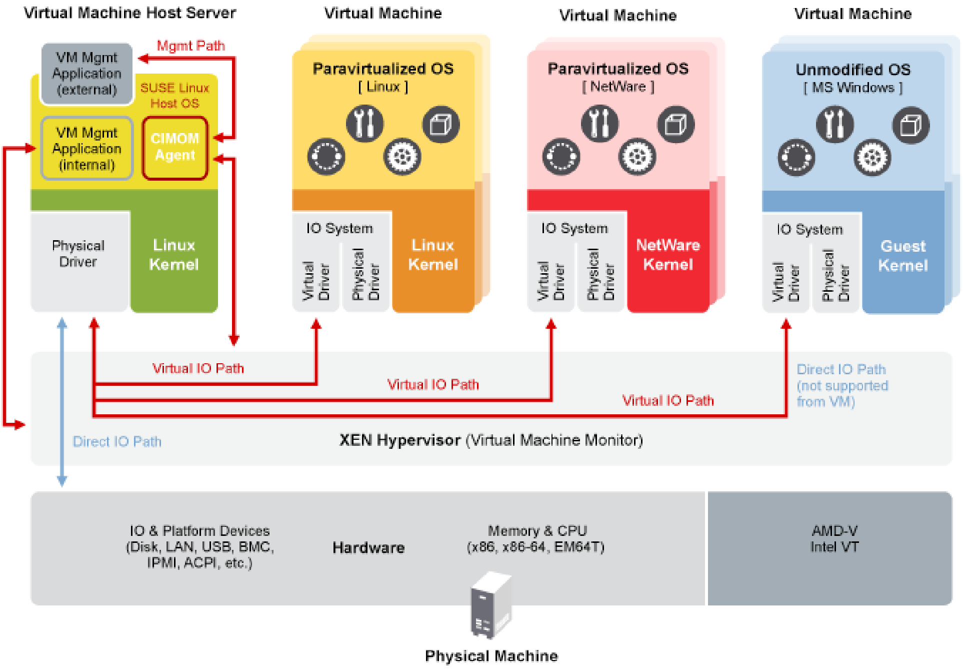
- Paravirtual mode does not require the host computer to support hardware-assisted virtualization technology but does require its operating system to be modified for the virtualization environment. Typically, operating systems running in paravirtual mode enjoy better performance than those requiring full virtualization mode.

9.1.2 Xen Virtualization Architecture

The following graphic depicts a virtual machine host with three virtual machines. The Xen hypervisor is shown as running directly on the physical hardware platform.

Above the Xen hypervisor layer, there are four virtual machines.

***Figure 9.1** Virtualization Architecture*



On the left, the virtual machine host's desktop (domain 0) is shown running a SUSE Linux operating system. The two virtual machines in the middle are shown running paravirtualized openSUSE systems. The virtual machine on the right shows a fully virtual machine running an unmodified operating system, such as Windows Server 2003 or Windows XP.

9.1.3 About the Virtual Machine Host

After you install the virtualization components and reboot the computer, the GRUB menu should display a Xen menu option. Selecting the Xen menu option loads the virtualization components and starts the host desktop `domain 0` running openSUSE.

Running on `domain 0`, openSUSE displays the installed text console or desktop environment, such as GNOME or KDE. When opened, each virtual machine is displayed in its own window.

9.2 Setting Up a Virtual Machine Host

This section documents how to set up and use openSUSE as a virtual machine host. For the latest Novell virtualization documentation, see http://www.novell.com/documentation/vmserver/.

9.2.1 Virtual Machine Host Requirements

The following sections list the system and software requirements for running virtualization.

Hardware Requirements

In most cases, the hardware requirements for the virtual machine host are the same as those for the openSUSE system, but additional CPU, disk, memory, and network resources should be added to accommodate the resource demands of all planned virtual machines.

TIP

Remember that virtual machines, just like a physical machines, perform better when they run on faster processors and have access to more system memory.

The following table lists the minimum hardware requirements for running openSUSE with a Linux virtual machine.

Table 9.1 *Hardware Requirements*

System Component	Minimum Requirements	Recommended Requirements
Computer	Server-class computer with Pentium* II or AMD[*] K7 450 MHz processor	Server-class computer with Pentium III, Pentium III Xeon*, Pentium 4, Intel Xeon 700 MHz, AMD K8 CPUs (Athlon[*] 64 and Opteron[*]),

System Component	Minimum Requirements	Recommended Requirements
		Intel EM64T or higher processor.
Memory	512 MB of RAM for the host	512 MB
	512 MB of RAM for each Linux virtual machine	1 GB of RAM
Free Disk Space	7 GB of available, unpartitioned disk space for the host. 3.7 GB of available, unpartitioned disk space for each Linux virtual machine	7 GB or more of available, unpartitioned disk space for the host. 10 GB or more of available, unpartitioned disk space. Additional disk space might be required depending on which components are selected and how they are used.
CD-ROM Drive	4X CD-ROM drive	48X CD-ROM drive
Hard Drive	20 GB	
Network Board	Ethernet 100 Mbps	
IP Address	• One IP address on a subnet for the host • One IP address on a subnet for each virtual machine • Subnet mask • Default gateway	
Mouse	N/A	USB or PS/2

System Component	Minimum Requirements	Recommended Requirements
Server Computer BIOS	If you are doing a CD-ROM installation, prepare the BIOS on your server computer so that it boots from the CD-ROM drive first.	

Software Requirements

The virtual machine host requires the following software packages and their dependencies to be installed:

- kernel-xen
- xen
- xen-tools
- xen-tools-ioemu (required for full-virtualization mode)
- kernel-xenpae (used instead of kernel-xen, this package is required to enable a 32-bit virtual machine host to access memory over 3 GB)
- yast2-vm

Updates are available through your update channel. Make sure to update to the most recent packages available.

9.2.2 Installing the Xen Virtual Machine Host Software

The Xen virtualization software can be installed during the initial installation of openSUSE or added to a computer already running an operating system.

Remember, for best performance, only those applications and processes required for virtualization should be installed on the virtual machine host.

Installing Virtualization During the Initial Setup of openSUSE

1. During installation of openSUSE, select the software pattern *Xen Virtual Machine Host Server*.

2. Select or remove additional software to meet your specific configuration requirements.

   It is recommended that you select only the software and system tasks required for the virtualization environment. Running additional services, such as Web servers, mail servers, or print servers directly in the management domain (domain 0) affects the performance of all hosted virtual machines.

   For best performance, deselect all but the following items:

   - *Server Base System*
   - *GNOME or KDE Desktop Environment*
   - *X Window System (required for graphical user interface)*
   - *Xen Virtual Machine Host Server*

3. Follow the on-screen instructions to complete the installation program.

The Xen software is now installed on the computer. Continue to Section 9.2.3, "Starting the Virtual Machine Host" (page 139).

Adding Virtualization to openSUSE

If you are already running openSUSE, use one of the following methods from the desktop or command line interface to install the virtualization software.

- From the desktop, run *YaST* > *Virtualization* > *Install Hypervisor and Tools*. Complete the on-screen instructions and restart the computer.

- From a command line interface, enter `yast2 xen`. Complete the on-screen instructions and restart the computer.

9.2.3 Starting the Virtual Machine Host

If virtualization software is correctly installed, the computer boots to display the GRUB boot loader with a *Xen* option on the menu. Select this option to start the virtual machine host.

You can manually edit the boot loader to select the Xen option by default.

1. Log in as the `root` user.

2. In YaST, click *System* > *Boot Loader*.

3. Change the default boot to the *Xen* label, then click *Set as Default*.

4. Click *Finish*.

5. Reboot the system by clicking *Computer* > *Log out* > *Log out*. Then click *Reboot* and enter the root password.

If the GRUB boot loader does not display or the *Xen* option is not on the menu, review the steps for installation and verify that the GRUB boot loader has been updated.

After booting, the management virtual machine starts and displays its graphical desktop environment. If you did not install a graphical desktop, the command line environment appears.

For more information about managing virtual machines, see Section 9.4, “Managing a Virtualization Environment” (page 146).

9.3 Setting Up Virtual Machines

A virtual machine is comprised of data and operating system files that define the virtual environment. Virtual machines are hosted and controlled by the virtual machine host.

This section provides generalized instructions for installing virtual machines. See the *Novell Virtualization Technology: Guest Operating System Guide* [`http://www.novell.com/documentation/vmserver/guest_os_sp1/data/`

`bookinfo.html#bookinfo`] for detailed information regarding installation and management issues for the OS you are installing.

9.3.1 System Requirements

Virtual machines have few if any requirements above those required to run the operating system. If the operating system has not been optimized for the virtual machine host environment, the unmodified OS can run only on hardware-assisted virtualization computer hardware, in full virtualization mode, and requires specific device drivers to be loaded.

You should be aware of any licensing issues related to running a single licensed copy of an operating system on multiple virtual machines. Consult the operating system license agreement for more information.

9.3.2 Virtual Machine System Architectures

The virtual machine host can emulate x86-32 bit and x86-64 bit architectures only. It does not emulate other system architectures such as Itanium*, or IBM* POWER (formerly IBM iSeries and IBM pSeries systems).

9.3.3 Prerequisites

Before creating a virtual machine, you need the following:

- If you want to use an automated installation file (AutoYaST, in openSUSE), you should create and download it to a directory on the host machine server or make it available on the network.

- If you are installing openSUSE, you need a network installation source. For procedures to create the installation sources, see Section 1.2, "Setting Up the Server Holding the Installation Sources" (page 12).

9.3.4 Creating a Virtual Machine

The Create Virtual Machine Wizard helps you through the steps required to create a virtual machine and install its operating system. The information that follows is generalized for installing any operating system. In addition to these instructions, you should consult the *Novell Virtualization Technology: Guest Operating System Guide [http://www.novell.com/documentation/vmserver/guest_os_sp1/data/bookinfo.html#bookinfo]* for information specific to the operating system you are installing.

Launch the Create Virtual Machine Wizard by using one of the following methods:

- From the virtualization host server desktop, click *YaST* > *Virtualization* > *Create Virtual Machine*.

  If YaST doesn't appear in your list of applications, you can run it by logging in as `root` and entering `yast2` at a terminal.

- From within Virtual Machine Manager, click *New*.

- At the command line, enter `vm-install`.

If the wizard does not appear or the `vm-install` command does not work, review the process of installing and starting the virtualization host server. The virtualization software might not be installed properly.

Installing an Operating System

You can choose to run an installation program or choose a disk or disk image that already has an installed and bootable operating system.

If you choose to run an installation program, you are presented with a list of operating systems. Select the one you want to install.

Summary

On the summary page you can click on any of the headings to edit the information. As you edit the information in the Summary, consult the *Novell Virtualization Technology: Guest Operating System Guide [http://www.novell.com/documentation/*

`vmserver/guest_os_sp1/data/bookinfo.html#bookinfo`] for instructions specific to the operating system you are installing.

Virtualization Method

The Virtualization Method page allows you to select the type of virtualization you want to implement.

If your computer supports hardware-assisted virtualization, you can create a virtual machine that runs in fully virtual mode. If you are installing an operating system that is modified for virtualization, you can create a virtual machine that runs in paravirtual mode. For more information about virtualization modes, see Section 9.1.1, "Understanding Virtualization Modes" (page 133).

Name of Virtual Machine

Each virtual machine must have a unique name. The name entered on this page is used to create and name the virtual machine's configuration file. The configuration file contains parameters that define the virtual machine and is saved to the `/etc/xen/vm/` directory.

Hardware

The Hardware page allows you to specify the amount of memory and number of virtual processors for your virtual machine.

Initial Memory:
: The amount of memory initially allocated to the virtual machine (specified in megabytes).

Maximum Memory:
: The largest amount of memory the VM will ever need.

Virtual Processors:
: If desired, you can specify that the virtual machine has more virtual CPUs than the number of physical CPUs. You can specify up to 32 virtual CPUs: however, for best performance, the number of virtual processors should be less than or equal to the number of physical processors.

Graphics

No Graphics Support
: The virtual machine operates like a server without a monitor. You can access the operating system through operating system supported services, such as SSH or VNC.

Paravirtualized Graphics Adapter
: Requires that an appropriate graphics driver is installed in the operating system.

Disks

A virtual machine must have at least one virtual disk. Virtual disks can be:

- File backed, which means that the virtual disk is a single image file on a larger physical disk.

- A sparse image file, which means that the virtual disk is a single image file, but the space is not pre-allocated.

- Configured from a block device, such as an entire disk, partition, or volume.

For best performance, create each virtual disk from an entire disk or a partition. For the next best performance, create an image file but do not create it as a sparse image file. A virtual disk based on a sparse image file delivers the most disk-space flexibility but the slowest installation and disk access speeds.

By default, a single, file-backed 4 GB virtual disk is created as a sparse image file in `/var/lib/xen/images/` *`vm_name`*where *`vm_name`* is the name of the virtual machine. You can change this configuration to meet your specific requirements.

Editing the Properties of a Virtual Disk

When creating the virtual disk, you can click *Edit* to edit its properties to meet your specific configuration requirements.

Type:
: Virtual disks have drive designations that use Linux device naming conventions, such as hda. The designation dictates how the virtual disk appears to the virtual machine. For example, a virtual drive designated as hda appears to the virtual ma-

chine as the first disk device. Paravirtual disks appear as generic disks (not IDE, SCSI, etc.). Fully virtualized disks appear to the guest as IDE. In all cases, naming is left up to the guest OS.

Source

Depending on your specific requirements, you might need to change the location where the disk image file is created and stored, or specify which disk, partition, or volume to use.

Sparse Image File

A virtual disk based on a sparse image file does not consume the entire amount of disk space specified but uses disk space only as needed. This is a good option for quickly creating virtual disks, but for best performance, you might want to “pre-allocate” the disk space by deselecting *Create Sparse Image File*.

Read-Only Access

A virtual disk can be safely shared among multiple virtual machines only if every use of the virtual disk is marked as Read Only.

Additional Virtual Disks

A virtual machine can have more than one virtual disk. When creating a virtual machine, you can create additional virtual disks based on files or specify existing disk images or block devices as additional virtual disks.

Network Adapters

By default, a single virtual network card is created for the virtual machine. It has a randomly generated MAC address that you can change to fit your desired configuration. You can also create additional virtual network cards. In paravirtual machines, virtual network cards communicate by using a generic network card driver compatible with Xen.

Operating System Installation

The operating system can be installed from a CD/DVD device or an ISO image file. In addition, if you are installing a openSUSE system, you can install the operating system from a network installation source.

If you are installing a paravirtual machine's operating system from CD, you should remove the virtual CD reader from the virtual machine after completing the installation because the virtual machine assumes that the original CD is still in the CD reader, even if it is ejected. If it is ejected, the virtual machine cannot access the CD (or any other newly inserted CD) and receives I/O errors.

For instructions on removing the virtual CD reader, see “ Virtual CD Readers [http://www.novell.com/documentation/vmserver/config_options/data/b9rtimf.html#b9rtimf] ” in *Configuration Options and Settings [http://www.novell.com/documentation/vmserver/config_options/data/bookinfo.html#bookinfo]* for more information.

If the installation program is capable of recognizing an installation profile, response file, or script, you can automate the installation settings by specifying the location of the profile, response file, or script you want to use. For example, openSUSE uses an AutoYaST profile.

You can also pass instructions to the kernel at install time by entering parameters for the *Additional Arguments* field.

For example, on openSUSE, if you wanted to specify the parameters for an IP address of `192.35.1.10`, a netmask of `255.255.255.0`, a gateway of `192.35.1.254` for the virtual server, and use SSH to access installation, you could enter the following parameters in the *Additional Argument* field:

```
hostip=192.35.1.10 netmask=255.255.255.0 gateway=192.35.1.254 \
usessh=1 sshpassword=<password>
```

When you have finished entering all the information in the Operating System Installation page, click *Apply* to return to the Summary page.

Continuing the Installation

If all the information on the Summary page is correct, click *OK* to create the virtual machine. A TightVNC screen appears and at this point you begin the installation of your OS. From this point on, follow the regular installation instructions for installing your OS and consult the *Novell Virtualization Technology: Guest Operating System Guide [http://www.novell.com/documentation/vmserver/guest_os_sp1/data/bookinfo.html#bookinfo]* for any instructions that might be different for installing your OS in a virtualized environment.

9.4 Managing a Virtualization Environment

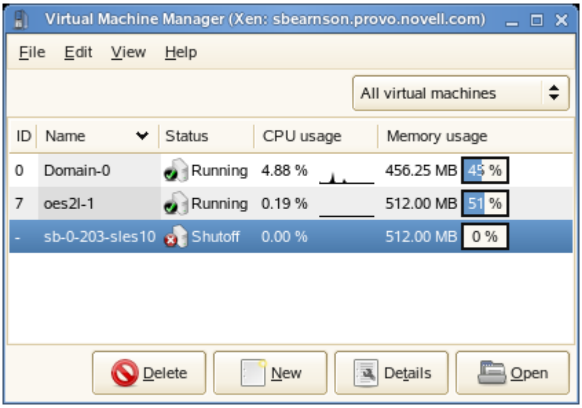
Graphical utilities, text-based commands, and modified configuration files are methods you can choose from to manage your virtualization environment.Virtual Machine Manager is a graphical utility available in YaST that can be launched from the virtual machine host desktop.

From a command line interface on the virtual machine host, you can use the vm-install program and `xm` commands to create and manage virtual machines. You can also edit configuration files to change the settings of the virtual machine host or a virtual machine.

9.4.1 Virtual Machine Manager

From the desktop of the management virtual machine, the YaST Virtual Machine Manager provides a graphical user interface you can use to create and manage virtual machines.

Figure 9.2 *Virtual Machine Manager Main Console*



ID	Name	Status	CPU usage	Memory usage
0	Domain-0	Running	4.88 %	456.25 MB 45 %
7	oes2l-1	Running	0.19 %	512.00 MB 51 %
-	sb-0-203-sles10	Shutoff	0.00 %	512.00 MB 0 %

- Selecting a virtual machine and clicking *Open* displays the virtual machine window showing the virtual machine's current state.

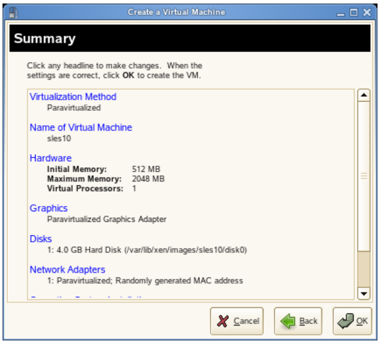
- Clicking *Run* on the virtual machine window boots the virtual machine and displays the user interface or text console running on the virtual machine.

- Selecting a virtual machine and clicking *Details* lets you view performance and configure hardware details associated with the virtual machine.

- Clicking *New* in Virtual Machine Manager launches the *Create Virtual Machine Wizard*, which walks you through the steps required to set up a virtual machine.

The wizard guides you through the process of defining the virtual machine settings and installing the operating system.

Figure 9.3 *Virtual Machine Summary Screen*



Summary

Click any headline to make changes. When the settings are correct, click **OK** to create the VM.

Virtualization Method
Paravirtualized

Name of Virtual Machine
sles10

Hardware
Initial Memory: 512 MB
Maximum Memory: 2048 MB
Virtual Processors: 1

Graphics
Paravirtualized Graphics Adapter

Disks
1: 4.0 GB Hard Disk (/var/lib/xen/images/sles10/disk0)

Network Adapters
1: Paravirtualized; Randomly generated MAC address

After specifying the settings on the *Summary* screen, the wizard starts the virtual machine and launches the operating system installation program, which guides you through the process of installation.

9.4.2 The vm-install Command

In an environment without a graphical user interface, you can launch and run a text version of the Create Virtual Machine Wizard by entering `vm-install` at the command line. The text version guides you through the same process presented in the GUI version.

9.4.3 Controlling the Host by Modifying xend Settings

An important component of the virtual machine host software is the xend daemon. It controls how the virtual machine host performs management functions and stores settings that relate to virtual machines. By modifying the xend configuration file, you can customize the host to meet your specific virtualization requirements.

To modify the xend configuration file, use a text editor to edit the `/etc/xen/xend-config.sxp` file from the management virtual machine's desktop.

After making changes, xend must be restarted for the changes to take effect on the virtual machine host. To restart the xend daemon, enter `service xend restart`

You might also find the following commands useful.

- To check the status of the xend daemon, enter `service xend status`
- To start the xend daemon, enter `service xend start`
- To stop the xend daemon, enter `service xend stop`

9.4.4 Configuring a Virtual Machine by Modifying its xend Settings

You can modify a virtual machine's settings by modifying its xend configuration information. A virtual machine's configuration information is stored in the xend database and can be exported to a file, modified to meet your specific virtualization requirements, and then re-imported back into the xend database.

1. In a terminal on the management virtual machine desktop, enter `xm list -l` *`name > filename`* to export a virtual machine's settings from xend to a file.

   - *`name`* is the virtual machine name
   - *`filename`* is the name you specify as the filename

2. Use a text editor to make and save any desired changes.

3. At the command line, enter `xm new -F filename` to import the new settings into the xend database.

4. Enter `xm start vm_name` to start the virtual machine with the new settings.

NOTE

It is no longer recommended that you edit the initial creation files stored in `etc/xen/vm`, which are used only during the creation of a new virtual machine.

9.4.5 The xm Command

The `xm` command lets you manage and view the status of virtual machines. For example, the `xm list` command displays the status of all virtual machines.

Other useful `xm` commands include:

- `xm start` starts a virtual machine
- `xm reboot` reboots a virtual machine
- `xm destroy` immediately terminates a virtual machine
- `xm block-list` displays all virtual block devices attached to a virtual machine

For a complete list of `xm` command parameters, enter `xm help` at the command line.

10 System Monitoring Utilities

A number of programs and mechanisms, some of which are presented here, can be used to examine the status of your system. Also described are some utilities that are useful for routine work, along with their most important parameters.

For each of the commands introduced, examples of the relevant outputs are presented. In these examples, the first line is the command itself (after the > or # sign prompt). Omissions are indicated with square brackets (`[...]`) and long lines are wrapped where necessary. Line breaks for long lines are indicated by a backslash (`\`).

```
# command -x -y
output line 1
output line 2
output line 3 is annoyingly long, so long that \
    we have to break it
output line 3
[...]
output line 98
output line 99
```

The descriptions have been kept short to allow as many utilities as possible to be mentioned. Further information for all the commands can be found in the man pages. Most of the commands also understand the parameter `--help`, which produces a brief list of the possible parameters.

10.1 Debugging

10.1.1 Specifying the Required Library: ldd

Use the command ldd to find out which libraries would load the dynamic executable specified as argument.

```
tux@mercury:~> ldd /bin/ls
        linux-gate.so.1 =>  (0xffffe000)
        librt.so.1 => /lib/librt.so.1 (0xb7f97000)
        libacl.so.1 => /lib/libacl.so.1 (0xb7f91000)
        libc.so.6 => /lib/libc.so.6 (0xb7e79000)
        libpthread.so.0 => /lib/libpthread.so.0 (0xb7e67000)
        /lib/ld-linux.so.2 (0xb7fb6000)
        libattr.so.1 => /lib/libattr.so.1 (0xb7e63000)
```

Static binaries do not need any dynamic libraries.

```
tux@mercury:~> ldd /bin/sash
        not a dynamic executable
tux@mercury:~> file /bin/sash
/bin/sash: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for
GNU/Linux 2.6.4, statically linked, for GNU/Linux 2.6.4, stripped
```

10.1.2 Library Calls of a Program Run: ltrace

The command ltrace enables you to trace the library calls of a process. This command is used in a similar fashion to strace. The parameter -c outputs the number and duration of the library calls that have occurred:

```
tux@mercury:~> ltrace -c find ~
% time     seconds  usecs/call     calls      function
------ ----------- ----------- --------- --------------------
 34.37    6.758937         245     27554 __errno_location
 33.53    6.593562         788      8358 __fprintf_chk
 12.67    2.490392         144     17212 strlen
 11.97    2.353302         239      9845 readdir64
  2.37    0.466754          27     16716 __ctype_get_mb_cur_max
  1.17    0.230765          27      8358 memcpy
[...]
  0.00    0.000036          36         1 textdomain
------ ----------- ----------- --------- --------------------
100.00   19.662715                105717 total
```

10.1.3 System Calls of a Program Run: `strace`

The utility `strace` enables you to trace all the system calls of a process currently running. Enter the command in the normal way, adding `strace` at the beginning of the line:

```
tux@mercury:~> strace ls
execve("/bin/ls", ["ls"], [/* 61 vars */]) = 0
uname({sys="Linux", node="mercury", ...}) = 0
brk(0)                                  = 0x805c000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or \
    directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=89696, ...}) = 0
mmap2(NULL, 89696, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7ef2000
close(3)                                = 0
open("/lib/librt.so.1", O_RDONLY)       = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0000\36\0"..., 512) \
   = 512
fstat64(3, {st_mode=S_IFREG|0755, st_size=36659, ...}) = 0
[...]
stat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 0), ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) \
    = 0xb7ca7000
write(1, "bin  Desktop  Documents  music\tM"..., 55bin  Desktop  Documents \
   \  music       Music  public_html  tmp
) = 55
close(1)                                = 0
munmap(0xb7ca7000, 4096)                = 0
exit_group(0)                           = ?
```

For example, to trace all attempts to open a particular file, use the following:

```
tux@mercury:~> strace -e open ls .bashrc
open("/etc/ld.so.cache", O_RDONLY)      = 3
open("/lib/librt.so.1", O_RDONLY)       = 3
open("/lib/libacl.so.1", O_RDONLY)      = 3
open("/lib/libc.so.6", O_RDONLY)        = 3
open("/lib/libpthread.so.0", O_RDONLY)  = 3
open("/lib/libattr.so.1", O_RDONLY)     = 3
[...]
```

To trace all the child processes, use the parameter `-f`. The behavior and output format of strace can be largely controlled. For information, see `man strace`.

10.2 Files and File Systems

10.2.1 Determine the File Type: `file`

The command `file` determines the type of a file or a list of files by checking `/etc/magic`.

```
tux@mercury:~> file /usr/bin/file
/usr/bin/file: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), \
    for GNU/Linux 2.2.5, dynamically linked (uses shared libs), stripped
```

The parameter `-f` *`list`* specifies a file with a list of filenames to examine. The `-z` allows `file` to look inside compressed files:

```
tux@mercury:~> file /usr/share/man/man1/file.1.gz
usr/share/man/man1/file.1.gz: gzip compressed data, from Unix, max compression
tux@mercury:~> file -z /usr/share/man/man1/file.1.gz
/usr/share/man/man1/file.1.gz: ASCII troff or preprocessor input text \
    (gzip compressed data, from Unix, max compression)
```

10.2.2 File Systems and Their Usage: `mount`, `df`, and `du`

The command `mount` shows which file system (device and type) is mounted at which mount point:

```
tux@mercury:~> mount
/dev/sda3 on / type reiserfs (rw,acl,user_xattr)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
udev on /dev type tmpfs (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/sda1 on /boot type ext2 (rw,acl,user_xattr)
/dev/sda4 on /local type reiserfs (rw,acl,user_xattr)
/dev/fd0 on /media/floppy type subfs (rw,nosuid,nodev,noatime,fs=floppyfss,p
```

Obtain information about total usage of the file systems with the command `df`. The parameter `-h` (or `--human-readable`) transforms the output into a form understandable for common users.

```
tux@mercury:~> df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda3              11G  3.2G  6.9G  32% /
udev                  252M  104K  252M   1% /dev
/dev/sda1              16M  6.6M  7.8M  46% /boot
/dev/sda4              27G   34M   27G   1% /local
```

Display the total size of all the files in a given directory and its subdirectories with the command `du`. The parameter `-s` suppresses the output of detailed information. `-h` again transforms the data into a human-readable form:

```
tux@mercury:~> du -sh /local
1.7M    /local
```

10.2.3 Additional Information about ELF Binaries

Read the content of binaries with the `readelf` utility. This even works with ELF files that were built for other hardware architectures:

```
tux@mercury:~> readelf --file-header /bin/ls
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Intel 80386
  Version:                           0x1
  Entry point address:               0x8049b60
  Start of program headers:          52 (bytes into file)
  Start of section headers:          81112 (bytes into file)
  Flags:                             0x0
  Size of this header:               52 (bytes)
  Size of program headers:           32 (bytes)
  Number of program headers:         9
  Size of section headers:           40 (bytes)
  Number of section headers:         30
  Section header string table index: 29
```

10.2.4 File Properties: `stat`

The command `stat` displays file properties:

```
tux@mercury:~> stat /etc/profile
  File: `/etc/profile'
  Size: 8080            Blocks: 16         IO Block: 4096   regular file
Device: 806h/2054d      Inode: 64942       Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2007-07-16 23:28:18.000000000 +0200
Modify: 2006-09-19 14:45:01.000000000 +0200
Change: 2006-12-05 14:54:55.000000000 +0100
```

The parameter `--filesystem` produces details of the properties of the file system in which the specified file is located:

```
tux@mercury:~> stat /etc/profile --filesystem
  File: "/etc/profile"
    ID: 0        Namelen: 255     Type: reiserfs
Block size: 4096       Fundamental block size: 4096
Blocks: Total: 2622526    Free: 1809771    Available: 1809771
Inodes: Total: 0          Free: 0
```

10.3 Hardware Information

10.3.1 PCI Resources: `lspci`

The command `lspci` lists the PCI resources:

```
mercury:~ # lspci
00:00.0 Host bridge: Intel Corporation 82845G/GL[Brookdale-G]/GE/PE \
    DRAM Controller/Host-Hub Interface (rev 01)
00:01.0 PCI bridge: Intel Corporation 82845G/GL[Brookdale-G]/GE/PE \
    Host-to-AGP Bridge (rev 01)
00:1d.0 USB Controller: Intel Corporation 82801DB/DBL/DBM \
    (ICH4/ICH4-L/ICH4-M) USB UHCI Controller #1 (rev 01)
00:1d.1 USB Controller: Intel Corporation 82801DB/DBL/DBM \
    (ICH4/ICH4-L/ICH4-M) USB UHCI Controller #2 (rev 01)
00:1d.2 USB Controller: Intel Corporation 82801DB/DBL/DBM \
    (ICH4/ICH4-L/ICH4-M) USB UHCI Controller #3 (rev 01)
00:1d.7 USB Controller: Intel Corporation 82801DB/DBM \
    (ICH4/ICH4-M) USB2 EHCI Controller (rev 01)
00:1e.0 PCI bridge: Intel Corporation 82801 PCI Bridge (rev 81)
00:1f.0 ISA bridge: Intel Corporation 82801DB/DBL (ICH4/ICH4-L) \
    LPC Interface Bridge (rev 01)
00:1f.1 IDE interface: Intel Corporation 82801DB (ICH4) IDE \
    Controller (rev 01)
00:1f.3 SMBus: Intel Corporation 82801DB/DBL/DBM (ICH4/ICH4-L/ICH4-M) \
    SMBus Controller (rev 01)
00:1f.5 Multimedia audio controller: Intel Corporation 82801DB/DBL/DBM \
    (ICH4/ICH4-L/ICH4-M) AC'97 Audio Controller (rev 01)
01:00.0 VGA compatible controller: Matrox Graphics, Inc. G400/G450 (rev 85)
02:08.0 Ethernet controller: Intel Corporation 82801DB PRO/100 VE (LOM) \
    Ethernet Controller (rev 81)
```

Using `-v` results in a more detailed listing:

```
mercury:~ # lspci
[...]
02:08.0 Ethernet controller: Intel Corporation 82801DB PRO/100 VE (LOM)\
    Ethernet Controller (rev 81)
        Subsystem: Fujitsu Siemens Computer GmbH: Unknown device 1001
        Flags: bus master, medium devsel, latency 66, IRQ 11
        Memory at d1000000 (32-bit, non-prefetchable) [size=4K]
        I/O ports at 3000 [size=64]
        Capabilities: [dc] Power Management version 2
```

Information about device name resolution is obtained from the file `/usr/share/pci.ids`. PCI IDs not listed in this file are marked “Unknown device.”

The parameter `-vv` produces all the information that could be queried by the program. To view the pure numeric values, use the parameter `-n`.

10.3.2 USB Devices: `lsusb`

The command `lsusb` lists all USB devices. With the option `-v`, print a more detailed list. The detailed information is read from the directory `/proc/bus/usb/`. The following is the output of `lsusb` with these USB devices attached: hub, memory stick, hard disk, and mouse.

```
mercury:/ # lsusb
Bus 004 Device 007: ID 0ea0:2168 Ours Technology, Inc. Transcend JetFlash \
    2.0 / Astone USB Drive
Bus 004 Device 006: ID 04b4:6830 Cypress Semiconductor Corp. USB-2.0 IDE \
    Adapter
Bus 004 Device 005: ID 05e3:0605 Genesys Logic, Inc.
Bus 004 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 005: ID 046d:c012 Logitech, Inc. Optical Mouse
Bus 001 Device 001: ID 0000:0000
```

10.3.3 Information about a SCSI Device: scsiinfo

The command scsiinfo lists information about a SCSI device. With the option -l, list all SCSI devices known to the system (similar information is obtained via the command lsscsi). The following is the output of scsiinfo -i /dev/sda, which gives information about a hard disk. The option -a gives even more information.

```
mercury:/ # scsiinfo -i /dev/sda
Inquiry command
---------------
Relative Address                   0
Wide bus 32                        0
Wide bus 16                        1
Synchronous neg.                   1
Linked Commands                    1
Command Queueing                   1
SftRe                              0
Device Type                        0
Peripheral Qualifier               0
Removable?                         0
Device Type Modifier               0
ISO Version                        0
ECMA Version                       0
ANSI Version                       3
AENC                               0
TrmIOP                             0
Response Data Format               2
Vendor:                    FUJITSU
Product:                   MAS3367NP
Revision level:            0104A0K7P43002BE
```

The option -d puts out a defects list with two tables of bad blocks of a hard disk: first the one supplied by the vendor (manufacturer table) and second the list of bad blocks that appeared in operation (grown table). If the number of entries in the grown table increases, it might be a good idea to replace the hard disk.

10.4 Networking

10.4.1 Show the Network Status: `netstat`

`netstat` shows network connections, routing tables (`-r`), interfaces (`-i`), masquerade connections (`-M`), multicast memberships (`-g`), and statistics (`-s`).

```
tux@mercury:~> netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.2.0     *               255.255.254.0   U         0 0          0 eth0
link-local      *               255.255.0.0     U         0 0          0 eth0
loopback        *               255.0.0.0       U         0 0          0 lo
default         192.168.2.254   0.0.0.0         UG        0 0          0 eth0
```

```
tux@mercury:~> netstat -i
Kernel Interface table
Iface   MTU Met    RX-OK RX-ERR RX-DRP RX-OVR  TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0   1500   0 1624507 129056      0      0   7055      0      0      0 BMNRU
lo    16436   0   23728      0      0      0  23728      0      0      0 LRU
```

When displaying network connections or statistics, you can specify the socket type to display: TCP (`-t`), UDP (`-u`), or raw (`-r`). The `-p` option shows the PID and name of the program to which each socket belongs.

The following example lists all TCP connections and the programs using these connections.

```
mercury:~ # netstat -t -p
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address  Foreign Address       State       PID/Pro

tcp      0      0 mercury:33513   www.novell.com:www-http ESTABLISHED 6862/fi
tcp      0    352 mercury:ssh      mercury2.:trc-netpoll     ESTABLISHED
19422/s
tcp      0      0 localhost:ssh  localhost:17828         ESTABLISHED -
```

In the following, statistics for the TCP protocol are displayed:

```
tux@mercury:~> netstat -s -t
Tcp:
    2427 active connections openings
    2374 passive connection openings
    0 failed connection attempts
    0 connection resets received
    1 connections established
    27476 segments received
    26786 segments send out
    54 segments retransmited
    0 bad segments received.
    6 resets sent
[...]
    TCPAbortOnLinger: 0
    TCPAbortFailed: 0
    TCPMemoryPressures: 0
```

10.5 The /proc File System

The /proc file system is a pseudo file system in which the kernel reserves important information in the form of virtual files. For example, display the CPU type with this command:

```
tux@mercury:~> cat /proc/cpuinfo
processor       : 0
vendor_id       : AuthenticAMD
cpu family      : 6
model           : 8
model name      : AMD Athlon(tm) XP 2400+
stepping        : 1
cpu MHz         : 2009.343
cache size      : 256 KB
fdiv_bug        : no
[...]
```

Query the allocation and use of interrupts with the following command:

```
tux@mercury:~> cat /proc/interrupts
           CPU0
  0:    3577519          XT-PIC  timer
  1:        130          XT-PIC  i8042
  2:          0          XT-PIC  cascade
  5:     564535          XT-PIC  Intel 82801DB-ICH4
  7:          1          XT-PIC  parport0
  8:          2          XT-PIC  rtc
  9:          1          XT-PIC  acpi, uhci_hcd:usb1, ehci_hcd:usb4
 10:          0          XT-PIC  uhci_hcd:usb3
 11:      71772          XT-PIC  uhci_hcd:usb2, eth0
 12:     101150          XT-PIC  i8042
 14:      33146          XT-PIC  ide0
 15:     149202          XT-PIC  ide1
NMI:          0
LOC:          0
ERR:          0
MIS:          0
```

Some of the important files and their contents are:

`/proc/devices`
    Available devices

`/proc/modules`
    Kernel modules loaded

`/proc/cmdline`
    Kernel command line

`/proc/meminfo`
    Detailed information about memory usage

`/proc/config.gz`
    `gzip`-compressed configuration file of the kernel currently running

Further information is available in the text file `/usr/src/linux/Documentation/filesystems/proc.txt`. Find information about processes currently running in the `/proc/`*`NNN`* directories, where *`NNN`* is the process ID (PID) of the relevant process. Every process can find its own characteristics in `/proc/self/`:

```
tux@mercury:~> ls -l /proc/self
lrwxrwxrwx 1 root root 64 2007-07-16 13:03 /proc/self -> 5356
tux@mercury:~> ls -l /proc/self/
total 0
dr-xr-xr-x 2 tux users 0 2007-07-16 17:04 attr
-r-------- 1 tux users 0 2007-07-16 17:04 auxv
-r--r--r-- 1 tux users 0 2007-07-16 17:04 cmdline
lrwxrwxrwx 1 tux users 0 2007-07-16 17:04 cwd -> /home/tux
-r-------- 1 tux users 0 2007-07-16 17:04 environ
lrwxrwxrwx 1 tux users 0 2007-07-16 17:04 exe -> /bin/ls
dr-x------ 2 tux users 0 2007-07-16 17:04 fd
-rw-r--r-- 1 tux users 0 2007-07-16 17:04 loginuid
-r--r--r-- 1 tux users 0 2007-07-16 17:04 maps
-rw------- 1 tux users 0 2007-07-16 17:04 mem
-r--r--r-- 1 tux users 0 2007-07-16 17:04 mounts
-rw-r--r-- 1 tux users 0 2007-07-16 17:04 oom_adj
-r--r--r-- 1 tux users 0 2007-07-16 17:04 oom_score
lrwxrwxrwx 1 tux users 0 2007-07-16 17:04 root -> /
-rw------- 1 tux users 0 2007-07-16 17:04 seccomp
-r--r--r-- 1 tux users 0 2007-07-16 17:04 smaps
-r--r--r-- 1 tux users 0 2007-07-16 17:04 stat
[...]
dr-xr-xr-x 3 tux users 0 2007-07-16 17:04 task
-r--r--r-- 1 tux users 0 2007-07-16 17:04 wchan
```

The address assignment of executables and libraries is contained in the `maps` file:

```
tux@mercury:~> cat /proc/self/maps
08048000-0804c000 r-xp 00000000 03:03 17753      /bin/cat
0804c000-0804d000 rw-p 00004000 03:03 17753      /bin/cat
0804d000-0806e000 rw-p 0804d000 00:00 0          [heap]
b7d27000-b7d5a000 r--p 00000000 03:03 11867      /usr/lib/locale/en_GB.utf8/
b7d5a000-b7e32000 r--p 00000000 03:03 11868      /usr/lib/locale/en_GB.utf8/
b7e32000-b7e33000 rw-p b7e32000 00:00 0
b7e33000-b7f45000 r-xp 00000000 03:03 8837       /lib/libc-2.3.6.so
b7f45000-b7f46000 r--p 00112000 03:03 8837       /lib/libc-2.3.6.so
b7f46000-b7f48000 rw-p 00113000 03:03 8837       /lib/libc-2.3.6.so
b7f48000-b7f4c000 rw-p b7f48000 00:00 0
b7f52000-b7f53000 r--p 00000000 03:03 11842      /usr/lib/locale/en_GB.utf8/
[...]
b7f5b000-b7f61000 r--s 00000000 03:03 9109       /usr/lib/gconv/gconv-module
b7f61000-b7f62000 r--p 00000000 03:03 9720       /usr/lib/locale/en_GB.utf8/
b7f62000-b7f76000 r-xp 00000000 03:03 8828       /lib/ld-2.3.6.so
b7f76000-b7f78000 rw-p 00013000 03:03 8828       /lib/ld-2.3.6.so
bfd61000-bfd76000 rw-p bfd61000 00:00 0          [stack]
ffffe000-fffff000 ---p 00000000 00:00 0          [vdso]
```

10.5.1 procinfo

Important information from the `/proc` file system is summarized by the command `procinfo`:

```
tux@mercury:~> procinfo
Linux 2.6.18.8-0.5-default (geeko@buildhost) (gcc 4.1.2 20061115) #1 2CPU

Memory:      Total        Used        Free      Shared     Buffers
Mem:       2060604     2011264       49340           0      200664
Swap:      2104472         112     2104360

Bootup: Tue Jul 10 10:29:15 2007    Load average: 0.86 1.10 1.11 3/118 21547

user  :       2:43:13.78   0.8%  page in :   71099181  disk 1:  2827023r 968
nice  :   1d 22:21:27.87  14.7%  page out:  690734737
system:      13:39:57.57   4.3%  page act:  138388345
IOwait:      18:02:18.59   5.7%  page dea:   29639529
hw irq:       0:03:39.44   0.0%  page flt: 9539791626
sw irq:       1:15:35.25   0.4%  swap in :         69
idle  :   9d 16:07:56.79  73.8%  swap out:        209
uptime:   6d 13:07:11.14         context :  542720687

irq  0: 141399308 timer          irq 14:   5074312 ide0
irq  1:     73784 i8042          irq 50:   1938076 uhci_hcd:usb1, ehci_
irq  4:         2                irq 58:         0 uhci_hcd:usb2
irq  6:         5 floppy [2]     irq 66:    872711 uhci_hcd:usb3, HDA I
irq  7:         2                irq 74:        15 uhci_hcd:usb4
irq  8:          0 rtc              irq 82: 178717720 0         PCI-MSI  e
irq  9:          0 acpi             irq169:  44352794 nvidia
irq 12:          3                  irq233:   8209068 0         PCI-MSI  l
```

To see all the information, use the parameter `-a`. The parameter `-nN` produces updates of the information every *N* seconds. In this case, terminate the program by pressing Q.

By default, the cumulative values are displayed. The parameter `-d` produces the differential values. `procinfo -dn5` displays the values that have changed in the last five seconds:

10.6 Processes

10.6.1 Interprocess Communication: `ipcs`

The command `ipcs` produces a list of the IPC resources currently in use:

```
------ Shared Memory Segments --------
key        shmid      owner     perms      bytes      nattch     status
0x00000000 58261504   tux    600        393216     2          dest
0x00000000 58294273   tux    600        196608     2          dest
0x00000000 83886083   tux    666        43264      2
0x00000000 83951622   tux    666        192000     2
0x00000000 83984391   tux    666        282464     2
0x00000000 84738056   root      644        151552     2          dest

------ Semaphore Arrays --------
key        semid      owner     perms      nsems
0x4d038abf 0          tux    600        8

------ Message Queues --------
key        msqid      owner      perms      used-bytes   messages
```

10.6.2 Process List: `ps`

The command `ps` produces a list of processes. Most parameters must be written without a minus sign. Refer to `ps --help` for a brief help or to the man page for extensive help.

To list all processes with user and command line information, use `ps axu`:

```
tux@mercury:~> ps axu
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0    696   272 ?        S    12:59   0:01 init [5]
root         2  0.0  0.0      0     0 ?        SN   12:59   0:00 [ksoftirqd
root         3  0.0  0.0      0     0 ?        S<   12:59   0:00 [events
[...]
tux    4047  0.0  6.0 158548 31400 ?        Ssl  13:02   0:06 mono-best
tux    4057  0.0  0.7   9036  3684 ?        Sl   13:02   0:00 /opt/gnome
tux    4067  0.0  0.1   2204   636 ?        S    13:02   0:00 /opt/gnome
tux    4072  0.0  1.0  15996  5160 ?        Ss   13:02   0:00 gnome-scre
tux    4114  0.0  3.7 130988 19172 ?        SLl  13:06   0:04 sound-juic
tux    4818  0.0  0.3   4192  1812 pts/0    Ss   15:59   0:00 -bash
tux    4959  0.0  0.1   2324   816 pts/0    R+   16:17   0:00 ps axu
```

To check how many `sshd` processes are running, use the option `-p` together with the command `pidof`, which lists the process IDs of the given processes.

```
tux@mercury:~> ps -p `pidof sshd`
  PID TTY      STAT   TIME COMMAND
 3524 ?        Ss     0:00 /usr/sbin/sshd -o PidFile=/var/run/sshd.init.pid
 4813 ?        Ss     0:00 sshd: tux [priv]
 4817 ?        R      0:00 sshd: tux@pts/0
```

The process list can be formatted according to your needs. The option `-L` returns a list of all keywords. Enter the following command to issue a list of all processes sorted by memory usage:

```
tux@mercury:~> ps ax --format pid,rss,cmd --sort rss
  PID   RSS CMD
    2     0 [ksoftirqd/0]
    3     0 [events/0]
    4     0 [khelper]
    5     0 [kthread]
   11     0 [kblockd/0]
   12     0 [kacpid]
  472     0 [pdflush]
  473     0 [pdflush]
[...]
 4028 17556 nautilus --no-default-window --sm-client-id default2
 4118 17800 ksnapshot
 4114 19172 sound-juicer
 4023 25144 gnome-panel --sm-client-id default1
 4047 31400 mono-best --debug /usr/lib/beagle/Best.exe --autostarted
 3973 31520 mono-beagled --debug /usr/lib/beagle/BeagleDaemon.exe --bg --aut
```

10.6.3 Process Tree: pstree

The command pstree produces a list of processes in the form of a tree:

```
tux@mercury:~> pstree
init-+-NetworkManagerD
     |-acpid
     |-3*[automount]
     |-cron
     |-cupsd
     |-2*[dbus-daemon]
     |-dbus-launch
     |-dcopserver
     |-dhcpcd
     |-events/0
     |-gpg-agent
     |-hald-+-hald-addon-acpi
     |      `-hald-addon-stor
     |-kded
     |-kdeinit-+-kdesu---su---kdesu_stub---yast2---y2controlcenter
     |         |-kio_file
     |         |-klauncher
     |         |-konqueror
     |         |-konsole-+-bash---su---bash
     |         |         `-bash
     |         `-kwin
     |-kdesktop---kdesktop_lock---xmatrix
     |-kdesud
     |-kdm-+-X
     |     `-kdm---startkde---kwrapper
[...]
```

The parameter -p adds the process ID to a given name. To have the command lines displayed as well, use the -a parameter:

10.6.4 Processes: top

The command top, which stands for "table of processes," displays a list of processes that is refreshed every two seconds. To terminate the program, press Q. The parameter -n 1 terminates the program after a single display of the process list. The following is an example output of the command top -n 1:

```
tux@mercury:~> top -n 1
top - 17:06:28 up  2:10,  5 users,  load average: 0.00, 0.00, 0.00
Tasks:  85 total,   1 running,  83 sleeping,   1 stopped,   0 zombie
Cpu(s):  5.5% us,  0.8% sy,  0.8% ni, 91.9% id,  1.0% wa,  0.0% hi,  0.0% si
Mem:    515584k total,   506468k used,     9116k free,    66324k buffers
Swap:   658656k total,        0k used,   658656k free,   353328k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
    1 root      16   0   700  272  236 S  0.0  0.1   0:01.33 init
    2 root      34  19     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/0
    3 root      10  -5     0    0    0 S  0.0  0.0   0:00.27 events/0
    4 root      10  -5     0    0    0 S  0.0  0.0   0:00.01 khelper
    5 root      10  -5     0    0    0 S  0.0  0.0   0:00.00 kthread
   11 root      10  -5     0    0    0 S  0.0  0.0   0:00.05 kblockd/0
   12 root      20  -5     0    0    0 S  0.0  0.0   0:00.00 kacpid
  472 root      20   0     0    0    0 S  0.0  0.0   0:00.00 pdflush
  473 root      15   0     0    0    0 S  0.0  0.0   0:00.06 pdflush
  475 root      11  -5     0    0    0 S  0.0  0.0   0:00.00 aio/0
  474 root      15   0     0    0    0 S  0.0  0.0   0:00.07 kswapd0
  681 root      10  -5     0    0    0 S  0.0  0.0   0:00.01 kseriod
  839 root      10  -5     0    0    0 S  0.0  0.0   0:00.02 reiserfs/0
  923 root      13  -4  1712  552  344 S  0.0  0.1   0:00.67 udevd
 1343 root      10  -5     0    0    0 S  0.0  0.0   0:00.00 khubd
 1587 root      20   0     0    0    0 S  0.0  0.0   0:00.00 shpchpd_event
 1746 root      15   0     0    0    0 S  0.0  0.0   0:00.00 w1_control
 1752 root      15   0     0    0    0 S  0.0  0.0   0:00.00 w1_bus_master1
 2151 root      16   0  1464  496  416 S  0.0  0.1   0:00.00 acpid
 2165 messageb  16   0  3340 1048  792 S  0.0  0.2   0:00.64 dbus-daemon
 2166 root      15   0  1840  752  556 S  0.0  0.1   0:00.01 syslog-ng
 2171 root      16   0  1600  516  320 S  0.0  0.1   0:00.00 klogd
 2235 root      15   0  1736  800  652 S  0.0  0.2   0:00.10 resmgrd
 2289 root      16   0  4192 2852 1444 S  0.0  0.6   0:02.05 hald
 2403 root      23   0  1756  600  524 S  0.0  0.1   0:00.00 hald-addon-acpi
 2709 root      19   0  2668 1076  944 S  0.0  0.2   0:00.00 NetworkManagerD
 2714 root      16   0  1756  648  564 S  0.0  0.1   0:00.56 hald-addon-stor
```

If you press F while `top` is running, a menu opens with which to make extensive changes to the format of the output.

The parameter `-U` *`UID`* monitors only the processes associated with a particular user. Replace *`UID`* with the user ID of the user. `` top -U `id -u` `` returns the UID of the user on the basis of the username and displays his processes.

10.7 System Information

10.7.1 System Activity Information: sar

To use sar, sadc (system activity data collector) needs to be running. Check its status or start it with rcsysstat {start|status}.

sar can generate extensive reports on almost all important system activities, among them CPU, memory, IRQ usage, IO, or networking. With its many options, it is too complex to explain further here. Refer to the man page for extensive documentation with examples.

10.7.2 Memory Usage: free

The utility free examines RAM usage. Details of both free and used memory and swap areas are shown:

```
tux@mercury:~> free
             total       used       free     shared    buffers     cached
Mem:        515584     501704      13880          0      73040     334592
-/+ buffers/cache:      94072     421512
Swap:       658656          0     658656
```

The options -b,-k,-m,-g show output in bytes, KB, MB, or GB, respectively. The parameter -d delay ensures that the display is refreshed every *delay* seconds. For example, free -d 1.5 produces an update every 1.5 seconds.

10.7.3 User Accessing Files: fuser

It can be useful to determine what processes or users are currently accessing certain files. Suppose, for example, you want to unmount a file system mounted at /mnt. umount returns "device is busy." The command fuser can then be used to determine what processes are accessing the device:

```
tux@mercury:~> fuser -v /mnt/*

                     USER        PID ACCESS COMMAND
/mnt/notes.txt       tux    26597 f....  less
```

Following termination of the `less` process, which was running on another terminal, the file system can successfully be unmounted.

10.7.4 Kernel Ring Buffer: `dmesg`

The Linux kernel keeps certain messages in a ring buffer. To view these messages, enter the command `dmesg`:

```
$ dmesg
[...]
end_request: I/O error, dev fd0, sector 0
subfs: unsuccessful attempt to mount media (256)
e100: eth0: e100_watchdog: link up, 100Mbps, half-duplex
NET: Registered protocol family 17
IA-32 Microcode Update Driver: v1.14 <tigran@veritas.com>
microcode: CPU0 updated from revision 0xe to 0x2e, date = 08112004
IA-32 Microcode Update Driver v1.14 unregistered
bootsplash: status on console 0 changed to on
NET: Registered protocol family 10
Disabled Privacy Extensions on device c0326ea0(lo)
IPv6 over IPv4 tunneling driver
powernow: This module only works with AMD K7 CPUs
bootsplash: status on console 0 changed to on
```

Older events are logged in the files `/var/log/messages` and `/var/log/warn`.

10.7.5 List of Open Files: `lsof`

To view a list of all the files open for the process with process ID *`PID`*, use `-p`. For example, to view all the files used by the current shell, enter:

```
tux@mercury:~> lsof -p $$
COMMAND  PID   USER   FD   TYPE DEVICE    SIZE   NODE NAME
bash    5552 tux  cwd    DIR    3,3    1512 117619 /home/tux
bash    5552 tux  rtd    DIR    3,3     584      2 /
bash    5552 tux  txt    REG    3,3  498816  13047 /bin/bash
bash    5552 tux  mem    REG    0,0              0 [heap] (stat: No such
bash    5552 tux  mem    REG    3,3  217016 115687 /var/run/nscd/passwd
bash    5552 tux  mem    REG    3,3  208464  11867 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3  882134  11868 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3 1386997   8837 /lib/libc-2.3.6.so
bash    5552 tux  mem    REG    3,3   13836   8843 /lib/libdl-2.3.6.so
bash    5552 tux  mem    REG    3,3  290856  12204 /lib/libncurses.so.5.5
bash    5552 tux  mem    REG    3,3   26936  13004 /lib/libhistory.so.5.1
bash    5552 tux  mem    REG    3,3  190200  13006 /lib/libreadline.so.5.
bash    5552 tux  mem    REG    3,3      54  11842 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3    2375  11663 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3     290  11736 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3      52  11831 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3      34  11862 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3      62  11839 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3     127  11664 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3      56  11735 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3      23  11866 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3   21544   9109 /usr/lib/gconv/gconv-m
bash    5552 tux  mem    REG    3,3     366   9720 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3   97165   8828 /lib/ld-2.3.6.so
bash    5552 tux    0u   CHR  136,5              7 /dev/pts/5
bash    5552 tux    1u   CHR  136,5              7 /dev/pts/5
bash    5552 tux    2u   CHR  136,5              7 /dev/pts/5
bash    5552 tux  255u   CHR  136,5              7 /dev/pts/5
```

The special shell variable `$$`, whose value is the process ID of the shell, has been used.

The command `lsof` lists all the files currently open when used without any parameters. Because there are often thousands of open files, listing all of them is rarely useful. However, the list of all files can be combined with search functions to generate useful lists. For example, list all used character devices:

```
tux@mercury:~> lsof | grep CHR
bash      3838     tux    0u      CHR  136,0                2 /dev/pts/0
bash      3838     tux    1u      CHR  136,0                2 /dev/pts/0
bash      3838     tux    2u      CHR  136,0                2 /dev/pts/0
bash      3838     tux  255u      CHR  136,0                2 /dev/pts/0
bash      5552     tux    0u      CHR  136,5                7 /dev/pts/5
bash      5552     tux    1u      CHR  136,5                7 /dev/pts/5
bash      5552     tux    2u      CHR  136,5                7 /dev/pts/5
bash      5552     tux  255u      CHR  136,5                7 /dev/pts/5
X         5646       root  mem       CHR    1,1              1006 /dev/mem
lsof      5673     tux    0u      CHR  136,5                7 /dev/pts/5
lsof      5673     tux    2u      CHR  136,5                7 /dev/pts/5
grep      5674     tux    1u      CHR  136,5                7 /dev/pts/5
grep      5674     tux    2u      CHR  136,5                7 /dev/pts/5
```

10.7.6 Kernel and udev Event Sequence Viewer: `udevmonitor`

`udevmonitor` listens to the kernel uevents and events sent out by a udev rule and prints the device path (DEVPATH) of the event to the console. This is a sequence of events while connecting a USB memory stick:

```
UEVENT[1138806687] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2
UEVENT[1138806687] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2/4-2.2
UEVENT[1138806687] add@/class/scsi_host/host4
UEVENT[1138806687] add@/class/usb_device/usbdev4.10
UDEV  [1138806687] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2
UDEV  [1138806687] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2/4-2.2
UDEV  [1138806687] add@/class/scsi_host/host4
UDEV  [1138806687] add@/class/usb_device/usbdev4.10
UEVENT[1138806692] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2/4-2.2
UEVENT[1138806692] add@/block/sdb
UEVENT[1138806692] add@/class/scsi_generic/sg1
UEVENT[1138806692] add@/class/scsi_device/4:0:0:0
UDEV  [1138806693] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2/4-2.2
UDEV  [1138806693] add@/class/scsi_generic/sg1
UDEV  [1138806693] add@/class/scsi_device/4:0:0:0
UDEV  [1138806693] add@/block/sdb
UEVENT[1138806694] add@/block/sdb/sdb1
UDEV  [1138806694] add@/block/sdb/sdb1
UEVENT[1138806694] mount@/block/sdb/sdb1
UEVENT[1138806697] umount@/block/sdb/sdb1
```

10.7.7 Server Resources Used by X11 Clients: `xrestop`

`xrestop` provides statistics for each connected X11 client's server-side resource. The output is very similar to Section 10.6.4, “Processes: `top`” (page 165).

```
xrestop - Display: localhost:0
          Monitoring 40 clients. XErrors: 0
          Pixmaps:   42013K total, Other:     206K total, All:   42219K total

res-base Wins   GCs Fnts Pxms Misc   Pxm mem  Other   Total   PID Identifier
3e00000   385   36    1  751  107    18161K     13K  18175K   ?   NOVELL: SU
4600000   391  122    1 1182  889     4566K     33K   4600K   ?   amaroK - S
1600000    35   11    0   76  142     3811K      4K   3816K   ?   KDE Deskto
3400000    52   31    1   69   74     2816K      4K   2820K   ?   Linux Shel
2c00000    50   25    1   43   50     2374K      3K   2378K   ?   Linux Shel
2e00000    50   10    1   36   42     2341K      3K   2344K   ?   Linux Shel
2600000    37   24    1   34   50     1772K      3K   1775K   ?   Root - Kon
4800000    37   24    1   34   49     1772K      3K   1775K   ?   Root - Kon
2a00000   209   33    1  323  238     1111K     12K   1123K   ?   Trekstor25
1800000   182   32    1  302  285     1039K     12K   1052K   ?   kicker
1400000   157  121    1  231  477      777K     18K    796K   ?   kwin
3c00000   175   36    1  248  168      510K      9K    520K   ?   de.comp.la
3a00000   326   42    1  579  444      486K     20K    506K   ?   [opensuse-
0a00000    85   38    1  317  224      102K      9K    111K   ?   Kopete
4e00000    25   17    1   60   66       63K      3K     66K   ?   YaST Contr
2400000    11   10    0   56   51       53K      1K     55K 22061 suseplugge
0e00000    20   12    1   50   92       50K      3K     54K 22016 kded
```

```
3200000     6   41    5   72   84        40K       8K      48K   ?   EMACS
2200000    54    9    1   30   31        42K       3K      45K   ?   SUSEWatche
4400000     2   11    1   30   34        34K       2K      36K 16489 kdesu
1a00000   255    7    0   42   11        19K       6K      26K   ?   KMix
3800000     2   14    1   34   37        21K       2K      24K 22242 knotify
1e00000    10    7    0   42    9        15K     624B      15K   ?   KPowersave
3600000   106    6    1   30    9         7K       3K      11K 22236 konqueror
2000000    10    5    0   21   34         9K       1K      10K   ?   klipper
3000000    21    7    0   11    9         7K     888B       8K   ?   KDE Wallet
```

10.8 User Information

10.8.1 Who Is Doing What: w

With the command w, find out who is logged onto the system and what each user is doing. For example:

```
tux@mercury:~> w
 16:33:03 up  3:33,  2 users,  load average: 0.14, 0.06, 0.02
USER     TTY        LOGIN@   IDLE   JCPU   PCPU WHAT
tux   :0        16:33   ?xdm?   9.42s  0.15s /bin/sh /opt/kde3/bin/startk
tux   pts/0     15:59    0.00s  0.19s  0.00s w
```

If any users of other systems have logged in remotely, the parameter -f shows the computers from which they have established the connection.

10.9 Time and Date

10.9.1 Time Measurement with time

Determine the time spent by commands with the time utility. This utility is available in two versions: as a shell built-in and as a program (/usr/bin/time).

```
tux@mercury:~> time find . > /dev/null

real    0m4.051s
user    0m0.042s
sys     0m0.205s
```

Part III. System

32-Bit and 64-Bit Applications in a 64-Bit System Environment

11

openSUSE® is available for 64-bit platforms. This does not necessarily mean that all the applications included have already been ported to 64-bit platforms. openSUSE supports the use of 32-bit applications in a 64-bit system environment. This chapter offers a brief overview of how this support is implemented on 64-bit openSUSE platforms. It explains how 32-bit applications are executed (runtime support) and how 32-bit applications should be compiled to enable them to run both in 32-bit and 64-bit system environments. Additionally, find information about the kernel API and an explanation of how 32-bit applications can run under a 64-bit kernel.

openSUSE for the 64-bit platforms amd64 and Intel 64 is designed so that existing 32-bit applications run in the 64-bit environment “out-of-the-box.” This support means that you can continue to use your preferred 32-bit applications without waiting for a corresponding 64-bit port to become available.

11.1 Runtime Support

IMPORTANT: Conflicts between Application Versions

If an application is available both for 32-bit and 64-bit environments, parallel installation of both versions is bound to lead to problems. In such cases, decide on one of the two versions and install and use this.

To be executed correctly, every application requires a range of libraries. Unfortunately, the names for the 32-bit and 64-bit versions of these libraries are identical. They must be differentiated from each other in another way.

To retain compatibility with the 32-bit version, the libraries are stored at the same place in the system as in the 32-bit environment. The 32-bit version of `libc.so.6` is located under `/lib/libc.so.6` in both the 32-bit and 64-bit environments.

All 64-bit libraries and object files are located in directories called `lib64`. The 64-bit object files you would normally expect to find under `/lib`, and `/usr/lib` are now found under `/lib64`, and `/usr/lib64`. This means that there is space for the 32-bit libraries under `/lib` and `/usr/lib`, so the filename for both versions can remain unchanged.

Subdirectories of 32-bit `/lib` directories whose data content does not depend on the word size are not moved. This scheme conforms to LSB (Linux Standards Base) and FHS (File System Hierarchy Standard).

11.2 Software Development

A biarch development tool chain allows generation of 32-bit and 64-bit objects. The default is to compile 64-bit objects. It is possible to generate 32-bit objects by using special flags. For GCC, this special flag is `-m32`.

All header files must be written in an architecture-independent form. The installed 32-bit and 64-bit libraries must have an API (application programming interface) that matches the installed header files. The normal openSUSE environment is designed according to this principle. In the case of manually updated libraries, resolve these issues yourself.

11.3 Software Compilation on Biarch Platforms

To develop binaries for the other architecture on a biarch architecture, the respective libraries for the second architecture must additionally be installed. These packages are called `rpmname-32bit`. You also need the respective headers and libraries from the rpmname-devel packages and the development libraries for the second architecture from rpmname-devel-32bit.

Most open source programs use an autoconf-based program configuration. To use autoconf for configuring a program for the second architecture, overwrite the normal compiler and linker settings of autoconf by running the configure script with additional environment variables.

The following example refers to an x86_64 system with x86 as the second architecture.

1. Use the 32-bit compiler:

   ```
   CC="gcc -m32"
   ```

2. Instruct the linker to process 32-bit objects (always use gcc as the linker front-end):

   ```
   LD="gcc -m32"
   ```

3. Set the assembler to generate 32-bit objects:

   ```
   AS="gcc -c -m32"
   ```

4. Determine that the libraries for libtool and so on come from /usr/lib:

   ```
   LDFLAGS="-L/usr/lib"
   ```

5. Determine that the libraries are stored in the lib subdirectory:

   ```
   --libdir=/usr/lib
   ```

6. Determine that the 32-bit X libraries are used:

   ```
   --x-libraries=/usr/lib/xorg
   ```

Not all of these variables are needed for every program. Adapt them to the respective program.

```
CC="gcc -m32"           \
LDFLAGS="-L/usr/lib;"  \
        .configure       \
          --prefix=/usr  \
          --libdir=/usr/lib
make
make install
```

11.4 Kernel Specifications

The 64-bit kernels for x86_64 offer both a 64-bit and a 32-bit kernel ABI (application binary interface). The latter is identical with the ABI for the corresponding 32-bit kernel. This means that the 32-bit application can communicate with the 64-bit kernel in the same way as with the 32-bit kernel.

The 32-bit emulation of system calls for a 64-bit kernel does not support all the APIs used by system programs. This depends on the platform. For this reason, a small number of applications, like `lspci`, must be compiled

A 64-bit kernel can only load 64-bit kernel modules that have been specially compiled for this kernel. It is not possible to use 32-bit kernel modules.

TIP

Some applications require separate kernel-loadable modules. If you intend to use such a 32-bit application in a 64-bit system environment, contact the provider of this application and Novell to make sure that the 64-bit version of the kernel-loadable module and the 32-bit compiled version of the kernel API are available for this module.

Booting and Configuring a Linux System

12

Booting a Linux system involves various different components. The hardware itself is initialized by the BIOS, which starts the kernel by means of a boot loader. After this point, the boot process with init and the runlevels is completely controlled by the operating system. The runlevel concept enables you to maintain setups for everyday usage as well as to perform maintenance tasks on the system.

12.1 The Linux Boot Process

The Linux boot process consists of several stages each represented by another component. The following list briefly summarizes the boot process and features all the major components involved.

1. **BIOS** After the computer has been turned on, the BIOS initializes the screen and keyboard and tests the main memory. Up to this stage, the machine does not access any mass storage media. Subsequently, the information about the current date, time, and the most important peripherals are loaded from the CMOS values. When the first hard disk and its geometry are recognized, the system control passes from the BIOS to the boot loader.

2. **Boot Loader** The first physical 512-byte data sector of the first hard disk is loaded into the main memory and the *boot loader* that resides at the beginning of this sector takes over. The commands executed by the boot loader determine the remaining part of the boot process. Therefore, the first 512 bytes on the first hard disk are referred to as the *Master Boot Record* (MBR). The boot loader then passes control to the actual operating system, in this case, the Linux kernel. More

information about GRUB, the Linux boot loader, can be found in Chapter 13, *The Boot Loader* (page 195).

3. **Kernel and initramfs** To pass system control, the boot loader loads both the kernel and an initial RAM–based file system (initramfs) into memory. The contents of the initramfs can be used by the kernel directly. initramfs contains a small executable called init that handles the mounting of the real root file system. In former versions of SUSE® Linux, these tasks were handled by initrd and linuxrc, respectively. For more information about initramfs, refer to Section 12.1.1, "initramfs" (page 180).

4. **init on initramfs** This program performs all actions needed to mount the proper root file system, like providing kernel functionality for the needed file system and device drivers for mass storage controllers with udev. After the root file system has been found, it is checked for errors and mounted. If this has been successful, the initramfs is cleaned and the init program on the root file system is executed. For more information about init, refer to Section 12.1.2, "init on initramfs" (page 181). Find more information about udev in Chapter 15, *Dynamic Kernel Device Management with udev* (page 227).

5. **init** init handles the actual booting of the system through several different levels providing different functionality. init is described in Section 12.2, "The init Process" (page 183).

12.1.1 initramfs

initramfs is a small cpio archive that the kernel can load to a RAM disk. It provides a minimal Linux environment that enables the execution of programs before the actual root file system is mounted. This minimal Linux environment is loaded into memory by BIOS routines and does not have specific hardware requirements other than sufficient memory. initramfs must always provide an executable named init that should execute the actual init program on the root file system for the boot process to proceed.

Before the root file system can be mounted and the operating system can be started, the kernel needs the corresponding drivers to access the device on which the root file system is located. These drivers may include special drivers for certain kinds of hard drives or even network drivers to access a network file system. The needed modules for the root file system may be loaded by init on initramfs. After the modules are loaded, udev provides the initramfs with the needed devices. Later in the boot process, after changing the root file system, it is necessary to regenerate the devices. This is done by boot.udev with the command udevtrigger.

If you need to change hardware (e.g. hard disks) in an installed system and this hardware requires different drivers to be present in the kernel at boot time, you must update initramfs. This is done in the same way as with its predecessor, initrd—by calling mkinitrd. Calling mkinitrd without any argument creates an initramfs. Calling mkinitrd -R creates an initrd. In openSUSE®, the modules to load are specified by the variable INITRD_MODULES in /etc/sysconfig/kernel. After installation, this variable is automatically set to the correct value. The modules are loaded in exactly the order in which they appear in INITRD_MODULES. This is only important if you rely on the correct setting of the device files /dev/sd?. However, in current systems you also may use the device files below /dev/disk/ that are sorted in several subdirectories, named by-id, by-path and by-uuid, and always represent the same disk. This is also possible at install time by specifying the respective mount option.

IMPORTANT: Updating initramfs or initrd

The boot loader loads initramfs or initrd in the same way as the kernel. It is not necessary to reinstall GRUB after updating initramfs or initrd, because GRUB searches the directory for the right file when booting.

12.1.2 init on initramfs

The main purpose of init on initramfs is to prepare the mounting of and access to the real root file system. Depending on your system configuration, init is responsible for the following tasks.

Loading Kernel Modules

Depending on your hardware configuration, special drivers may be needed to access the hardware components of your computer (the most important component being your hard drive). To access the final root file system, the kernel needs to load the proper file system drivers.

Providing Device Special Files

For each loaded module, the kernel generates device events. udev handles these events and generates the required device special files on a RAM file system in

`/dev`. Without those special files, the file system and other devices would not be accessible.

Managing RAID and LVM Setups

If you configured your system to hold the root file system under RAID or LVM, init sets up LVM or RAID to enable access to the root file system later. Find information about RAID in Section 2.3, "Soft RAID Configuration" (page 55). Find information about LVM in Section 2.2, "LVM Configuration" (page 49).

Managing Network Configuration

If you configured your system to use a network-mounted root file system (mounted via NFS), init must make sure that the proper network drivers are loaded and that they are set up to allow access to the root file system.

When init is called during the initial boot as part of the installation process, its tasks differ from those mentioned earlier:

Finding the Installation Medium

As you start the installation process, your machine loads an installation kernel and a special initrd with the YaST installer from the installation medium. The YaST installer, which is run in a RAM file system, needs to have information about the location of the installation medium to access it and install the operating system.

Initiating Hardware Recognition and Loading Appropriate Kernel Modules

As mentioned in Section 12.1.1, "initramfs" (page 180), the boot process starts with a minimum set of drivers that can be used with most hardware configurations. init starts an initial hardware scanning process that determines the set of drivers suitable for your hardware configuration. The names of the modules needed for the boot process are written to `INITRD_MODULES` in `/etc/sysconfig/kernel`. These names are used to generate a custom initramfs that is needed to boot the system. If the modules are not needed for boot but for coldplug, the modules are written to `/etc/sysconfig/hardware/hwconfig-*`. All devices that are described with configuration files in this directory are initialized in the boot process.

Loading the Installation System or Rescue System

As soon as the hardware has been properly recognized, the appropriate drivers have been loaded, and udev has created the device special files, init starts the installation system, which contains the actual YaST installer, or the rescue system.

Starting YaST

Finally, init starts YaST, which starts package installation and system configuration.

12.2 The init Process

The program init is the process with process ID 1. It is responsible for initializing the system in the required way. init is started directly by the kernel and resists signal 9, which normally kills processes. All other programs are either started directly by init or by one of its child processes.

init is centrally configured in the `/etc/inittab` file where the *runlevels* are defined (see Section 12.2.1, “Runlevels” (page 183)). The file also specifies which services and daemons are available in each of the runlevels. Depending on the entries in `/etc/inittab`, several scripts are run by init. By default, the first script that is started after booting is `/etc/init.d/boot`. Once the system initialization phase is finished, the system changes the runlevel to its default runlevel with the `/etc/init.d/rc` script. For reasons of clarity, these scripts, called *init scripts*, all reside in the directory `/etc/init.d` (see Section 12.2.2, “Init Scripts” (page 186)).

The entire process of starting the system and shutting it down is maintained by init. From this point of view, the kernel can be considered a background process whose task is to maintain all other processes and adjust CPU time and hardware access according to requests from other programs.

12.2.1 Runlevels

In Linux, *runlevels* define how the system is started and what services are available in the running system. After booting, the system starts as defined in `/etc/inittab` in the line `initdefault`. Usually this is `3` or `5`. See Table 12.1, “Available Runlevels” (page 183). As an alternative, the runlevel can be specified at boot time (at the boot prompt, for instance). Any parameters that are not directly evaluated by the kernel itself are passed to init. To boot into runlevel 3, just add a the single number 3 to the boot prompt.

***Table 12.1** Available Runlevels*

Runlevel	Description
0	System halt
S or 1	Single user mode

Runlevel	Description
2	Local multiuser mode without remote network (NFS, etc.)
3	Full multiuser mode with network
4	Not used
5	Full multiuser mode with network and X display manager—KDM, GDM, or XDM
6	System reboot

IMPORTANT: Avoid Runlevel 2 with a Partition Mounted via NFS

You should not use runlevel 2 if your system mounts a partition like `/usr` via NFS. The system might behave unexpectedly if program files or libraries are missing because the NFS service is not available in runlevel 2 (local multiuser mode without remote network).

To change runlevels while the system is running, enter `telinit` and the corresponding number as an argument. Only the system administrator is allowed to do this. The following list summarizes the most important commands in the runlevel area.

`telinit 1` or `shutdown now`
: The system changes to *single user mode*. This mode is used for system maintenance and administration tasks.

`telinit 3`
: All essential programs and services (including network) are started and regular users are allowed to log in and work with the system without a graphical environment.

`telinit 5`
: The graphical environment is enabled. Usually a display manager like XDM, GDM, or KDM is started. If autologin is enabled, the local user is logged in to the preselected window manager (GNOME or KDE or any other window manager).

`telinit 0` or `shutdown -h now`
The system halts.

`telinit 6` or `shutdown -r now`
The system halts then reboots.

Runlevel `5` is the default runlevel in all openSUSE standard installations. Users are prompted for login with a graphical interface or the default user is logged in automatically. If the default runlevel is `3`, the X Window System must be configured properly, as described in Chapter 8, *The X Window System* (page 117), before the runlevel can be switched to `5`. If this is done, check whether the system works in the desired way by entering `telinit 5`. If everything turns out as expected, you can use YaST to set the default runlevel to `5`.

WARNING: Errors in /etc/inittab May Result in a Faulty System Boot

If `/etc/inittab` is damaged, the system might not boot properly. Therefore, be extremely careful while editing `/etc/inittab`. Always let init reread `/etc/inittab` with the command `telinit q` before rebooting the machine.

Generally, two things happen when you change runlevels. First, stop scripts of the current runlevel are launched, closing down some programs essential for the current runlevel. Then start scripts of the new runlevel are started. Here, in most cases, a number of programs are started. For example, the following occurs when changing from runlevel 3 to 5:

1. The administrator (`root`) requests init to change to a different runlevel by entering `telinit 5`.

2. init consults its configuration file (`/etc/inittab`) and determines it should start `/etc/init.d/rc` with the new runlevel as a parameter.

3. Now `rc` calls the stop scripts of the current runlevel for which there is no start script in the new runlevel. In this example, these are all the scripts that reside in `/etc/init.d/rc3.d` (old runlevel was 3) and start with a `K`. The number following `K` specifies the order to start, because there are some dependencies to consider.

4. The last things to start are the start scripts of the new runlevel. These are, in this example, in `/etc/init.d/rc5.d` and begin with an `S`. The same procedure regarding the order in which they are started is applied here.

When changing into the same runlevel as the current runlevel, init only checks `/etc/inittab` for changes and starts the appropriate steps, for example, for starting a `getty` on another interface. The same functionality may be achieved with the command `telinit q`.

12.2.2 Init Scripts

There are two types of scripts in `/etc/init.d`:

Scripts Executed Directly by init
: This is the case only during the boot process or if an immediate system shutdown is initiated (power failure or a user pressing Ctrl + Alt + Del). The execution of these scripts is defined in `/etc/inittab`.

Scripts Executed Indirectly by init
: These are run when changing the runlevel and always call the master script `/etc/init.d/rc`, which guarantees the correct order of the relevant scripts.

All scripts are located in `/etc/init.d`. Scripts that are run at boot time are called through symbolic links from `/etc/init.d/boot.d`. Scripts for changing the runlevel are called through symbolic links from one of the subdirectories (`/etc/init.d/rc0.d` to `/etc/init.d/rc6.d`). This is just for clarity reasons and avoids duplicate scripts if they are used in several runlevels. Because every script can be executed as both a start and a stop script, these scripts must understand the parameters `start` and `stop`. The scripts also understand the `restart`, `reload`, `force-reload`, and `status` options. These different options are explained in Table 12.2, "Possible init Script Options" (page 186). Scripts that are run directly by init do not have these links. They are run independently from the runlevel when needed.

***Table 12.2** Possible init Script Options*

Option	Description
`start`	Start service.

Option	Description
`stop`	Stop service.
`restart`	If the service is running, stop it then restart it. If it is not running, start it.
`reload`	Reload the configuration without stopping and restarting the service.
`force-reload`	Reload the configuration if the service supports this. Otherwise, do the same as if `restart` had been given.
`status`	Show the current status of service.

Links in each runlevel-specific subdirectory make it possible to associate scripts with different runlevels. When installing or uninstalling packages, these links are added and removed with the help of the program insserv (or using `/usr/lib/lsb/install_initd`, which is a script calling this program). See the insserv(8) man page for details.

All of these settings may also be changed with the help of the YaST module. If you need to check the status on the command line, use the tool chkconfig, described in the chkconfig(8) man page.

A short introduction to the boot and stop scripts launched first or last, respectively, follows as well as an explanation of the maintaining script.

`boot`

Executed while starting the system directly using init. It is independent of the chosen runlevel and is only executed once. Here, the `/proc` and `/dev/pts` file systems are mounted and blogd (boot logging daemon) is activated. If the system is booted for the first time after an update or an installation, the initial system configuration is started.

The blogd daemon is a service started by boot and rc before any other one. It is stopped after the actions triggered by these scripts (running a number of subscripts, for example) are completed. blogd writes any screen output to the log file `/var/log/boot.msg`, but only if and when `/var` is mounted read-write. Otherwise, blogd buffers all screen data until `/var` becomes available. Get further information about blogd on the blogd(8) man page.

The script `boot` is also responsible for starting all the scripts in `/etc/init.d/boot.d` with a name that starts with `S`. There, the file systems are checked and loop devices are configured if needed. The system time is also set. If an error occurs while automatically checking and repairing the file system, the system administrator can intervene after first entering the root password. Last executed is the script `boot.local`.

`boot.local`
: Here, enter additional commands to execute at boot before changing into a runlevel. It can be compared to `AUTOEXEC.BAT` on DOS systems.

`boot.setup`
: This script is executed when changing from single user mode to any other runlevel and is responsible for a number of basic settings, such as the keyboard layout and initialization of the virtual consoles.

`halt`
: This script is only executed while changing into runlevel 0 or 6. Here, it is executed either as `halt` or as `reboot`. Whether the system shuts down or reboots depends on how `halt` is called.

`rc`
: This script calls the appropriate stop scripts of the current runlevel and the start scripts of the newly selected runlevel.

You can create your own scripts and easily integrate them into the scheme described above. For instructions about formatting, naming, and organizing custom scripts, refer to the specifications of the LSB and to the man pages of `init`, `init.d`, `chkconfig`, and `insserv`. Additionally consult the man pages of `startproc` and `killproc`.

WARNING: Faulty init Scripts May Halt Your System

Faulty init scripts may hang your machine. Edit such scripts with great care and, if possible, subject them to heavy testing in the multiuser environment. Find some useful information about init scripts in Section 12.2.1, “Runlevels” (page 183).

To create a custom init script for a given program or service, use the file /etc/init.d/skeleton as a template. Save a copy of this file under the new name and edit the relevant program and filenames, paths, and other details as needed. You may also need to enhance the script with your own parts, so the correct actions are triggered by the init procedure.

The INIT INFO block at the top is a required part of the script and must be edited. See Example 12.1, “A Minimal INIT INFO Block” (page 189).

Example 12.1 *A Minimal INIT INFO Block*

```
### BEGIN INIT INFO
# Provides:          FOO
# Required-Start:    $syslog $remote_fs
# Required-Stop:     $syslog $remote_fs
# Default-Start:     3 5
# Default-Stop:      0 1 2 6
# Description:       Start FOO to allow XY and provide YZ
### END INIT INFO
```

In the first line of the INFO block, after Provides:, specify the name of the program or service controlled by this init script. In the Required-Start: and Required-Stop: lines, specify all services that need to be started or stopped before the service itself is started or stopped. This information is used later to generate the numbering of script names, as found in the runlevel directories. After Default-Start: and Default-Stop:, specify the runlevels in which the service should automatically be started or stopped. Finally, for Description:, provide a short description of the service in question.

To create the links from the runlevel directories (/etc/init.d/rc?.d/) to the corresponding scripts in /etc/init.d/, enter the command insserv *new-script-name*. The insserv program evaluates the INIT INFO header to create the necessary links for start and stop scripts in the runlevel directories (/etc/init.d/rc?.d/). The program also takes care of the correct start and stop order for each runlevel by including the necessary numbers in the names of these links. If you prefer a graphical tool to create such links, use the runlevel editor provided by YaST, as described in Section 12.2.3, "Configuring System Services (Runlevel) with YaST" (page 190).

If a script already present in `/etc/init.d/` should be integrated into the existing runlevel scheme, create the links in the runlevel directories right away with insserv or by enabling the corresponding service in the runlevel editor of YaST. Your changes are applied during the next reboot—the new service is started automatically.

Do not set these links manually. If something is wrong in the `INFO` block, problems will arise when `insserv` is run later for some other service. The manually-added service will be removed with the next run of `insserv` for this script.

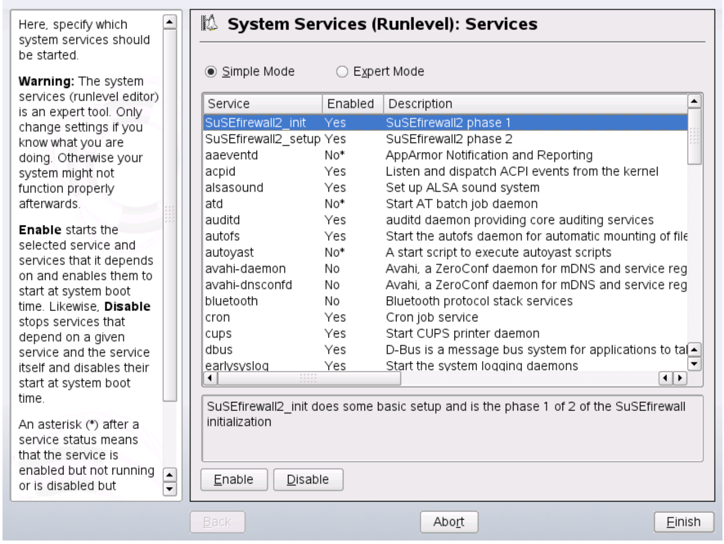
12.2.3 Configuring System Services (Runlevel) with YaST

After starting this YaST module with *YaST* > *System* > *System Services (Runlevel)*, it displays an overview listing all the available services and the current status of each service (disabled or enabled). Decide whether to use the module in *Simple Mode* or in *Expert Mode*. The default *Simple Mode* should be sufficient for most purposes. The left column shows the name of the service, the center column indicates its current status, and the right column gives a short description. For the selected service, a more detailed description is provided in the lower part of the window. To enable a service, select it in the table then select *Enable*. The same steps apply to disable a service.

For detailed control over the runlevels in which a service is started or stopped or to change the default runlevel, first select *Expert Mode*. The current default runlevel or "initdefault" (the runlevel into which the system boots by default) is displayed at the top. Normally, the default runlevel of a openSUSE system is runlevel 5 (full multiuser mode with network and X). A suitable alternative might be runlevel 3 (full multiuser mode with network).

This YaST dialog allows the selection of one of the runlevels (as listed in Table 12.1, "Available Runlevels" (page 183)) as the new default. Additionally use the table in this window to enable or disable individual services and daemons. The table lists the services and daemons available, shows whether they are currently enabled on your system, and, if so, for which runlevels. After selecting one of the rows with the mouse, click the check boxes representing the runlevels (*B*, *0*, *1*, *2*, *3*, *5*, *6*, and *S*) to define the runlevels in which the selected service or daemon should be running. Runlevel 4 is undefined to allow creation of a custom runlevel. A brief description of the currently selected service or daemon is provided below the table overview.

Figure 12.1 *System Services (Runlevel)*



Here, specify which system services should be started.

Warning: The system services (runlevel editor) is an expert tool. Only change settings if you know what you are doing. Otherwise your system might not function properly afterwards.

Enable starts the selected service and services that it depends on and enables them to start at system boot time. Likewise, **Disable** stops services that depend on a given service and the service itself and disables their start at system boot time.

An asterisk (*) after a service status means that the service is enabled but not running or is disabled but

System Services (Runlevel): Services

● Simple Mode ○ Expert Mode

Service	Enabled	Description
SuSEfirewall2_init	Yes	SuSEfirewall2 phase 1
SuSEfirewall2_setup	Yes	SuSEfirewall2 phase 2
aaeventd	No*	AppArmor Notification and Reporting
acpid	Yes	Listen and dispatch ACPI events from the kernel
alsasound	Yes	Set up ALSA sound system
atd	No*	Start AT batch job daemon
auditd	Yes	auditd daemon providing core auditing services
autofs	Yes	Start the autofs daemon for automatic mounting of file
autoyast	No*	A start script to execute autoyast scripts
avahi-daemon	No	Avahi, a ZeroConf daemon for mDNS and service reg
avahi-dnsconfd	No	Avahi, a ZeroConf daemon for mDNS and service reg
bluetooth	No	Bluetooth protocol stack services
cron	Yes	Cron job service
cups	Yes	Start CUPS printer daemon
dbus	Yes	D-Bus is a message bus system for applications to tal
earlysyslog	Yes	Start the system logging daemons

SuSEfirewall2_init does some basic setup and is the phase 1 of 2 of the SuSEfirewall initialization

With *Start, Stop, or Refresh*, decide whether a service should be activated. *Refresh status* checks the current status. *Set or Reset* lets you select whether to apply your changes to the system or to restore the settings that existed before starting the runlevel editor. Selecting *Finish* saves the changed settings to disk.

WARNING: Faulty Runlevel Settings May Damage Your System

Faulty runlevel settings may render a system unusable. Before applying your changes, make absolutely sure that you know their consequences.

12.3 System Configuration via /etc/sysconfig

The main configuration of openSUSE is controlled by the configuration files in `/etc/sysconfig`. The individual files in `/etc/sysconfig` are only read by the scripts to which they are relevant. This ensures that network settings, for example, only need to be parsed by network-related scripts.

There are two ways to edit the system configuration. Either use the YaST sysconfig Editor or edit the configuration files manually.

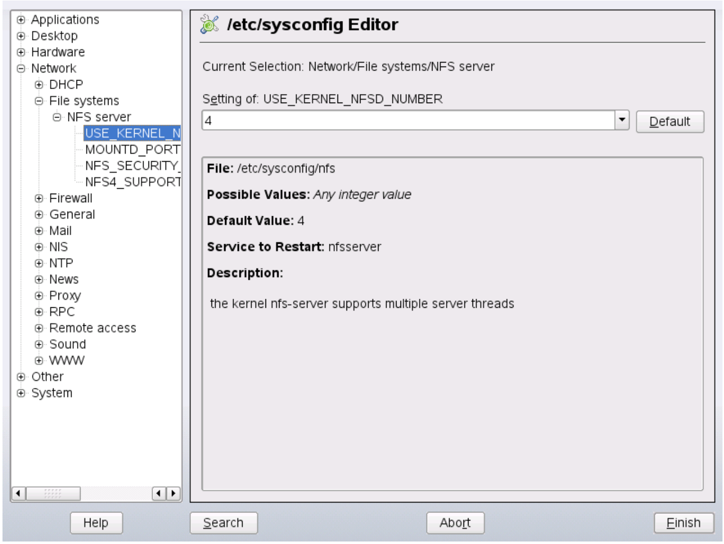
12.3.1 Changing the System Configuration Using the YaST sysconfig Editor

The YaST sysconfig editor provides an easy-to-use front-end to system configuration. Without any knowledge of the actual location of the configuration variable you need to change, you can just use the built-in search function of this module, change the value of the configuration variable as needed, and let YaST take care of applying these changes, updating configurations that depend on the values set in `sysconfig` and restarting services.

WARNING: Modifying `/etc/sysconfig/*` Files Can Damage Your Installation

Do not modify the `/etc/sysconfig` files if you lack previous experience and knowledge. It could do considerable damage to your system. The files in `/etc/sysconfig` include a short comment for each variable to explain what effect they actually have.

Figure 12.2 *System Configuration Using the sysconfig Editor*



/etc/sysconfig Editor

Current Selection: Network/File systems/NFS server

Setting of: USE_KERNEL_NFSD_NUMBER

4

File: /etc/sysconfig/nfs

Possible Values: *Any integer value*

Default Value: 4

Service to Restart: nfsserver

Description:

the kernel nfs-server supports multiple server threads

The YaST sysconfig dialog is split into three parts. The left part of the dialog shows a tree view of all configurable variables. When you select a variable, the right part displays both the current selection and the current setting of this variable. Below, a third window displays a short description of the variable's purpose, possible values, the default value, and the actual configuration file from which this variable originates. The dialog also provides information about which configuration script is executed after changing the variable and which new service is started as a result of the change. YaST prompts you to confirm your changes and informs you which scripts will be executed after you leave the dialog by selecting *Finish*. Also select the services and scripts to skip for now, so they are started later. YaST applies all changes automatically and restarts any services involved for your changes to take an effect.

12.3.2 Changing the System Configuration Manually

To manually change the system configuration, proceed as follows

1 Become `root`.

2 Bring the system into single user mode (runlevel 1) with `telinit 1`.

3 Change the configuration files as needed with an editor of your choice.

   If you do not use YaST to change the configuration files in `/etc/sysconfig`, make sure that empty variable values are represented by two quotation marks (`KEYTABLE=""`) and that values with blanks in them are enclosed in quotation marks. Values consisting of one word only do not need to be quoted.

4 Execute `SuSEconfig` to make sure that the changes take effect.

5 Bring your system back to the previous runlevel with a command like `telinit` *`default_runlevel`*. Replace *`default_runlevel`* with the default runlevel of the system. Choose `5` if you want to return to full multiuser with network and X or choose `3` if you prefer to work in full multiuser with network.

This procedure is mainly relevant when changing systemwide settings, such as the network configuration. Small changes should not require going into single user mode, but you may still do so to make absolutely sure that all the programs concerned are correctly restarted.

TIP: Configuring Automated System Configuration

To disable the automated system configuration by SuSEconfig, set the variable `ENABLE_SUSECONFIG` in `/etc/sysconfig/suseconfig` to `no`. Do not disable SuSEconfig if you want to use the SUSE installation support. It is also possible to disable the autoconfiguration partially.

The Boot Loader 13

This chapter describes how to configure GRUB, the boot loader used in openSUSE®. A special YaST module is available for performing all settings. If you are not familiar with the subject of booting in Linux, read the following sections to acquire some background information. This chapter also describes some of the problems frequently encountered when booting with GRUB and their solutions.

This chapter focuses on boot management and the configuration of the boot loader GRUB. The boot procedure as a whole is outlined in Chapter 12, *Booting and Configuring a Linux System* (page 179). A boot loader represents the interface between machine (BIOS) and the operating system (openSUSE). The configuration of the boot loader directly impacts the start of the operating system.

The following terms appear frequently in this chapter and might need some explanation:

Master Boot Record

The structure of the MBR is defined by an operating system–independent convention. The first 446 bytes are reserved for the program code. They typically hold part of a boot loader program or an operating system selector. The next 64 bytes provide space for a partition table of up to four entries (see Section 2.1.1, "Partition Types" (page 42)). The partition table contains information about the partitioning of the hard disk and the file system types. The operating system needs this table for handling the hard disk. With conventional generic code in the MBR, exactly one partition must be marked *active*. The last two bytes of the MBR must contain a static "magic number" (AA55). An MBR containing a different value is regarded as invalid by some BIOSs, so is not considered for booting.

Boot Sectors

Boot sectors are the first sectors of hard disk partitions with the exception of the extended partition, which merely serves as a "container" for other partitions. These boot sectors have 512 bytes of space for code used to boot an operating system installed in the respective partition. This applies to boot sectors of formatted DOS, Windows, and OS/2 partitions, which also contain some important basic data of the file system. In contrast, the boot sectors of Linux partitions are initially empty after setting up a file system other than XFS. Therefore, a Linux partition is not bootable by itself, even if it contains a kernel and a valid root file system. A boot sector with valid code for booting the system has the same magic number as the MBR in its last two bytes (`AA55`).

13.1 Selecting a Boot Loader

By default, the boot loader GRUB is used in openSUSE. However, in some cases and for special hardware and software constellations, LILO may be necessary. If you update from an older openSUSE version that uses LILO, LILO is installed.

Information about the installation and configuration of LILO is available in the Support Database under the keyword LILO and in `/usr/share/doc/packages/lilo`.

13.2 Booting with GRUB

GRUB (Grand Unified Bootloader) comprises two stages. stage1 consists of 512 bytes and its only task is to load the second stage of the boot loader. Subsequently, stage2 is loaded. This stage contains the main part of the boot loader.

In some configurations, an intermediate stage 1.5 can be used, which locates and loads stage 2 from an appropriate file system. If possible, this method is chosen by default on installation or when initially setting up GRUB with YaST.

stage2 is able to access many file systems. Currently, Ext2, Ext3, ReiserFS, Minix, and the DOS FAT file system used by Windows are supported. To a certain extent, XFS, and UFS and FFS used by BSD systems are also supported. Since version 0.95, GRUB is also able to boot from a CD or DVD containing an ISO 9660 standard file system pursuant to the "El Torito" specification. Even before the system is booted, GRUB can access file systems of supported BIOS disk devices (floppy disks or hard disks, CD drives, and DVD drives detected by the BIOS). Therefore, changes to the GRUB configuration file (`menu.lst`) do not require a reinstallation of the boot manager. When the system is booted, GRUB reloads the menu file with the valid paths and partition data of the kernel or the initial RAM disk (`initrd`) and locates these files.

The actual configuration of GRUB is based on three files that are described below:

`/boot/grub/menu.lst`
: This file contains all information about partitions or operating systems that can be booted with GRUB. Without this information, the GRUB command line prompts the user for how to proceed (see Section “Editing Menu Entries during the Boot Procedure” (page 201) for details).

`/boot/grub/device.map`
: This file translates device names from the GRUB and BIOS notation to Linux device names.

`/etc/grub.conf`
: This file contains the commands, parameters, and options the GRUB shell needs for installing the boot loader correctly.

GRUB can be controlled in various ways. Boot entries from an existing configuration can be selected from the graphical menu (splash screen). The configuration is loaded from the file `menu.lst`.

In GRUB, all boot parameters can be changed prior to booting. For example, errors made when editing the menu file can be corrected in this way. Boot commands can also be entered interactively at a kind of input prompt (see Section “Editing Menu Entries during the Boot Procedure” (page 201)). GRUB offers the possibility of determining the location of the kernel and the `initrd` prior to booting. In this way, you can even boot an installed operating system for which no entry exists in the boot loader configuration.

GRUB actually exists in two versions: as a boot loader and as a normal Linux program in `/usr/sbin/grub`. This program is referred to as the *GRUB shell*. It provides an emulation of GRUB in the installed system and can be used to install GRUB or test new settings before applying them. The functionality to install GRUB as the boot loader on a hard disk or floppy disk is integrated in GRUB in the form of the commands `install` and `setup`. This is available in the GRUB shell when Linux is loaded.

13.2.1 The GRUB Boot Menu

The graphical splash screen with the boot menu is based on the GRUB configuration file /boot/grub/menu.lst, which contains all information about all partitions or operating systems that can be booted by the menu.

Every time the system is booted, GRUB loads the menu file from the file system. For this reason, GRUB does not need to be reinstalled after every change to the file. Use the YaST boot loader to modify the GRUB configuration as described in Section 13.3, “Configuring the Boot Loader with YaST” (page 205).

The menu file contains commands. The syntax is very simple. Every line contains a command followed by optional parameters separated by spaces like in the shell. For historical reasons, some commands permit an = in front of the first parameter. Comments are introduced by a hash (#).

To identify the menu items in the menu overview, set a title for every entry. The text (including any spaces) following the keyword title is displayed as a selectable option in the menu. All commands up to the next title are executed when this menu item is selected.

The simplest case is the redirection to boot loaders of other operating systems. The command is chainloader and the argument is usually the boot block of another partition, in GRUB block notation. For example:

```
chainloader (hd0,3)+1
```

The device names in GRUB are explained in Section “Naming Conventions for Hard Disks and Partitions” (page 199). This example specifies the first block of the fourth partition of the first hard disk.

Use the command kernel to specify a kernel image. The first argument is the path to the kernel image in a partition. The other arguments are passed to the kernel on its command line.

If the kernel does not have built-in drivers for access to the root partition or a recent Linux system with advanced hotplug features is used, initrd must be specified with a separate GRUB command whose only argument is the path to the initrd file. Because the loading address of the initrd is written into the loaded kernel image, the command initrd must follow after the kernel command.

The command root simplifies the specification of kernel and initrd files. The only argument of root is a device or a partition. This device is used for all kernel, initrd, or other file paths for which no device is explicitly specified until the next root command.

The boot command is implied at the end of every menu entry, so it does not need to be written into the menu file. However, if you use GRUB interactively for booting, you must enter the boot command at the end. The command itself has no arguments. It merely boots the loaded kernel image or the specified chain loader.

After writing all menu entries, define one of them as the default entry. Otherwise, the first one (entry 0) is used. You can also specify a time-out in seconds after which the default entry should boot. timeout and default usually precede the menu entries. An example file is described in Section "An Example Menu File" (page 200).

Naming Conventions for Hard Disks and Partitions

The naming conventions GRUB uses for hard disks and partitions differ from those used for normal Linux devices. It more closely resembles the simple disk enumeration the BIOS does and the syntax is similar to that used in some BSD derivatives. In GRUB, the numbering of the partitions starts with zero. This means that (hd0,0) is the first partition of the first hard disk. On a common desktop machine with a hard disk connected as primary master, the corresponding Linux device name is /dev/sda1.

The four possible primary partitions are assigned the partition numbers 0 to 3. The logical partitions are numbered from 4:

```
(hd0,0)   first primary partition of the first hard disk
(hd0,1)   second primary partition
(hd0,2)   third primary partition
(hd0,3)   fourth primary partition (usually an extended partition)
(hd0,4)   first logical partition
(hd0,5)   second logical partition
```

Being dependent on BIOS devices, GRUB does not distinguish between IDE, SATA, SCSI, and hardware RAID devices. All hard disks recognized by the BIOS or other controllers are numbered according to the boot sequence preset in the BIOS.

Unfortunately, it is often not possible to map the Linux device names to BIOS device names exactly. It generates this mapping with the help of an algorithm and saves it to the file device.map, which can be edited if necessary. Information about the file device.map is available in Section 13.2.2, "The File device.map" (page 202).

A complete GRUB path consists of a device name written in parentheses and the path to the file in the file system in the specified partition. The path begins with a slash. For example, the bootable kernel could be specified as follows on a system with a single IDE hard disk containing Linux in its first partition:

```
(hd0,0)/boot/vmlinuz
```

An Example Menu File

The following example shows the structure of a GRUB menu file. The example installation has a Linux boot partition under `/dev/sda5`, a root partition under `/dev/sda7`, and a Windows installation under `/dev/sda1`.

```
gfxmenu (hd0,4)/message
color white/blue black/light-gray
default 0
timeout 8

title linux
   kernel (hd0,4)/vmlinuz root=/dev/sda7 vga=791
   initrd (hd0,4)/initrd

title windows
   chainloader(hd0,0)+1

title floppy
   chainloader(fd0)+1

title failsafe
   kernel (hd0,4)/vmlinuz.shipped root=/dev/sda7 ide=nodma \
   apm=off acpi=off vga=normal nosmp maxcpus=0 3
   initrd (hd0,4)/initrd.shipped
```

The first block defines the configuration of the splash screen:

gfxmenu (hd0,4)/message
: The background image `message` is located in the top directory of the `/dev/sda5` partition.

color white/blue black/light-gray
: Color scheme: white (foreground), blue (background), black (selection), and light gray (background of the selection). The color scheme has no effect on the splash screen, only on the customizable GRUB menu that you can access by exiting the splash screen with Esc.

default 0
: The first menu entry `title linux` is the one to boot by default.

timeout 8
: After eight seconds without any user input, GRUB automatically boots the default entry. To deactivate automatic boot, delete the `timeout` line. If you set `timeout 0`, GRUB boots the default entry immediately.

The second and largest block lists the various bootable operating systems. The sections for the individual operating systems are introduced by `title`.

- The first entry (`title linux`) is responsible for booting openSUSE. The kernel (`vmlinuz`) is located in the first logical partition (the boot partition) of the first hard disk. Kernel parameters, such as the root partition and VGA mode, are appended here. The root partition is specified according to the Linux naming convention (`/dev/sda7/`), because this information is read by the kernel and has nothing to do with GRUB. The `initrd` is also located in the first logical partition of the first hard disk.

- The second entry is responsible for loading Windows. Windows is booted from the first partition of the first hard disk (`hd0,0`). The command `chainloader +1` causes GRUB to read and execute the first sector of the specified partition.

- The next entry enables booting from floppy disk without modifying the BIOS settings.

- The boot option `failsafe` starts Linux with a selection of kernel parameters that enables Linux to boot even on problematic systems.

The menu file can be changed whenever necessary. GRUB then uses the modified settings during the next boot. Edit the file permanently using YaST or an editor of your choice. Alternatively, make temporary changes interactively using the edit function of GRUB. See Section “Editing Menu Entries during the Boot Procedure” (page 201).

Editing Menu Entries during the Boot Procedure

In the graphical boot menu, select the operating system to boot with the arrow keys. If you select a Linux system, you can enter additional boot parameters at the boot prompt. To edit individual menu entries directly, press Esc to exit the splash screen and get to the GRUB text-based menu then press E. Changes made in this way only apply to the current boot and are not adopted permanently.

IMPORTANT: Keyboard Layout during the Boot Procedure

The US keyboard layout is the only one available when booting. See Figure "US Keyboard Layout" (↑Start-Up) for a figure.

Editing menu entries facilitates the repair of a defective system that can no longer be booted, because the faulty configuration file of the boot loader can be circumvented by manually entering parameters. Manually entering parameters during the boot procedure is also useful for testing new settings without impairing the native system.

After activating the editing mode, use the arrow keys to select the menu entry of the configuration to edit. To make the configuration editable, press E again. In this way, edit incorrect partitions or path specifications before they have a negative effect on the boot process. Press Enter to exit the editing mode and return to the menu. Then press B to boot this entry. Further possible actions are displayed in the help text at the bottom.

To enter changed boot options permanently and pass them to the kernel, open the file `menu.lst` as the user `root` and append the respective kernel parameters to the existing line, separated by spaces:

```
title linux
   kernel (hd0,0)/vmlinuz root=/dev/sda3 additional parameter
   initrd (hd0,0)/initrd
```

GRUB automatically adopts the new parameters the next time the system is booted. Alternatively, this change can also be made with the YaST boot loader module. Append the new parameters to the existing line, separated by spaces.

13.2.2 The File device.map

The file `device.map` maps GRUB and BIOS device names to Linux device names. In a mixed system containing IDE and SCSI hard disks, GRUB must try to determine the boot sequence by a special procedure, because GRUB may not have access to the BIOS information on the boot sequence. GRUB saves the result of this analysis in the file `/boot/grub/device.map`. For a system on which the boot sequence in the BIOS is set to IDE before SCSI, the file `device.map` could appear as follows:

```
(fd0)  /dev/fd0
(hd0)  /dev/sda
(hd1)  /dev/sdb
```

Because the order of IDE, SCSI, and other hard disks depends on various factors and Linux is not able to identify the mapping, the sequence in the file `device.map` can be set manually. If you encounter problems when booting, check if the sequence in this file corresponds to the sequence in the BIOS and use the GRUB prompt to modify it temporarily if necessary. After the Linux system has booted, the file `device.map` can be edited permanently with the YaST boot loader module or an editor of your choice.

After manually changing `device.map`, execute the following command to reinstall GRUB. This command causes the file `device.map` to be reloaded and the commands listed in `grub.conf` to be executed:

```
grub --batch < /etc/grub.conf
```

13.2.3 The File /etc/grub.conf

The third most important GRUB configuration file after `menu.lst` and `device.map` is `/etc/grub.conf`. This file contains the commands, parameters, and options the GRUB shell needs for installing the boot loader correctly:

```
root (hd0,4)
   install /grub/stage1 (hd0,3) /grub/stage2 0x8000 (hd0,4)/grub/menu.lst
   quit
```

Meaning of the individual entries:

root (hd0,4)
: This command tells GRUB to apply the following commands to the first logical partition of the first hard disk (the location of the boot files).

install `parameter`
: The command `grub` should be run with the parameter `install`. `stage1` of the boot loader should be installed in the the extended partition container (`/grub/stage1 (hd0,3)`). This is a slightly esoteric configuration, but it is known to work in many cases. `stage2` should be loaded to the memory address 0x8000 (`/grub/stage2 0x8000`). The last entry (`(hd0,4)/grub/menu.lst`) tells GRUB where to look for the menu file.

13.2.4 Setting a Boot Password

Even before the operating system is booted, GRUB enables access to file systems. Users without root permissions can access files in your Linux system to which they have no access once the system is booted. To block this kind of access or prevent users from booting certain operating systems, set a boot password.

IMPORTANT: Boot Password and Splash Screen

If you use a boot password for GRUB, the usual splash screen is not displayed.

As the user `root`, proceed as follows to set a boot password:

1. At the root prompt, encrypt the password using grub-md5-crypt:

   ```
   # grub-md5-crypt
   Password: ****
   Retype password: ****
   Encrypted: $1$lS2dv/$JOYcdxIn7CJk9xShzzJVw/
   ```

2. Paste the encrypted string into the global section of the file `menu.lst`:

   ```
   gfxmenu (hd0,4)/message
   color white/blue black/light-gray
   default 0
   timeout 8
   password --md5 $1$lS2dv/$JOYcdxIn7CJk9xShzzJVw/
   ```

   Now GRUB commands can only be executed at the boot prompt after pressing P and entering the password. However, users can still boot all operating systems from the boot menu.

3. To prevent one or several operating systems from being booted from the boot menu, add the entry `lock` to every section in `menu.lst` that should not be bootable without entering a password. For example:

   ```
   title linux
      kernel (hd0,4)/vmlinuz root=/dev/sda7 vga=791
      initrd (hd0,4)/initrd
      lock
   ```

   After rebooting the system and selecting the Linux entry from the boot menu, the following error message is displayed:

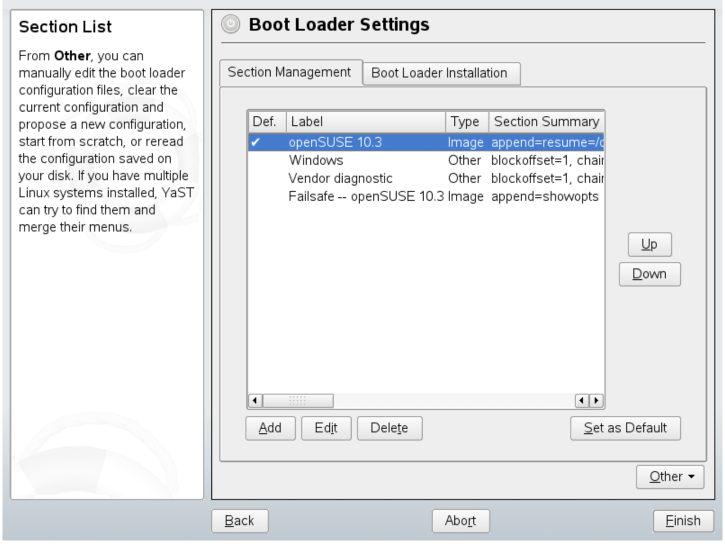
   ```
   Error 32: Must be authenticated
   ```

Press Enter to enter the menu. Then press P to get a password prompt. After entering the password and pressing Enter, the selected operating system (Linux in this case) should boot.

13.3 Configuring the Boot Loader with YaST

The easiest way to configure the boot loader in your openSUSE system is to use the YaST module. In the YaST Control Center, select *System* > *Boot Loader*. As in Figure 13.1, "Boot Loader Settings" (page 205), this shows the current boot loader configuration of your system and allows you to make changes.

Figure 13.1 *Boot Loader Settings*



Section List

From **Other**, you can manually edit the boot loader configuration files, clear the current configuration and propose a new configuration, start from scratch, or reread the configuration saved on your disk. If you have multiple Linux systems installed, YaST can try to find them and merge their menus.

Def.	Label	Type	Section Summary
✓	openSUSE 10.3	Image	append=resume=/c
	Windows	Other	blockoffset=1, chair
	Vendor diagnostic	Other	blockoffset=1, chair
	Failsafe -- openSUSE 10.3	Image	append=showopts

Use the *Section Management* tab to edit, change, and delete boot loader sections for the individual operating systems. To add an option, click *Add*. To change the value of an existing option, select it with the mouse and click *Edit*. To remove an existing entry, select it and click *Delete*. If you are not familiar with boot loader options, read Section 13.2, "Booting with GRUB" (page 196) first.

Use the *Boot Loader Installation* tab to view and change settings related to type, location, and advanced loader settings.

Access advanced configuration options from the drop-down menu that opens after you click on *Other*. The build-in editor lets you change the GRUB configuration files (see Section 13.2, "Booting with GRUB" (page 196) for details). You can also delete the existing configuration and *Start from Scratch* or let YaST *Propose a New Configuration*. It is also possible to write the configuration to disk or reread the configuration from the disk. To restore the original Master Boot Record that was saved during the installation, choose *Restore MBR of Hard Disk*.

13.3.1 Boot Loader Type

Set the boot loader type in *Boot Loader Installation*. The default boot loader in openSUSE is GRUB. To use LILO, proceed as follows:

Procedure 13.1 *Changing the Boot Loader Type*

1. Select the *Boot Loader Installation* tab.

2. For *Boot Loader*, select *LILO*.

3. In the dialog box that opens, select one of the following actions:

   Propose New Configuration
   Have YaST propose a new configuration.

   Convert Current Configuration
   Have YaST convert the current configuration. When converting the configuration, some settings may be lost.

   Start New Configuration from Scratch
   Write a custom configuration. This action is not available during the installation of openSUSE.

   Read Configuration Saved on Disk
   Load your own `/etc/lilo.conf`. This action is not available during the installation of openSUSE.

4. Click *OK* to save the changes

5. Click *Finish* in the main dialog to apply the changes.

During the conversion, the old GRUB configuration is saved to disk. To use it, simply change the boot loader type back to GRUB and choose *Restore Configuration Saved before Conversion*. This action is available only on an installed system.

NOTE: Custom Boot Loader

To use a boot loader other than GRUB or LILO, select *Do Not Install Any Boot Loader*. Read the documentation of your boot loader carefully before choosing this option.

13.3.2 Boot Loader Location

To change the location of the boot loader, follow these steps:

Procedure 13.2 *Changing the Boot Loader Location*

1. Select the *Boot Loader Installation* tab then select one of the following options for *Boot Loader Location*:

   Boot from Boot Partition
      The boot sector of the `/boot` partition.

   Boot from Extended Partition
      This installs the boot loader in the extended partition container.

   Boot from Master Boot Record
      This installs the boot loader in the MBR of the first disk (according to the boot sequence preset in the BIOS).

   Boot from Root Partition
      This installs the boot loader in the boot sector of the `/` partition.

   Custom Boot Partition
      Use this option to specify the location of the boot loader manually.

2. Click *Finish* to apply your changes.

13.3.3 Default System

To change the system that is booted by default, proceed as follows:

Procedure 13.3 *Setting the Default System*

1. Open the *Section Management* tab.

2. Select the desired entry from the list.

3. Click *Set as Default*.

4. Click *Finish* to activate these changes.

13.3.4 Boot Loader Time-Out

The boot loader does not boot the default system immediately. During the time-out, you can select the system to boot or write some kernel parameters. To set the boot loader time-out, proceed as follows:

Procedure 13.4 *Changing the Boot Loader Time-Out*

1. Open the *Boot Loader Installation* tab.

2. Click *Boot Loader Options*.

3. Change the value of *Timeout in Seconds* by typing in a new value, clicking the appropriate arrow key with your mouse, or by using the arrow keys on the keyboard.

4. Click *OK*.

5. Click *Finish* to save the changes.

13.3.5 Security Settings

Using this YaST module, you can also set a password to protect booting. This gives you an additional level of security.

Procedure 13.5 *Setting a Boot Loader Password*

1. Open the *Boot Loader Installation* tab.

2. Click *Boot Loader Options*.

3. Set your password in *Password for the Menu Interface*.

4. Click *OK*.

5. Click *Finish* to save the changes.

13.4 Uninstalling the Linux Boot Loader

YaST can be used to uninstall the Linux boot loader and restore the MBR to the state it had prior to the installation of Linux. During the installation, YaST automatically creates a backup copy of the original MBR and restores it on request.

To uninstall GRUB, start the YaST boot loader module (*System* > *Boot Loader*). Select *Other* > *Restore MBR of Hard Disk* and confirm with *Yes, Rewrite*.

13.5 Creating Boot CDs

If problems occur booting your system using a boot manager or if the boot manager cannot be installed on the MBR of your hard disk or a floppy disk, it is also possible to create a bootable CD with all the necessary start-up files for Linux. This requires a CD writer installed in your system.

Creating a bootable CD-ROM with GRUB merely requires a special form of *stage2* called `stage2_eltorito` and, optionally, a customized `menu.lst`. The classic files `stage1` and `stage2` are not required.

Procedure 13.6 *Creating Boot CDs*

1. Change into a directory in which to create the ISO image, for example: `cd /tmp`

2 Create a subdirectory for GRUB:

```
mkdir -p iso/boot/grub
```

3 Copy the kernel, the files `stage2_eltorito`, `initrd`, `menu.lst`, and `message` to `iso/boot/`:

```
cp /boot/vmlinuz iso/boot/
cp /boot/initrd iso/boot/
cp /boot/message iso/boot/
cp /usr/lib/grub/stage2_eltorito iso/boot/grub
cp /boot/grub/menu.lst iso/boot/grub
```

4 Adjust the path entries in `iso/boot/grub/menu.lst` to make them point to a CD-ROM device. Do this by replacing the device name of the hard disks, listed in the format `(sd*)`, in the pathnames with the device name of the CD-ROM drive, which is `(cd)`:

```
timeout 8
default 0
gfxmenu (cd)/boot/message

title Linux
   root (cd)
   kernel /boot/vmlinuz root=/dev/sda5 vga=794 resume=/dev/sda1 \
   splash=verbose showopts
   initrd /boot/initrd
```

Use `splash=silent` instead of `splash=verbose` to prevent the boot messages from appearing during the boot procedure.

5 Create the ISO image with the following command:

```
mkisofs -R -b boot/grub/stage2_eltorito -no-emul-boot \
-boot-load-size 4 -boot-info-table -o grub.iso /tmp/iso
```

6 Write the resulting file `grub.iso` to a CD using your preferred utility. Do not burn the ISO image as data file, but use the option for burning a CD image in your burning utility.

13.6 The Graphical SUSE Screen

Since SUSE Linux 7.2, the graphical SUSE screen is displayed on the first console if the option `vga=`*`value`* is used as a kernel parameter. If you install using YaST, this option is automatically activated in accordance with the selected resolution and the graphics card. There are three ways to disable the SUSE screen, if desired:

Disabling the SUSE Screen When Necessary
: Enter the command `echo 0 >/proc/splash` on the command line to disable the graphical screen. To activate it again, enter `echo 1 >/proc/splash`.

Disabling the SUSE screen by default.
: Add the kernel parameter `splash=0` to your boot loader configuration. Chapter 13, *The Boot Loader* (page 195) provides more information about this. However, if you prefer the text mode, which was the default in earlier versions, set `vga=normal`.

Completely Disabling the SUSE Screen
: Compile a new kernel and disable the option *Use splash screen instead of boot logo* in *framebuffer support*.

> **TIP**
>
> Disabling framebuffer support in the kernel automatically disables the splash screen as well. SUSE cannot provide any support for your system if you run it with a custom kernel.

13.7 Troubleshooting

This section lists some of the problems frequently encountered when booting with GRUB and a short description of possible solutions. Some of the problems are covered in articles in the Support Database at `http://en.opensuse.org/SDB:SDB`. Use the search dialog to search for keywords like *GRUB*, *boot*, and *boot loader*.

GRUB and XFS
: XFS leaves no room for `stage1` in the partition boot block. Therefore, do not specify an XFS partition as the location of the boot loader. This problem can be solved by creating a separate boot partition that is not formatted with XFS.

GRUB Reports GRUB Geom Error
: GRUB checks the geometry of connected hard disks when the system is booted. Sometimes, the BIOS returns inconsistent information and GRUB reports a GRUB Geom Error. If this is the case, use LILO or update the BIOS. Detailed information

about the installation, configuration, and maintenance of LILO is available in the Support Database under the keyword LILO.

GRUB also returns this error message if Linux was installed on an additional hard disk that is not registered in the BIOS. *stage1* of the boot loader is found and loaded correctly, but *stage2* is not found. This problem can be remedied by registering the new hard disk in the BIOS.

System Containing IDE and SCSI Hard Disks Does Not Boot

During the installation, YaST may have incorrectly determined the boot sequence of the hard disks. For example, GRUB may regard the IDE disk as `hd0` and the SCSI disk as `hd1`, although the boot sequence in the BIOS is reversed (SCSI *before* IDE).

In this case, correct the hard disks during the boot process with the help of the GRUB command line. After the system has booted, edit `device.map` to apply the new mapping permanently. Then check the GRUB device names in the files `/boot/grub/menu.lst` and `/boot/grub/device.map` and reinstall the boot loader with the following command:

```
grub --batch < /etc/grub.conf
```

Booting Windows from the Second Hard Disk

Some operating systems, such as Windows, can only boot from the first hard disk. If such an operating system is installed on a hard disk other than the first hard disk, you can effect a logical change for the respective menu entry.

```
...
title windows
   map (hd0) (hd1)
   map (hd1) (hd0)
   chainloader(hd1,0)+1
...
```

In this example, Windows is started from the second hard disk. For this purpose, the logical order of the hard disks is changed with `map`. This change does not affect the logic within the GRUB menu file. Therefore, the second hard disk must be specified for `chainloader`.

13.8 For More Information

Extensive information about GRUB is available at http://www.gnu.org/software/grub/. Also refer to the grub info page. You can also search for the keyword “SDB:GRUB” in the Support Database at http://www.opensuse.org/ to get information about special issues.

Special System Features 14

This chapter starts with information about various software packages, the virtual consoles, and the keyboard layout. We talk about software components like `bash`, `cron`, and `logrotate`, because they were changed or enhanced during the last release cycles. Even if they are small or considered of minor importance, users may want to change their default behavior, because these components are often closely coupled with the system. The chapter is finished by a section about language and country-specific settings (I18N and L10N).

14.1 Information about Special Software Packages

The programs `bash`, `cron`, `logrotate`, `locate`, `ulimit`, and `free`, and the file `resolv.conf` are very important for system administrators and many users. Man pages and info pages are two useful sources of information about commands, but both are not always available. GNU Emacs is a popular and very configurable text editor.

14.1.1 The bash Package and /etc/profile

Bash is the default system shell. When used as a login shell, it reads several initialization files. Bash processes them in the order they appear in this list:

1. /etc/profile
2. ~/.profile
3. /etc/bash.bashrc
4. ~/.bashrc

Make custom settings in ~/.profile or ~/.bashrc. To ensure the correct processing of these files, it is necessary to copy the basic settings from /etc/skel/.profile or /etc/skel/.bashrc into the home directory of the user. It is recommended to copy the settings from /etc/skel after an update. Execute the following shell commands to prevent the loss of personal adjustments:

```
mv ~/.bashrc ~/.bashrc.old
cp /etc/skel/.bashrc ~/.bashrc
mv ~/.profile ~/.profile.old
cp /etc/skel/.profile ~/.profile
```

Copy personal adjustments back from the *.old files.

14.1.2 The cron Package

If you want to run commands regularly and automatically in the background at predefined times, cron is the tool to use. cron is driven by specially formatted time tables. Some of of them come with the system and users can write their own tables if needed.

The cron tables are located in /var/spool/cron/tabs. /etc/crontab serves as a systemwide cron table. Enter the username to run the command directly after the time table and before the command. In Example 14.1, “Entry in /etc/crontab” (page 216), root is entered. Package-specific tables, located in /etc/cron.d, have the same format. See the cron man page (man cron).

Example 14.1 *Entry in /etc/crontab*

```
1-59/5 * * * *   root   test -x /usr/sbin/atrun && /usr/sbin/atrun
```

You cannot edit /etc/crontab by calling the command crontab -e. This file must be loaded directly into an editor, modified, then saved.

A number of packages install shell scripts to the directories /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly, and /etc/cron.monthly, whose execution is controlled by /usr/lib/cron/run-crons. /usr/lib/cron/run-crons is run every 15 minutes from the main table (/etc/crontab). This guarantees that processes that may have been neglected can be run at the proper time.

To run the hourly, daily, or other periodic maintenance scripts at custom times, remove the time stamp files regularly using /etc/crontab entries (see Example 14.2, "/etc/crontab: Remove Time Stamp Files" (page 217), which removes the hourly one before every full hour, the daily one once a day at 2:14 a.m., etc.).

Example 14.2 */etc/crontab: Remove Time Stamp Files*

```
59 *  * * *     root  rm -f /var/spool/cron/lastrun/cron.hourly
14 2  * * *     root  rm -f /var/spool/cron/lastrun/cron.daily
29 2  * * 6     root  rm -f /var/spool/cron/lastrun/cron.weekly
44 2  1 * *     root  rm -f /var/spool/cron/lastrun/cron.monthly
```

Alternatively, set DAILY_TIME in /etc/sysconfig/cron to the time at which cron.daily should start. The setting of MAX_NOT_RUN ensures that the daily jobs get triggered to run, even if the user did not turn on the computer at the specified DAILY_TIME for a longer period of time. The maximum value of MAX_NOT_RUN is 14 days.

The daily system maintenance jobs are distributed to various scripts for reasons of clarity. They are contained in the package aaa_base. /etc/cron.daily contains, for example, the components suse.de-backup-rpmdb, suse.de-clean-tmp, or suse.de-cron-local.

14.1.3 Log Files: Package logrotate

There are a number of system services (*daemons*) that, along with the kernel itself, regularly record the system status and specific events to log files. This way, the administrator can regularly check the status of the system at a certain point in time, recognize errors or faulty functions, and troubleshoot them with pinpoint precision. These log files are normally stored in /var/log as specified by FHS and grow on a daily basis. The logrotate package helps control the growth of these files.

Configure logrotate with the file /etc/logrotate.conf. In particular, the include specification primarily configures the additional files to read. Programs that produce log files install individual configuration files in /etc/logrotate.d. For example, such files ship with the packages, e.g. apache2 (/etc/logrotate.d/apache2) and syslogd (/etc/logrotate.d/syslog).

Example 14.3 *Example for /etc/logrotate.conf*

```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own lastlog or wtmp - we'll rotate them here
#/var/log/wtmp {
#    monthly
#    create 0664 root utmp
#    rotate 1
#}

# system-specific logs may be also be configured here.
```

logrotate is controlled through cron and is called daily by /etc/cron.daily/logrotate.

IMPORTANT

The create option reads all settings made by the administrator in /etc/permissions*. Ensure that no conflicts arise from any personal modifications.

14.1.4 The locate Command

locate, a command for quickly finding files, is not included in the standard scope of installed software. If desired, install the package findutils-locate. The updatedb process is started automatically every night or about 15 minutes after booting the system.

14.1.5 The ulimit Command

With the `ulimit` (*user limits*) command, it is possible to set limits for the use of system resources and to have these displayed. `ulimit` is especially useful for limiting the memory available for applications. With this, an application can be prevented from using too much memory on its own, which could bring the system to a standstill.

`ulimit` can be used with various options. To limit memory usage, use the options listed in Table 14.1, "`ulimit`: Setting Resources for the User" (page 219).

Table 14.1 *ulimit: Setting Resources for the User*

`-m`	Maximum size of physical memory
`-v`	Maximum size of virtual memory
`-s`	Maximum size of the stack
`-c`	Maximum size of the core files
`-a`	Display of limits set

Systemwide entries can be made in `/etc/profile`. There, enable creation of core files, needed by programmers for *debugging*. A normal user cannot increase the values specified in `/etc/profile` by the system administrator, but can make special entries in `~/.bashrc`.

Example 14.4 *ulimit: Settings in ~/.bashrc*

```
# Limits of physical memory:
ulimit -m 98304

# Limits of virtual memory:
ulimit -v 98304
```

Memory amounts must be specified in KB. For more detailed information, see `man bash`.

14.1.6 The free Command

The free command is somewhat misleading if your goal is to find out how much RAM is currently being used. That information can be found in /proc/meminfo. These days, users with access to a modern operating system, such as Linux, should not really need to worry much about memory. The concept of *available RAM* dates back to before the days of unified memory management. The slogan *free memory is bad memory* applies well to Linux. As a result, Linux has always made the effort to balance out caches without actually allowing free or unused memory.

Basically, the kernel does not have direct knowledge of any applications or user data. Instead, it manages applications and user data in a *page cache*. If memory runs short, parts of it are written to the swap partition or to files, from which they can initially be read with the help of the mmap command (see man mmap).

The kernel also contains other caches, such as the *slab cache*, where the caches used for network access are stored. This may explain differences between the counters in /proc/meminfo. Most, but not all of them, can be accessed via /proc/slabinfo.

14.1.7 The /etc/resolv.conf File

Domain name resolution is handled through the file /etc/resolv.conf. Refer to Chapter 22, *The Domain Name System* (page 353).

This file is updated by the script /sbin/modify_resolvconf exclusively, with no other program having permission to modify /etc/resolv.conf directly. Enforcing this rule is the only way to guarantee that the system's network configuration and the relevant files are kept in a consistent state.

14.1.8 Man Pages and Info Pages

For some GNU applications (such as tar), the man pages are no longer maintained. For these commands, use the --help option to get a quick overview of the info pages, which provide more in-depth instructions. info is GNU's hypertext system. Read an introduction to this system by entering info info. Info pages can be viewed with Emacs by entering emacs -f info or directly in a console with info. You can also use tkinfo, xinfo, or the help system to view info pages.

14.1.9 Settings for GNU Emacs

GNU Emacs is a complex work environment. The following sections cover the configuration files processed when GNU Emacs is started. More information is available at http://www.gnu.org/software/emacs/.

On start-up, Emacs reads several files containing the settings of the user, system administrator, and distributor for customization or preconfiguration. The initialization file ~/.emacs is installed to the home directories of the individual users from /etc/skel. .emacs, in turn, reads the file /etc/skel/.gnu-emacs. To customize the program, copy .gnu-emacs to the home directory (with cp /etc/skel/.gnu-emacs ~/.gnu-emacs) and make the desired settings there.

.gnu-emacs defines the file ~/.gnu-emacs-custom as custom-file. If users make settings with the customize options in Emacs, the settings are saved to ~/.gnu-emacs-custom.

With openSUSE, the emacs package installs the file site-start.el in the directory /usr/share/emacs/site-lisp. The file site-start.el is loaded before the initialization file ~/.emacs. Among other things, site-start.el ensures that special configuration files distributed with Emacs add-on packages, such as psgml, are loaded automatically. Configuration files of this type are located in /usr/share/emacs/site-lisp, too, and always begin with suse-start-. The local system administrator can specify systemwide settings in default.el.

More information about these files is available in the Emacs info file under *Init File*: info:/emacs/InitFile. Information about how to disable loading these files (if necessary) is also provided at this location.

The components of Emacs are divided into several packages:

- The base package emacs.
- emacs-x11 (usually installed): the program *with* X11 support.
- emacs-nox: the program *without* X11 support.
- emacs-info: online documentation in info format.

- emacs-el: the uncompiled library files in Emacs Lisp. These are not required at runtime.

- Numerous add-on packages can be installed if needed: emacs-auctex (for LaTeX), psgml (for SGML and XML), gnuserv (for client and server operation), and others.

14.2 Virtual Consoles

Linux is a multiuser and multitasking system. The advantages of these features can be appreciated even on a stand-alone PC system. In text mode, there are six virtual consoles available. Switch between them using Alt + F1 to Alt + F6. The seventh console is reserved for X and the tenth console shows kernel messages. More or fewer consoles can be assigned by modifying the file /etc/inittab.

To switch to a console from X without shutting it down, use Ctrl + Alt + F1 to Ctrl + Alt + F6. To return to X, press Alt + F7.

14.3 Keyboard Mapping

To standardize the keyboard mapping of programs, changes were made to the following files:

```
/etc/inputrc
/etc/X11/Xmodmap
/etc/skel/.Xmodmap
/etc/skel/.exrc
/etc/skel/.less
/etc/skel/.lesskey
/etc/csh.cshrc
/etc/termcap
/usr/lib/terminfo/x/xterm
/usr/share/X11/app-defaults/XTerm
/usr/share/emacs/VERSION/site-lisp/term/*.el
```

These changes only affect applications that use terminfo entries or whose configuration files are changed directly (vi, less, etc.). Applications not shipped with the system should be adapted to these defaults.

Under X, the compose key (multikey) can be accessed using Ctrl + Shift (right). Also see the corresponding entry in `/etc/X11/Xmodmap`.

Further settings are possible using the X Keyboard Extension (XKB). This extension is also used by the desktop environments GNOME (gswitchit) and KDE (kxkb).

TIP: For More Information

Information about XKB is available in `/etc/X11/xkb/README` and the documents listed there.

Detailed information about the input of Chinese, Japanese, and Korean (CJK) is available at Mike Fabian's page: `http://www.suse.de/~mfabian/suse-cjk/input.html`.

14.4 Language and Country-Specific Settings

The system is, to a very large extent, internationalized and can be modified for local needs in a flexible manner. In other words, internationalization (*I18N*) allows specific localizations (*L10N*). The abbreviations I18N and L10N are derived from the first and last letters of the words and, in between, the number of letters omitted.

Settings are made with `LC_` variables defined in the file `/etc/sysconfig/language`. This refers not only to *native language support*, but also to the categories *Messages* (Language), *Character Set*, *Sort Order*, *Time and Date*, *Numbers*, and *Money*. Each of these categories can be defined directly with its own variable or indirectly with a master variable in the file `language` (see the `locale` man page).

`RC_LC_MESSAGES`, `RC_LC_CTYPE`, `RC_LC_COLLATE`, `RC_LC_TIME`, `RC_LC_NUMERIC`, `RC_LC_MONETARY`

These variables are passed to the shell without the `RC_` prefix and represent the listed categories. The shell profiles concerned are listed below. The current setting can be shown with the command `locale`.

`RC_LC_ALL`

This variable, if set, overwrites the values of the variables already mentioned.

`RC_LANG`
: If none of the previous variables are set, this is the fallback. By default, only `RC_LANG` is set. This makes it easier for users to enter their own values.

`ROOT_USES_LANG`
: A `yes` or `no` variable. If it is set to `no`, `root` always works in the POSIX environment.

The variables can be set with the YaST sysconfig editor (see Section 12.3.1, "Changing the System Configuration Using the YaST sysconfig Editor" (page 192)). The value of such a variable contains the language code, country code, encoding, and modifier. The individual components are connected by special characters:

```
  LANG=<language>[[_<COUNTRY>].<Encoding>[@<Modifier>]]
```

14.4.1 Some Examples

You should always set the language and country codes together. Language settings follow the standard ISO 639 available at `http://www.evertype.com/standards/iso639/iso639-en.html` and `http://www.loc.gov/standards/iso639-2/`. Country codes are listed in ISO 3166 available at `http://www.din.de/gremien/nas/nabd/iso3166ma/codlstp1/en_listp1.html`.

It only makes sense to set values for which usable description files can be found in `/usr/lib/locale`. Additional description files can be created from the files in `/usr/share/i18n` using the command `localedef`. The description files are part of the `glibc-i18ndata` package. A description file for `en_US.UTF-8` (for English and United States) can be created with:

```
localedef -i en_US -f UTF-8 en_US.UTF-8
```

`LANG=en_US.UTF-8`
: This is the default setting if American English is selected during installation. If you selected another language, that language is enabled but still with UTF-8 as the character encoding.

`LANG=en_US.ISO-8859-1`
: This sets the language to English, country to United States, and the character set to `ISO-8859-1`. This character set does not support the Euro sign, but it can be useful sometimes for programs that have not been updated to support `UTF-8`. The string defining the charset (`ISO-8859-1` in this case) is then evaluated by programs like Emacs.

`LANG=en_IE@euro`
: The above example explicitly includes the Euro sign in a language setting. Strictly speaking, this setting is obsolete now, because UTF-8 also covers the Euro symbol. It is only useful if an application does not support UTF-8, but ISO-8859-15.

SuSEconfig reads the variables in `/etc/sysconfig/language` and writes the necessary changes to `/etc/SuSEconfig/profile` and `/etc/SuSEconfig/csh.cshrc`. `/etc/SuSEconfig/profile` is read or *sourced* by `/etc/profile`. `/etc/SuSEconfig/csh.cshrc` is sourced by `/etc/csh.cshrc`. This makes the settings available systemwide.

Users can override the system defaults by editing their `~/.bashrc` accordingly. For instance, if you do not want to use the systemwide `en_US` for program messages, include `LC_MESSAGES=es_ES` so messages are displayed in Spanish instead.

14.4.2 Locale Settings in ~/.i18n

If you are not satisfied with locale system defaults, change the settings in `~/.i18n` according to the Bash scripting syntax. Entries in `~/.i18n` override system defaults from `/etc/sysconfig/language`. Use the same variable names but without the `RC_` namespace prefixes, for example, use `LANG` instead of `RC_LANG`:

```
LANG=cs_CZ.UTF-8
LC_COLLATE=C
```

14.4.3 Settings for Language Support

Files in the category *Messages* are, as a rule, only stored in the corresponding language directory (like `en`) to have a fallback. If you set `LANG` to `en_US` and the message file in `/usr/share/locale/en_US/LC_MESSAGES` does not exist, it falls back to `/usr/share/locale/en/LC_MESSAGES`.

A fallback chain can also be defined, for example, for Breton to French or for Galician to Spanish to Portuguese:

```
LANGUAGE="br_FR:fr_FR"
```

```
LANGUAGE="gl_ES:es_ES:pt_PT"
```

If desired, use the Norwegian variants Nynorsk and Bokmål instead (with additional fallback to `no`):

```
LANG="nn_NO"
```

```
LANGUAGE="nn_NO:nb_NO:no"
```

or

```
LANG="nb_NO"
```

```
LANGUAGE="nb_NO:nn_NO:no"
```

Note that in Norwegian, `LC_TIME` is also treated differently.

One problem that can arise is a separator used to delimit groups of digits not being recognized properly. This occurs if `LANG` is set to only a two-letter language code like `de`, but the definition file glibc uses is located in `/usr/share/lib/de_DE/LC_NUMERIC`. Thus `LC_NUMERIC` must be set to `de_DE` to make the separator definition visible to the system.

14.4.4 For More Information

- *The GNU C Library Reference Manual*, Chapter “Locales and Internationalization”. It is included in `glibc-info`.
- Markus Kuhn, *UTF-8 and Unicode FAQ for Unix/Linux*, currently at `http://www.cl.cam.ac.uk/~mgk25/unicode.html`.
- *Unicode-Howto*, by Bruno Haible: `/usr/share/doc/howto/en/html/Unicode-HOWTO.html`.

Dynamic Kernel Device Management with udev

15

The kernel can add or remove almost any device in the running system. Changes in device state (whether a device is plugged in or removed) need to be propagated to userspace. Devices need to be configured as soon as they are plugged in and discovered. Users of a certain device need to be informed about any state changes of this device. udev provides the needed infrastructure to dynamically maintain the device node files and symbolic links in the `/dev` directory. udev rules provide a way to plug external tools into the kernel device event processing. This enables you to customize udev device handling, for example, by adding certain scripts to execute as part of kernel device handling, or request and import additional data to evaluate during device handling.

15.1 The `/dev` Directory

The device nodes in the `/dev` directory provide access to the corresponding kernel devices. With udev, the `/dev` directory reflects the current state of the kernel. Every kernel device has one corresponding device file. If a device is disconnected from the system, the device node is removed.

The content of the `/dev` directory is kept on a temporary file system and all files are created from scratch at every system start-up. Manually created or changed files intentionally do not survive a reboot. Static files and directories that should always be present in the `/dev` directory regardless of the state of the corresponding kernel device can be placed in the `/lib/udev/devices` directory. At system start-up, the contents of that directory is copied to the `/dev` directory with the same ownership and permissions as the files in `/lib/udev/devices`.

15.2 Kernel uevents and udev

The required device information is exported by the sysfs file system. For every device the kernel has detected and initialized, a directory with the device name is created. It contains attribute files with device-specific properties.

Every time a device is added or removed, the kernel sends a uevent to notify udev of the change. The udev daemon reads and parses all provided rules from the `/etc/udev/rules.d/*.rules` files once at start-up and keeps them in memory. If rules files are changed, added, or removed, the daemon receives an event and updates the in-memory representation of the rules. For more details on udev rules and their syntax, refer to Section 15.6, “Influencing Kernel Device Event Handling with udev Rules” (page 231).

Every received event is matched against the set of provides rules. The rules can add or change event environment keys, request a specific name for the device node to create, add symlinks pointing to the node, or add programs to run after the device node is created. The driver core uevents are received from a kernel netlink socket.

15.3 Drivers, Kernel Modules, and Devices

The kernel bus drivers probe for devices. For every detected device, the kernel creates an internal device structure and the driver core sends a uevent to the udev daemon. Bus devices identify themselves by a specially-formatted ID, which tells what kind of device it is. Usually these IDs consist of vendor and product ID and other subsystem-specific values. Every bus has its own scheme for these IDs, called `MODALIAS`. The kernel takes the device information, composes a `MODALIAS` ID string from it, and sends that string along with the event. For a USB mouse, it looks like this:

```
MODALIAS=usb:v046DpC03Ed2000dc00dsc00dp00ic03isc01ip02
```

Every device driver carries a list of known aliases for devices it can handle. The list is contained in the kernel module file itself. The program depmod reads the ID lists and creates the file `modules.alias` in the kernel's `/lib/modules` directory for all currently available modules. With this infrastructure, module loading is as easy as calling `modprobe` for every event that carries a `MODALIAS` key. If `modprobe`

$MODALIAS is called, it matches the device alias composed for the device with the aliases provided by the modules. If a matching entry is found, that module is loaded. All this is triggered by udev and happens automatically.

15.4 Booting and Initial Device Setup

All device events happening during the boot process before the udev daemon is running are lost, because the infrastructure to handle these events lives on the root file system and is not available at that time. To cover that loss, the kernel provides a `uevent` file located in the device directory of every device in the sysfs file system. By writing `add` to that file, the kernel resends the same event as the one lost during boot. A simple loop over all `uevent` files in `/sys` triggers all events again to create the device nodes and perform device setup.

As an example, a USB mouse present during boot may not be initialized by the early boot logic, because the driver is not available at that time. The event for the device discovery was lost and failed to find a kernel module for the device. Instead of manually searching for possibly connected devices, udev just requests all device events from the kernel after the root file system is available, so the event for the USB mouse device just runs again. Now it finds the kernel module on the mounted root file system and the USB mouse can be initialized.

From userspace, there is no visible difference between a device coldplug sequence and a device discovery during runtime. In both cases, the same rules are used to match and the same configured programs are run.

15.5 Monitoring the Running udev Daemon

The program `udevmonitor` can be used to visualize the driver core events and the timing of the udev event processes.

```
UEVENT[1185238505.276660] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1 (usb)
UDEV  [1185238505.279198] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1 (usb)
UEVENT[1185238505.279527] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0 (usb)
```

```
UDEV  [1185238505.285573] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0 (usb)
UEVENT[1185238505.298878] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10 (input)
UDEV  [1185238505.305026] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10 (input)
UEVENT[1185238505.305442] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/mouse2 (input)
UEVENT[1185238505.306440] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/event4 (input)
UDEV  [1185238505.325384] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/event4 (input)
UDEV  [1185238505.342257] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/mouse2 (input)
```

The `UEVENT` lines show the events the kernel has sent over netlink. The `UDEV` lines show the finished udev event handlers. The timing is printed in microseconds. The time between `UEVENT` and `UDEV` is the time udev took to process this event or the udev daemon has delayed its execution to synchronize this event with related and already running events. For example, events for hard disk partitions always wait for the main disk device event to finish, because the partition events may rely on the data the main disk event has queried from the hardware.

`udevmonitor --env` shows the complete event environment:

```
ACTION=add
DEVPATH=/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10
SUBSYSTEM=input
SEQNUM=1181
NAME="Logitech USB-PS/2 Optical Mouse"
PHYS="usb-0000:00:1d.2-1/input0"
UNIQ=""
EV=7
KEY=70000 0 0 0 0
REL=103
MODALIAS=input:b0003v046DpC03Ee0110-e0,1,2,k110,111,112,r0,1,8,amlsfw
```

udev also sends messages to syslog. The default syslog priority that controls which messages are sent to syslog is specified in the udev configuration file `/etc/udev/udev.conf`. The log priority of the running daemon can be changed with `udevcontrol log_priority=`*`level/number`*.

15.6 Influencing Kernel Device Event Handling with udev Rules

A udev rule can match any property the kernel adds to the event itself or any information that the kernel exports to `sysfs`. The rule can also request additional information from external programs. Every event is matched against all provided rules. All rules are located in the `/etc/udev/rules.d` directory.

Every line in the rules file contains at least one key value pair. There are two kinds of keys, match and assignment keys. If all match keys match their values, the rule is applied and the assignment keys are assigned the specified value. A matching rule may specify the name of the device node, add symlinks pointing to the node, or run a specified program as part of the event handling. If no matching rule is found, the default device node name is used to create the device node. Detailed information about the rule syntax and the provided keys to match or import data are described in the udev man page. The following example rules provide a basic introduction to udev rule syntax. The example rules are all taken from the udev default rule set that is located under `/etc/udev/rules.d/50-udev-default.rules`.

Example 15.1 *Example udev Rules*

```
# console
KERNEL=="console", MODE="0600", OPTIONS="last_rule"

# serial devices
KERNEL=="ttyUSB*", ATTRS{product}=="[Pp]alm*Handheld*", SYMLINK+="pilot"

# printer
SUBSYSTEM=="usb", KERNEL=="lp*", NAME="usb/%k", SYMLINK+="usb%k", GROUP="lp"

# kernel firmware loader
SUBSYSTEM=="firmware", ACTION=="add", RUN+="firmware.sh"
```

The `console` rule consist of three keys. One match key (`KERNEL`), and two assign keys (`MODE`, `OPTIONS`). The `KERNEL` match rule searches the device list for any items of the type `console`. Only exact matches are valid and trigger this rule to be executed. The `MODE` key assigns special permissions to the device node, in this case, read and write permissions to the owner of this device and none else. The `OPTIONS` key makes this rule the last rule to be applied to any device of this type. Any later rule matching this particular device type does not have any effect.

The serial devices rule consists of two match keys (KERNEL and ATTRS) and one assign key (SYMLINK). The KERNEL key searches for all devices of the ttyUSB type. Using the * wild card, this key matches several of these devices. The second match key, ATTRS, checks whether the product attribute file in sysfs for any ttyUSB device contains a certain string. The assign key (SYMLINK) triggers the addition of a symbolic link to this device under /dev/pilot. The operator used in this key (+=) tells udev to additionally perform this action, even if previous or later rules add other symbolic links. As this rule contains two match keys, it is only applied if both conditions are met.

The printer rule deals with USB printers and contains two match keys which must both apply to get the entire rule applied (SUBSYSTEM and KERNEL). Three assign keys deal with the naming for this device type (NAME), the creation of symbolic device links (SYMLINK), and the group membership for this device type (GROUP). Using the * wild card in the KERNEL key makes it match several lp printer devices. Substitutions are used in both the NAME and the SYMLINK keys to extend these strings by the internal device name. For example, the symlink to the first lp USB printer would read /dev/usblp0.

The kernel firmware loader rule makes udev load additional firmware by an external helper script during runtime. The SUBSYSTEM match key searches for the firmware subsystem. The ACTION key checks whether any device belonging to the firmware subsystem has been added. The RUN+= key triggers the execution of the firmware.sh script to locate the firmware that is to be loaded.

Some general characteristics are common to all rules:

- Each rule consists of one or more key value pairs separated by a comma.
- A key's operation is determined by the operator. udev rules support several different operators.
- Each given value must be enclosed by quotation marks.
- Each line of the rules file represents one rule. If a rule is longer than just one line, use \ to join the different lines just as you would do in shell syntax.
- udev rules support a shell-style pattern matching for the *, ?, and [] patterns.
- udev rules support substitutions.

15.6.1 Using Operators in udev Rules

Creating keys you can choose from several different operators, depending on the type of key you want to create. Match keys will normally just be used to find a value that either matches or explicitly mismatches the search value. Match keys contain either of the following operators:

`==`
: Compare for equality. If the key contains a search pattern, then all results matching this pattern are valid.

`!=`
: Compare for non-equality. If the key contains a search pattern, then all results matching this pattern are valid.

Any of the following operators can be used with assign keys:

`=`
: Assign a value to a key. If the key previously consisted of a list of values, the key resets and only the single value is assigned.

`+=`
: Add a value to a key that contains a list of entries.

`:=`
: Assign a final value. Disallow any later change (by later rules).

15.6.2 Using Substitutions in udev Rules

udev rules support the use of placeholders and substitutions. Use them in a similar fashion as you would do in any other scripts. The following substitutions can be used with udev rules:

`%r`, `$root`
: The device directory, `/dev` by default

`%p`, `$devpath`
: The value of `DEVPATH`

`%k`, `$kernel`
: The value of `KERNEL` or the internal device name

`%n`, `$number`
: The device number

`%N`, `$tempnode`
: The temporary name of the device file

`%M`, `$major`
: The major number of the device

`%m`, `$minor`
: The minor number of the device

`%s{`*`attribute`*`}`, `$attr{`*`attribute`*`}`
: The value of a `sysfs` attribute (specified by *`attribute`*)

`%E{`*`variable`*`}`, `$attr{`*`variable`*`}`
: The value of an environment variable (specified by *`variable`*)

`%c`, `$result`
: The output of `PROGRAM`

`%%`
: The `%` character

`$$`
: The `$` character

15.6.3 Using udev Match Keys

Match keys describe conditions that must be met before a udev rule can be applied. The following match keys are available:

`ACTION`
: The name of the event action, e.g. `add` or `remove` for a device add or remove action.

`DEVPATH`
: The device path of the event device, e.g. `DEVPATH=/bus/pci/drivers/ipw3945` to search for all events related to the ipw3945 driver.

`KERNEL`
: The internal (kernel) name of the event device.

`SUBSYSTEM`
: The subsystem of the event device, e.g. `SUBSYSTEM=usb` for all events related to USB devices.

`ATTR{`*`filename`*`}`
: sysfs attributes of the event device. To match a string contained in the `vendor` attribute file name, you could use `ATTR{vendor}=="On[sS]tream"`, for example.

`KERNELS`
: Let udev search the device path upwards for a matching device name.

`SUBSYSTEMS`
: Let udev search the device path upwards for a matching device subsystem name.

`DRIVERS`
: Let udev search the device path upwards for a matching device driver name.

`ATTRS{`*`filename`*`}`
: Let udev search the device path upwards for a device with matching sysfs attribute values.

`ENV{`*`key`*`}`
: The value of an environment variable, e.g. `ENV{ID_BUS}="ieee1394` to search for all events related to the FireWire bus ID.

`PROGRAM`
: Let udev execute an external program. For this key to be true, the program must return without exit code zero. The program's output is printed to stdout and available to the `RESULT` key.

RESULT
: Match the return value/string of the last PROGRAM call. Either include this key in the same rule as the PROGRAM key or in a later one.

15.6.4 Using udev Assign Keys

In contrast to the match keys described above, assign keys do not describe conditions that must be met, but assign values, names and actions to the device nodes maintained by udev.

NAME
: The name of the device node to be created. Once a rule has set a node name, all other rules with a NAME key for this node are ignored.

SYMLINK
: The name of a symlink related to the node to be created. Multiple matching rules can add symlinks to be created with the device node. You can also specify multiple symlinks for one node in one rule using the space character to separate the symlink names.

OWNER, GROUP, MODE
: The permissions for the new device node. Values specified here overwrite anything that has been compiled in.

ATTR{*key*}
: Specify a value to be written to a sysfs attribute of the event device. If the == operator is used, this key is also used to match against the value of a sysfs attribute.

ENV{*key*}
: Tell udev to export a variable to the environment. If the == operator is used, this key is also used to match against an environment variable.

RUN
: Tell udev to add a program to the list of programs to be executed for this device. Mind to restrict this to very short tasks to avoid blocking further events for this device.

LABEL
: Add a label where a GOTO can jump to.

`GOTO`

Tell udev to skip a number of rules and continue with the one that carries the label referenced by the `GOTO` key.

`IMPORT{`*`type`*`}`

Load variables into the event environment such as the output of an external program. udev imports variables of several different types. If no type is specified, udev tries to determine the type itself based on the executable bit of the file permissions.

- `program` tells udev to execute an external program and import its output.
- `file` tells udev to import a text file.
- `parent` tells udev to import the stored keys from the parent device.

`WAIT_FOR_SYSFS`

Tells udev to wait for the specified sysfs file to be created for a certain device, e.g. `WAIT_FOR_SYSFS="ioerr_cnt"` informs udev to wait until the `ioerr_cnt` file has been created.

`OPTIONS`

There are several possible values to the `OPTION` key:

- `last_rule` tells udev to ignore all later rules.
- `ignore_device` tells udev to ignore this event completely.
- `ignore_remove` tells udev to ignore all later remove event the device.
- `all_partitions` tells udev to create device nodes for all available partitions on a block device.

15.7 Persistent Device Naming

The dynamic device directory and the udev rules infrastructure make it possible to provide stable names for all disk devices—regardless of their order of recognition or the connection used for the device. Every appropriate block device the kernel creates is examined by tools with special knowledge about certain buses, drive types, or file systems. Along with the dynamic kernel-provided device node name, udev maintains classes of persistent symbolic links pointing to the device:

```
/dev/disk
|-- by-id
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B -> ../../sda
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part1 -> ../../sda1
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part6 -> ../../sda6
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part7 -> ../../sda7
|   |-- usb-Generic_STORAGE_DEVICE_02773 -> ../../sdd
|   `-- usb-Generic_STORAGE_DEVICE_02773-part1 -> ../../sdd1
|-- by-label
|   |-- Photos -> ../../sdd1
|   |-- SUSE10 -> ../../sda7
|   `-- devel -> ../../sda6
|-- by-path
|   |-- pci-0000:00:1f.2-scsi-0:0:0:0 -> ../../sda
|   |-- pci-0000:00:1f.2-scsi-0:0:0:0-part1 -> ../../sda1
|   |-- pci-0000:00:1f.2-scsi-0:0:0:0-part6 -> ../../sda6
|   |-- pci-0000:00:1f.2-scsi-0:0:0:0-part7 -> ../../sda7
|   |-- pci-0000:00:1f.2-scsi-1:0:0:0 -> ../../sr0
|   |-- usb-02773:0:0:2 -> ../../sdd
|   |-- usb-02773:0:0:2-part1 -> ../../sdd1
`-- by-uuid
    |-- 159a47a4-e6e6-40be-a757-a629991479ae -> ../../sda7
    |-- 3e999973-00c9-4917-9442-b7633bd95b9e -> ../../sda6
    `-- 4210-8F8C -> ../../sdd1
```

15.8 Files used by udev

`/sys/*`
: Virtual file system provided by the Linux kernel, exporting all currently known devices. This information is used by udev to create device nodes in `/dev`

`/dev/*`
: Dynamically created device nodes and static content copied at bootup from `/lib/udev/devices/*`

The following files and directories contain the crucial elements of the udev infrastructure:

`/etc/udev/udev.conf`
: Main udev configuration file

`/etc/udev/rules.d/*`
: udev event matching rules

`/lib/udev/devices/*`
: Static `/dev` content

`/lib/udev/*`
: Helper programs called from udev rules

15.9 For More Information

For more information about the udev infrastructure, refer to the following man pages:

udev
: General information about udev, keys, rules, and other important configuration issues.

udevinfo
: udevinfo can be used to query device information from the udev database.

udevd
: Information about the udev event managing daemon.

udevmonitor
: udevmonitor prints the kernel and udev event sequence to the console. This tool is mainly used for debugging purposes.

16 File Systems in Linux

openSUSE® ships with a number of different file systems, including ReiserFS, Ext2, Ext3, and XFS, from which to choose at installation time. Each file system has its own advantages and disadvantages that can make it more suited to a scenario. Professional high-performance setups may require a different choice of file system than a home user's setup.

16.1 Terminology

metadata
: A file system–internal data structure that assures all the data on disk is properly organized and accessible. Essentially, it is "data about the data." Almost every file system has its own structure of metadata, which is part of why the file systems show different performance characteristics. It is extremely important to maintain metadata intact, because otherwise all data on the file system could become inaccessible.

inode
: Inodes contain various information about a file, including size, number of links, pointers to the disk blocks where the file contents are actually stored, and date and time of creation, modification, and access.

journal
: In the context of a file system, a journal is an on-disk structure containing a kind of log in which the file system stores what it is about to change in the file system's metadata. Journaling greatly reduces the recovery time of a Linux system because

it obsoletes the lengthy search process that checks the entire file system at system start-up. Instead, only the journal is replayed.

16.2 Major File Systems in Linux

Unlike two or three years ago, choosing a file system for a Linux system is no longer a matter of a few seconds (Ext2 or ReiserFS?). Kernels starting from 2.4 offer a variety of file systems from which to choose. The following is an overview of how these file systems basically work and which advantages they offer.

It is very important to bear in mind that there may be no file system that best suits all kinds of applications. Each file system has its particular strengths and weaknesses, which must be taken into account. Even the most sophisticated file system cannot replace a reasonable backup strategy, however.

The terms *data integrity* and *data consistency*, when used in this chapter, do not refer to the consistency of the user space data (the data your application writes to its files). Whether this data is consistent must be controlled by the application itself.

IMPORTANT: Setting Up File Systems

Unless stated otherwise in this chapter, all the steps required to set up or change partitions and file systems can be performed using YaST.

16.2.1 ReiserFS

Officially one of the key features of the 2.4 kernel release, ReiserFS has been available as a kernel patch for 2.2.x SUSE kernels since version 6.4. ReiserFS was designed by Hans Reiser and the Namesys development team. It has proven itself to be a powerful alternative to Ext2. Its key assets are better disk space utilization, better disk access performance, and faster crash recovery.

ReiserFS's strengths, in more detail, are:

Better Disk Space Utilization

In ReiserFS, all data is organized in a structure called B*-balanced tree. The tree structure contributes to better disk space utilization because small files can be stored directly in the B* tree leaf nodes instead of being stored elsewhere and just maintaining a pointer to the actual disk location. In addition to that, storage is not allocated in chunks of 1 or 4 KB, but in portions of the exact size needed. Another benefit lies in the dynamic allocation of inodes. This keeps the file system more flexible than traditional file systems, like Ext2, where the inode density must be specified at file system creation time.

Better Disk Access Performance

For small files, file data and “stat_data” (inode) information are often stored next to each other. They can be read with a single disk I/O operation, meaning that only one access to disk is required to retrieve all the information needed.

Fast Crash Recovery

Using a journal to keep track of recent metadata changes makes a file system check a matter of seconds, even for huge file systems.

Reliability through Data Journaling

ReiserFS also supports data journaling and ordered data modes similar to the concepts outlined in the Ext3 section, Section 16.2.3, “Ext3” (page 244). The default mode is `data=ordered`, which ensures both data and metadata integrity, but uses journaling only for metadata.

16.2.2 Ext2

The origins of Ext2 go back to the early days of Linux history. Its predecessor, the Extended File System, was implemented in April 1992 and integrated in Linux 0.96c. The Extended File System underwent a number of modifications and, as Ext2, became the most popular Linux file system for years. With the creation of journaling file systems and their astonishingly short recovery times, Ext2 became less important.

A brief summary of Ext2's strengths might help understand why it was—and in some areas still is—the favorite Linux file system of many Linux users.

Solidity

Being quite an “old-timer,” Ext2 underwent many improvements and was heavily tested. This may be the reason why people often refer to it as rock-solid. After a system outage when the file system could not be cleanly unmounted, e2fsck starts to analyze the file system data. Metadata is brought into a consistent state and pending files or data blocks are written to a designated directory (called `lost`

+found). In contrast to journaling file systems, e2fsck analyzes the entire file system and not just the recently modified bits of metadata. This takes significantly longer than checking the log data of a journaling file system. Depending on file system size, this procedure can take half an hour or more. Therefore, it is not desirable to choose Ext2 for any server that needs high availability. However, because Ext2 does not maintain a journal and uses significantly less memory, it is sometimes faster than other file systems.

Easy Upgradability

The code for Ext2 is the strong foundation on which Ext3 could become a highly-acclaimed next-generation file system. Its reliability and solidity were elegantly combined with the advantages of a journaling file system.

16.2.3 Ext3

Ext3 was designed by Stephen Tweedie. Unlike all other next-generation file systems, Ext3 does not follow a completely new design principle. It is based on Ext2. These two file systems are very closely related to each other. An Ext3 file system can be easily built on top of an Ext2 file system. The most important difference between Ext2 and Ext3 is that Ext3 supports journaling. In summary, Ext3 has three major advantages to offer:

Easy and Highly Reliable Upgrades from Ext2

Because Ext3 is based on the Ext2 code and shares its on-disk format as well as its metadata format, upgrades from Ext2 to Ext3 are incredibly easy. Unlike transitions to other journaling file systems, such as ReiserFS or XFS, which can be quite tedious (making backups of the entire file system and recreating it from scratch), a transition to Ext3 is a matter of minutes. It is also very safe, because recreating an entire file system from scratch might not work flawlessly. Considering the number of existing Ext2 systems that await an upgrade to a journaling file system, you can easily figure out why Ext3 might be of some importance to many system administrators. Downgrading from Ext3 to Ext2 is as easy as the upgrade. Just perform a clean unmount of the Ext3 file system and remount it as an Ext2 file system.

Reliability and Performance

Some other journaling file systems follow the “metadata-only” journaling approach. This means your metadata is always kept in a consistent state, but the same cannot be automatically guaranteed for the file system data itself. Ext3 is designed to take care of both metadata and data. The degree of “care” can be customized. Enabling

Ext3 in the `data=journal` mode offers maximum security (data integrity), but can slow down the system because both metadata and data are journaled. A relatively new approach is to use the `data=ordered` mode, which ensures both data and metadata integrity, but uses journaling only for metadata. The file system driver collects all data blocks that correspond to one metadata update. These data blocks are written to disk before the metadata is updated. As a result, consistency is achieved for metadata and data without sacrificing performance. A third option to use is `data=writeback`, which allows data to be written into the main file system after its metadata has been committed to the journal. This option is often considered the best in performance. It can, however, allow old data to reappear in files after crash and recovery while internal file system integrity is maintained. Unless you specify something else, Ext3 is run with the `data=ordered` default.

16.2.4 Converting an Ext2 File System into Ext3

To convert an Ext2 file system to Ext3, proceed as follows:

1. Create an Ext3 journal by running `tune2fs -j` as `root`. This creates an Ext3 journal with the default parameters.

   To decide yourself how large the journal should be and on which device it should reside, run `tune2fs -J` instead together with the desired journal options `size=` and `device=`. More information about the tune2fs program is available in the tune2fs manual page.

2. To ensure that the Ext3 file system is recognized as such, edit the file `/etc/fstab` as `root`, changing the file system type specified for the corresponding partition from `ext2` to `ext3`. The change takes effect after the next reboot.

3. To boot a root file system set up as an Ext3 partition, include the modules `ext3` and `jbd` in the `initrd`. To do this, edit `/etc/sysconfig/kernel` as `root`, adding `ext3` and `jbd` to the `INITRD_MODULES` variable. After saving the changes, run the `mkinitrd` command. This builds a new initrd and prepares it for use.

16.2.5 XFS

Originally intended as the file system for their IRIX OS, SGI started XFS development in the early 1990s. The idea behind XFS was to create a high-performance 64-bit journaling file system to meet the extreme computing challenges of today. XFS is very good at manipulating large files and performs well on high-end hardware. However, even XFS has a drawback. Like ReiserFS, XFS takes great care of metadata integrity, but less of data integrity.

A quick review of XFS's key features explains why it may prove a strong competitor for other journaling file systems in high-end computing.

High Scalability through the Use of Allocation Groups

> At the creation time of an XFS file system, the block device underlying the file system is divided into eight or more linear regions of equal size. Those are referred to as *allocation groups*. Each allocation group manages its own inodes and free disk space. Practically, allocation groups can be seen as file systems in a file system. Because allocation groups are rather independent of each other, more than one of them can be addressed by the kernel simultaneously. This feature is the key to XFS's great scalability. Naturally, the concept of independent allocation groups suits the needs of multiprocessor systems.

High Performance through Efficient Management of Disk Space

> Free space and inodes are handled by B^+ trees inside the allocation groups. The use of B^+ trees greatly contributes to XFS's performance and scalability. XFS uses *delayed allocation*. It handles allocation by breaking the process into two pieces. A pending transaction is stored in RAM and the appropriate amount of space is reserved. XFS still does not decide where exactly (speaking of file system blocks) the data should be stored. This decision is delayed until the last possible moment. Some short-lived temporary data may never make its way to disk, because it may be obsolete by the time XFS decides where actually to save it. Thus XFS increases write performance and reduces file system fragmentation. Because delayed allocation results in less frequent write events than in other file systems, it is likely that data loss after a crash during a write is more severe.

Preallocation to Avoid File System Fragmentation

> Before writing the data to the file system, XFS *reserves* (preallocates) the free space needed for a file. Thus, file system fragmentation is greatly reduced. Performance is increased because the contents of a file are not distributed all over the file system.

16.3 Some Other Supported File Systems

Table 16.1, “File System Types in Linux” (page 247) summarizes some other file systems supported by Linux. They are supported mainly to ensure compatibility and interchange of data with different kinds of media or foreign operating systems.

***Table 16.1** File System Types in Linux*

`cramfs`	*Compressed ROM file system*: A compressed read-only file system for ROMs.
`hpfs`	*High Performance File System*: The IBM OS/2 standard file system—only supported in read-only mode.
`iso9660`	Standard file system on CD-ROMs.
`minix`	This file system originated from academic projects on operating systems and was the first file system used in Linux. Today, it is used as a file system for floppy disks.
`msdos`	*fat*, the file system originally used by DOS, is today used by various operating systems.
`ncpfs`	File system for mounting Novell volumes over networks.
`nfs`	*Network File System*: Here, data can be stored on any machine in a network and access may be granted via a network.
`smbfs`	*Server Message Block* is used by products such as Windows to enable file access over a network.
`sysv`	Used on SCO UNIX, Xenix, and Coherent (commercial UNIX systems for PCs).
`ufs`	Used by BSD, SunOS, and NeXTSTEP. Only supported in read-only mode.

`umsdos` *UNIX on MSDOS*: Applied on top of a normal `fat` file system, achieves UNIX functionality (permissions, links, long filenames) by creating special files.

`vfat` *Virtual FAT*: Extension of the `fat` file system (supports long filenames).

`ntfs` *Windows NT file system*, read-only.

16.4 Large File Support in Linux

Originally, Linux supported a maximum file size of 2 GB. This was enough before the explosion of multimedia and as long as no one tried to manipulate huge databases on Linux. Becoming more and more important for server computing, the kernel and C library were modified to support file sizes larger than 2 GB when using a new set of interfaces that applications must use. Today, almost all major file systems offer LFS support, allowing you to perform high-end computing. Table 16.2, "Maximum Sizes of File Systems (On-Disk Format)" (page 248) offers an overview of the current limitations of Linux files and file systems.

Table 16.2 *Maximum Sizes of File Systems (On-Disk Format)*

File System	File Size (Bytes)	File System Size (Bytes)
Ext2 or Ext3 (1 KB block size)	2^{34} (16 GB)	2^{41} (2 TB)
Ext2 or Ext3 (2 KB block size)	2^{38} (256 GB)	2^{43} (8 TB)
Ext2 or Ext3 (4 KB block size)	2^{41} (2 TB)	2^{43} – 4096 (16 TB – 4096 Bytes)
Ext2 or Ext3 (8 KB block size) (systems with 8 KB pages, like Alpha)	2^{46} (64 TB)	2^{45} (32 TB)
ReiserFS v3	2^{46} (64 TB)	2^{45} (32 TB)

File System	File Size (Bytes)	File System Size (Bytes)
XFS	2^{63} (8 EB)	2^{63} (8 EB)
NFSv2 (client side)	2^{31} (2 GB)	2^{63} (8 EB)
NFSv3 (client side)	2^{63} (8 EB)	2^{63} (8 EB)

IMPORTANT: Linux Kernel Limits

Table 16.2, "Maximum Sizes of File Systems (On-Disk Format)" (page 248) describes the limitations regarding the on-disk format. The 2.6 kernel imposes its own limits on the size of files and file systems handled by it. These are as follows:

File Size
: On 32-bit systems, files may not exceed the size of 2 TB (2^{41} bytes).

File System Size
: File systems may be up to 2^{73} bytes in size. However, this limit is still out of reach for the currently available hardware.

16.5 For More Information

Each of the file system projects described above maintains its own home page on which to find mailing list information, further documentation, and FAQs.

- http://e2fsprogs.sourceforge.net/
- http://www.zipworld.com.au/~akpm/linux/ext3/
- http://www.namesys.com/
- http://oss.sgi.com/projects/xfs/
- http://oss.oracle.com/projects/ocfs2/

A comprehensive multipart tutorial about Linux file systems can be found at *IBM developerWorks*: `http://www-106.ibm.com/developerworks/library/l-fs.html`. A very in-depth comparison of file systems (not only Linux file systems) is available from the Wikipedia project `http://en.wikipedia.org/wiki/Comparison_of_file_systems#Comparison`.

Access Control Lists in Linux 17

POSIX ACLs (access control lists) can be used as an expansion of the traditional permission concept for file system objects. With ACLs, permissions can be defined more flexibly than the traditional permission concept allows.

The term *POSIX ACL* suggests that this is a true POSIX (*portable operating system interface*) standard. The respective draft standards POSIX 1003.1e and POSIX 1003.2c have been withdrawn for several reasons. Nevertheless, ACLs as found on many systems belonging to the UNIX family are based on these drafts and the implementation of file system ACLs as described in this chapter follows these two standards as well. They can be viewed at `http://wt.xpilot.org/publications/posix.1e/`.

17.1 Traditional File Permissions

The basics of traditional Linux file permissions are explained in Section 19.2, "Users and Access Permissions" (page 277). More advanced features are the setuid, setgid, and sticky bit.

17.1.1 The setuid Bit

In certain situations, the access permissions may be too restrictive. Therefore, Linux has additional settings that enable the temporary change of the current user and group identity for a specific action. For example, the `passwd` program normally requires root permissions to access `/etc/passwd`. This file contains some important information, like the home directories of users and user and group IDs. Thus, a normal user would not be able to change `passwd`, because it would be too dangerous to grant all users direct access to this file. A possible solution to this problem is the *setuid* mechanism. setuid (set user ID) is a special file attribute that instructs the system to execute programs marked accordingly under a specific user ID. Consider the `passwd` command:

```
-rwsr-xr-x  1 root shadow 80036 2004-10-02 11:08 /usr/bin/passwd
```

You can see the `s` that denotes that the setuid bit is set for the user permission. By means of the setuid bit, all users starting the `passwd` command execute it as `root`.

17.1.2 The setgid Bit

The setuid bit applies to users. However, there is also an equivalent property for groups: the *setgid* bit. A program for which this bit was set runs under the group ID under which it was saved, no matter which user starts it. Therefore, in a directory with the setgid bit, all newly created files and subdirectories are assigned to the group to which the directory belongs. Consider the following example directory:

```
drwxrws--- 2 tux archive 48 Nov 19 17:12  backup
```

You can see the `s` that denotes that the setgid bit is set for the group permission. The owner of the directory and members of the group `archive` may access this directory. Users that are not members of this group are “mapped” to the respective group. The effective group ID of all written files will be `archive`. For example, a backup program that runs with the group ID `archive` is able to access this directory even without root privileges.

17.1.3 The Sticky Bit

There is also the *sticky bit*. It makes a difference whether it belongs to an executable program or a directory. If it belongs to a program, a file marked in this way is loaded to RAM to avoid needing to get it from the hard disk each time it is used. This attribute is used rarely, because modern hard disks are fast enough. If this bit is assigned to a directory, it prevents users from deleting each other's files. Typical examples include the `/tmp` and `/var/tmp` directories:

```
drwxrwxrwt 2 root root 1160 2002-11-19 17:15 /tmp
```

17.2 Advantages of ACLs

Traditionally, three permission sets are defined for each file object on a Linux system. These sets include the read (`r`), write (`w`), and execute (`x`) permissions for each of three types of users—the file owner, the group, and other users. In addition to that, it is possible to set the *set user id*, the *set group id*, and the *sticky* bit. This lean concept is fully adequate for most practical cases. However, for more complex scenarios or advanced applications, system administrators formerly had to use a number of tricks to circumvent the limitations of the traditional permission concept.

ACLs can be used as an extension of the traditional file permission concept. They allow assignment of permissions to individual users or groups even if these do not correspond to the original owner or the owning group. Access control lists are a feature of the Linux kernel and are currently supported by ReiserFS, Ext2, Ext3, JFS, and XFS. Using ACLs, complex scenarios can be realized without implementing complex permission models on the application level.

The advantages of ACLs are evident if you want to replace a Windows server with a Linux server. Some of the connected workstations may continue to run under Windows even after the migration. The Linux system offers file and print services to the Windows clients with Samba. With Samba supporting access control lists, user permissions can be configured both on the Linux server and in Windows with a graphical user interface (only Windows NT and later). With `winbindd`, part of the samba suite, it is even possible to assign permissions to users only existing in the Windows domain without any account on the Linux server.

17.3 Definitions

user class

The conventional POSIX permission concept uses three *classes* of users for assigning permissions in the file system: the owner, the owning group, and other users. Three permission bits can be set for each user class, giving permission to read (`r`), write (`w`), and execute (`x`).

access ACL

The user and group access permissions for all kinds of file system objects (files and directories) are determined by means of access ACLs.

default ACL
: Default ACLs can only be applied to directories. They determine the permissions a file system object inherits from its parent directory when it is created.

ACL entry
: Each ACL consists of a set of ACL entries. An ACL entry contains a type, a qualifier for the user or group to which the entry refers, and a set of permissions. For some entry types, the qualifier for the group or users is undefined.

17.4 Handling ACLs

Table 17.1, “ACL Entry Types” (page 255) summarizes the six possible types of ACL entries, each defining permissions for a user or a group of users. The *owner* entry defines the permissions of the user owning the file or directory. The *owning group* entry defines the permissions of the file's owning group. The superuser can change the owner or owning group with `chown` or `chgrp`, in which case the owner and owning group entries refer to the new owner and owning group. Each *named user* entry defines the permissions of the user specified in the entry's qualifier field. Each *named group* entry defines the permissions of the group specified in the entry's qualifier field. Only the named user and named group entries have a qualifier field that is not empty. The *other* entry defines the permissions of all other users.

The *mask* entry further limits the permissions granted by named user, named group, and owning group entries by defining which of the permissions in those entries are effective and which are masked. If permissions exist in one of the mentioned entries as well as in the mask, they are effective. Permissions contained only in the mask or only in the actual entry are not effective—meaning the permissions are not granted. All permissions defined in the owner and owning group entries are always effective. The example in Table 17.2, “Masking Access Permissions” (page 255) demonstrates this mechanism.

There are two basic classes of ACLs: A *minimum* ACL contains only the entries for the types owner, owning group, and other, which correspond to the conventional permission bits for files and directories. An *extended* ACL goes beyond this. It must contain a mask entry and may contain several entries of the named user and named group types.

Table 17.1 *ACL Entry Types*

Type	Text Form
owner	`user::rwx`
named user	`user:name:rwx`
owning group	`group::rwx`
named group	`group:name:rwx`
mask	`mask::rwx`
other	`other::rwx`

Table 17.2 *Masking Access Permissions*

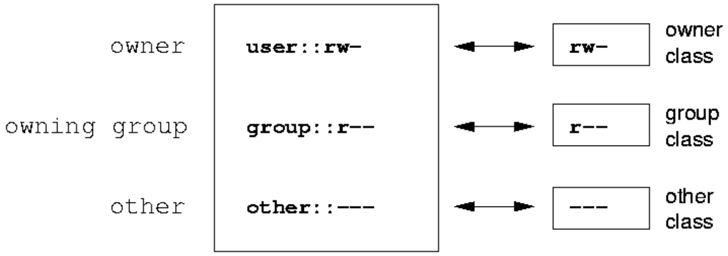
Entry Type	Text Form	Permissions
named user	`user:geeko:r-x`	`r-x`
mask	`mask::rw-`	`rw-`
	effective permissions:	`r--`

17.4.1 ACL Entries and File Mode Permission Bits

Figure 17.1, “Minimum ACL: ACL Entries Compared to Permission Bits” (page 256) and Figure 17.2, “Extended ACL: ACL Entries Compared to Permission Bits” (page 256) illustrate the two cases of a minimum ACL and an extended ACL. The figures are structured in three blocks—the left block shows the type specifications of the ACL entries, the center block displays an example ACL, and the right block shows the respective permission bits according to the conventional permission concept, for example, as displayed by `ls -l`. In both cases, the *owner class* permissions are mapped to the

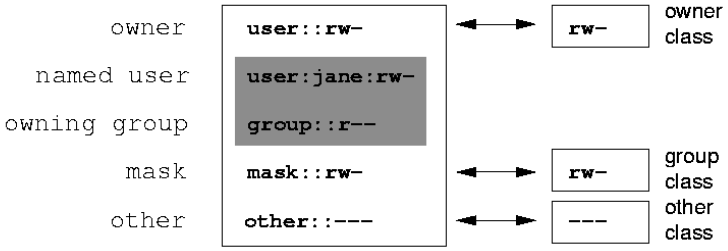
ACL entry owner. *Other class* permissions are mapped to the respective ACL entry. However, the mapping of the *group class* permissions is different in the two cases.

Figure 17.1 *Minimum ACL: ACL Entries Compared to Permission Bits*



In the case of a minimum ACL—without mask—the group class permissions are mapped to the ACL entry owning group. This is shown in Figure 17.1, “Minimum ACL: ACL Entries Compared to Permission Bits” (page 256). In the case of an extended ACL—with mask—the group class permissions are mapped to the mask entry. This is shown in Figure 17.2, “Extended ACL: ACL Entries Compared to Permission Bits” (page 256).

Figure 17.2 *Extended ACL: ACL Entries Compared to Permission Bits*



This mapping approach ensures the smooth interaction of applications, regardless of whether they have ACL support. The access permissions that were assigned by means of the permission bits represent the upper limit for all other “fine adjustments” made with an ACL. Changes made to the permission bits are reflected by the ACL and vice versa.

17.4.2 A Directory with an Access ACL

With `getfacl` and `setfacl` on the command line, you can access ACLs. The usage of these commands is demonstrated in the following example.

Before creating the directory, use the umask command to define which access permissions should be masked each time a file object is created. The command umask 027 sets the default permissions by giving the owner the full range of permissions (0), denying the group write access (2), and giving other users no permissions at all (7). umask actually masks the corresponding permission bits or turns them off. For details, consult the umask man page.

mkdir mydir creates the mydir directory with the default permissions as set by umask. Use ls -dl mydir to check whether all permissions were assigned correctly. The output for this example is:

```
drwxr-x--- ... tux project3 ... mydir
```

With getfacl mydir, check the initial state of the ACL. This gives information like:

```
# file: mydir
# owner: tux
# group: project3
user::rwx
group::r-x
other::---
```

The first three output lines display the name, owner, and owning group of the directory. The next three lines contain the three ACL entries owner, owning group, and other. In fact, in the case of this minimum ACL, the getfacl command does not produce any information you could not have obtained with ls.

Modify the ACL to assign read, write, and execute permissions to an additional user geeko and an additional group mascots with:

```
setfacl -m user:geeko:rwx,group:mascots:rwx mydir
```

The option -m prompts setfacl to modify the existing ACL. The following argument indicates the ACL entries to modify (multiple entries are separated by commas). The final part specifies the name of the directory to which these modifications should be applied. Use the getfacl command to take a look at the resulting ACL.

```
# file: mydir
# owner: tux
# group: project3
user::rwx
user:geeko:rwx
group::r-x
group:mascots:rwx
```

```
mask::rwx
other::---
```

In addition to the entries initiated for the user geeko and the group mascots, a mask entry has been generated. This mask entry is set automatically so that all permissions are effective. setfacl automatically adapts existing mask entries to the settings modified, unless you deactivate this feature with -n. mask defines the maximum effective access permissions for all entries in the group class. This includes named user, named group, and owning group. The group class permission bits displayed by ls -dl mydir now correspond to the mask entry.

```
drwxrwx---+ ... tux project3 ... mydir
```

The first column of the output contains an additional + to indicate that there is an *extended* ACL for this item.

According to the output of the ls command, the permissions for the mask entry include write access. Traditionally, such permission bits would mean that the owning group (here project3) also has write access to the directory mydir. However, the effective access permissions for the owning group correspond to the overlapping portion of the permissions defined for the owning group and for the mask—which is r-x in our example (see Table 17.2, "Masking Access Permissions" (page 255)). As far as the effective permissions of the owning group in this example are concerned, nothing has changed even after the addition of the ACL entries.

Edit the mask entry with setfacl or chmod. For example, use chmod g-w mydir. ls -dl mydir then shows:

```
drwxr-x---+ ... tux project3 ... mydir
```

getfacl mydir provides the following output:

```
# file: mydir
# owner: tux
# group: project3
user::rwx
user:geeko:rwx          # effective: r-x
group::r-x
group:mascots:rwx       # effective: r-x
mask::r-x
other::---
```

After executing the chmod command to remove the write permission from the group class bits, the output of the ls command is sufficient to see that the mask bits must have changed accordingly: write permission is again limited to the owner of mydir.

The output of the getfacl confirms this. This output includes a comment for all those entries in which the effective permission bits do not correspond to the original permissions, because they are filtered according to the mask entry. The original permissions can be restored at any time with chmod g+w mydir.

17.4.3 A Directory with a Default ACL

Directories can have a default ACL, which is a special kind of ACL defining the access permissions that objects in the directory inherit when they are created. A default ACL affects both subdirectories and files.

Effects of a Default ACL

There are two ways in which the permissions of a directory's default ACL are passed to the files and subdirectories:

- A subdirectory inherits the default ACL of the parent directory both as its default ACL and as an access ACL.
- A file inherits the default ACL as its access ACL.

All system calls that create file system objects use a mode parameter that defines the access permissions for the newly created file system object. If the parent directory does not have a default ACL, the permission bits as defined by the umask are subtracted from the permissions as passed by the mode parameter, with the result being assigned to the new object. If a default ACL exists for the parent directory, the permission bits assigned to the new object correspond to the overlapping portion of the permissions of the mode parameter and those that are defined in the default ACL. The umask is disregarded in this case.

Application of Default ACLs

The following three examples show the main operations for directories and default ACLs:

1. Add a default ACL to the existing directory mydir with:

   ```
   setfacl -d -m group:mascots:r-x mydir
   ```

The option `-d` of the `setfacl` command prompts `setfacl` to perform the following modifications (option `-m`) in the default ACL.

Take a closer look at the result of this command:

```
getfacl mydir

# file: mydir
# owner: tux
# group: project3
user::rwx
user:geeko:rwx
group::r-x
group:mascots:rwx
mask::rwx
other::---
default:user::rwx
default:group::r-x
default:group:mascots:r-x
default:mask::r-x
default:other::---
```

`getfacl` returns both the access ACL and the default ACL. The default ACL is formed by all lines that start with `default`. Although you merely executed the `setfacl` command with an entry for the `mascots` group for the default ACL, `setfacl` automatically copied all other entries from the access ACL to create a valid default ACL. Default ACLs do not have an immediate effect on access permissions. They only come into play when file system objects are created. These new objects inherit permissions only from the default ACL of their parent directory.

2. In the next example, use `mkdir` to create a subdirectory in `mydir`, which inherits the default ACL.

```
mkdir mydir/mysubdir

getfacl mydir/mysubdir

# file: mydir/mysubdir
# owner: tux
# group: project3
user::rwx
group::r-x
group:mascots:r-x
mask::r-x
other::---
default:user::rwx
default:group::r-x
default:group:mascots:r-x
```

```
default:mask::r-x
default:other::---
```

As expected, the newly-created subdirectory mysubdir has the permissions from the default ACL of the parent directory. The access ACL of mysubdir is an exact reflection of the default ACL of mydir. The default ACL that this directory will hand down to its subordinate objects is also the same.

3. Use touch to create a file in the mydir directory, for example, touch mydir/myfile. ls -l mydir/myfile then shows:

   ```
   -rw-r-----+ ... tux project3 ... mydir/myfile
   ```

   The output of getfacl mydir/myfile is:

   ```
   # file: mydir/myfile
   # owner: tux
   # group: project3
   user::rw-
   group::r-x          # effective:r--
   group:mascots:r-x   # effective:r--
   mask::r--
   other::---
   ```

   touch uses a mode with the value 0666 when creating new files, which means that the files are created with read and write permissions for all user classes, provided no other restrictions exist in umask or in the default ACL (see Section “Effects of a Default ACL” (page 259)). In effect, this means that all access permissions not contained in the mode value are removed from the respective ACL entries. Although no permissions were removed from the ACL entry of the group class, the mask entry was modified to mask permissions not set in mode.

   This approach ensures the smooth interaction of applications, such as compilers, with ACLs. You can create files with restricted access permissions and subsequently mark them as executable. The mask mechanism guarantees that the right users and groups can execute them as desired.

17.4.4 The ACL Check Algorithm

A check algorithm is applied before any process or application is granted access to an ACL-protected file system object. As a basic rule, the ACL entries are examined in the following sequence: owner, named user, owning group or named group, and other. The access is handled in accordance with the entry that best suits the process. Permissions do not accumulate.

Things are more complicated if a process belongs to more than one group and would potentially suit several group entries. An entry is randomly selected from the suitable entries with the required permissions. It is irrelevant which of the entries triggers the final result “access granted”. Likewise, if none of the suitable group entries contains the required permissions, a randomly selected entry triggers the final result “access denied”.

17.5 ACL Support in Applications

ACLs can be used to implement very complex permission scenarios that meet the requirements of modern applications. The traditional permission concept and ACLs can be combined in a smart manner. The basic file commands (`cp`, `mv`, `ls`, etc.) support ACLs, as do Samba and Konqueror.

Unfortunately, many editors and file managers still lack ACL support. When copying files with Emacs, for instance, the ACLs of these files are lost. When modifying files with an editor, the ACLs of files are sometimes preserved and sometimes not, depending on the backup mode of the editor used. If the editor writes the changes to the original file, the access ACL is preserved. If the editor saves the updated contents to a new file that is subsequently renamed to the old filename, the ACLs may be lost, unless the editor supports ACLs. Except for the star archiver, there are currently no backup applications that preserve ACLs.

17.6 For More Information

Detailed information about ACLs is available at `http://acl.bestbits.at/`. Also see the man pages for `getfacl(1)`, `acl(5)`, and `setfacl(1)`.

Authentication with PAM

18

Linux uses PAM (pluggable authentication modules) in the authentication process as a layer that mediates between user and application. PAM modules are available on a systemwide basis, so they can be requested by any application. This chapter describes how the modular authentication mechanism works and how it is configured.

System administrators and programmers often want to restrict access to certain parts of the system or to limit the use of certain functions of an application. Without PAM, applications must be adapted every time a new authentication mechanism, such as LDAP, Samba or Kerberos, is introduced. This process, however, is rather time-consuming and error-prone. One way to avoid these drawbacks is to separate applications from the authentication mechanism and delegate authentication to centrally managed modules. Whenever a newly required authentication scheme is needed, it is sufficient to adapt or write a suitable PAM module for use by the program in question.

Every program that relies on the PAM mechanism has its own configuration file in the directory `/etc/pam.d/`*`programname`*. These files define the PAM modules used for authentication. In addition, there are global configuration files for PAM modules under `/etc/security`, which define the exact behavior of these modules (examples include `pam_env.conf`, and `time.conf`). Every application that uses a PAM module actually calls a set of PAM functions, which then process the information in the various configuration files and return the result to the calling application.

To facilitate the creation and maintenance of PAM modules, common default configuration files for the `auth`, `account`, `password`, and `session` modules have been introduced. These are pulled in from every application's PAM configuration. Updates to the global PAM configuration modules in `common-*` are thus propagated across all PAM configuration files without requiring the administrator to update every single PAM configuration file.

The global common PAM configuration files are maintained using the pam-config tool. This tool automatically adds new modules to the configuration, changes the configuration of existing ones or deletes modules or options from the configurations. Manual intervention in maintaining PAM configurations is minimized or no longer required.

18.1 Structure of a PAM Configuration File

Each line in a PAM configuration file contains a maximum of four columns:

```
<Type of module> <Control flag> <Module path> <Options>
```

PAM modules are processed as stacks. Different types of modules have different purposes, for example, one module checks the password, another one verifies the location from which the system is accessed, and yet another one reads user-specific settings. PAM knows about four different types of modules:

`auth`
: The purpose of this type of module is to check the user's authenticity. This is traditionally done by querying a password, but it can also be achieved with the help of a chip card or through biometrics (fingerprints or iris scan).

`account`
: Modules of this type check whether the user has general permission to use the requested service. As an example, such a check should be performed to ensure that no one can log in under the username of an expired account.

`password`
: The purpose of this type of module is to enable the change of an authentication token. In most cases, this is a password.

`session`
: Modules of this type are responsible for managing and configuring user sessions. They are started before and after authentication to register login attempts in system

logs and configure the user's specific environment (mail accounts, home directory, system limits, etc.).

The second column contains control flags to influence the behavior of the modules started:

`required`
: A module with this flag must be successfully processed before the authentication may proceed. After the failure of a module with the `required` flag, all other modules with the same flag are processed before the user receives a message about the failure of the authentication attempt.

`requisite`
: Modules having this flag must also be processed successfully, in much the same way as a module with the `required` flag. However, in case of failure a module with this flag gives immediate feedback to the user and no further modules are processed. In case of success, other modules are subsequently processed, just like any modules with the `required` flag. The `requisite` flag can be used as a basic filter checking for the existence of certain conditions that are essential for a correct authentication.

`sufficient`
: After a module with this flag has been successfully processed, the calling application receives an immediate message about the success and no further modules are processed, provided there was no preceding failure of a module with the `required` flag. The failure of a module with the `sufficient` flag has no direct consequences, in the sense that any subsequent modules are processed in their respective order.

`optional`
: The failure or success of a module with this flag does not have any direct consequences. This can be useful for modules that are only intended to display a message (for example, to tell the user that mail has arrived) without taking any further action.

include
: If this flag is given, the file specified as argument is inserted at this place.

The module path does not need to be specified explicitly, as long as the module is located in the default directory `/lib/security` (for all 64-bit platforms supported by openSUSE®, the directory is `/lib64/security`). The fourth column may contain an option for the given module, such as debug (enables debugging) or nullok (allows the use of empty passwords).

18.2 The PAM Configuration of sshd

To show how the theory behind PAM works, consider the PAM configuration of sshd as a practical example:

Example 18.1 *PAM Configuration for sshd*

```
#%PAM-1.0
auth     include         common-auth
auth     required        pam_nologin.so
account  include         common-account
password include         common-password
session  include         common-session
# Enable the following line to get resmgr support for
# ssh sessions (see /usr/share/doc/packages/resmgr/README.SuSE)
#session  optional       pam_resmgr.so fake_ttyname
```

The typical PAM configuration of an application (sshd, in this case) contains four include statements referring to the configuration files of four module types: common-auth, common-account, common-password, and common-session. These four files hold the default configuration for each module type. By including them instead of calling each module separately for each PAM application, automatically get an updated PAM configuration if the administrator changes the defaults. In former times, you had to adjust all configuration files manually for all applications when changes to PAM occurred or a new application was installed. Now the PAM configuration is made with central configuration files and all changes are automatically inherited by the PAM configuration of each service.

The first include file (common-auth) calls two modules of the auth type: pam_env and pam_unix2. See Example 18.2, "Default Configuration for the auth Section" (page 266).

Example 18.2 *Default Configuration for the auth Section*

```
auth    required        pam_env.so
auth    required        pam_unix2.so
```

The first one, pam_env, loads the file /etc/security/pam_env.conf to set the environment variables as specified in this file. This can be used to set the DISPLAY variable to the correct value, because the pam_env module knows about the location from which the login is taking place. The second one, pam_unix2, checks the user's login and password against /etc/passwd and /etc/shadow.

After the modules specified in common-auth have been successfully called, a third module called pam_nologin checks whether the file /etc/nologin exists. If it does, no user other than root may log in. The whole stack of auth modules is processed before sshd gets any feedback about whether the login has succeeded. Given that all modules of the stack have the required control flag, they must all be processed successfully before sshd receives a message about the positive result. If one of the modules is not successful, the entire module stack is still processed and only then is sshd notified about the negative result.

As soon as all modules of the auth type have been successfully processed, another include statement is processed, in this case, that in Example 18.3, "Default Configuration for the account Section" (page 267). common-account contains just one module, pam_unix2. If pam_unix2 returns the result that the user exists, sshd receives a message announcing this success and the next stack of modules (password) is processed, shown in Example 18.4, "Default Configuration for the password Section" (page 267).

Example 18.3 *Default Configuration for the account Section*

```
account required        pam_unix2.so
```

Example 18.4 *Default Configuration for the password Section*

```
password required       pam_pwcheck.so  nullok cracklib
password required       pam_unix2.so    nullok use_authtok
#password required      pam_make.so     /var/yp
```

Again, the PAM configuration of sshd involves just an include statement referring to the default configuration for password modules located in common-password. These modules must successfully be completed (control flag required) whenever the application requests the change of an authentication token. Changing a password or another authentication token requires a security check. This is achieved with the pam_pwcheck module. The pam_unix2 module used afterwards carries over any old and new passwords from pam_pwcheck, so the user does not need to authenticate again. This also makes it impossible to circumvent the checks carried out by pam_pwcheck. The modules of the password type should be used wherever the preceding modules of the account or the auth type are configured to complain about an expired password.

Example 18.5 *Default Configuration for the session Section*

```
session required        pam_limits.so
session required        pam_unix2.so
session optional        pam_umask.so
```

As the final step, the modules of the `session` type, bundled in the `common-session` file are called to configure the session according to the settings for the user in question. The `pam_limits` module loads the file `/etc/security/limits.conf`, which may define limits on the use of certain system resources. The `pam_unix2` module is processed again. The `pam_umask` module can be used to set the file mode creation mask. Since this module carries the `optional` flag, a failure of this module would not affect the successful completion of the entire session module stack. The `session` modules are called a second time when the user logs out.

18.3 Configuring PAM Using pam-config

The pam-config tool helps you configure the global PAM configuration files under `/etc/pam.d/common-*-pc`. Use the `pam-config` command to maintain your PAM configuration files. Add new modules to your PAM configurations, delete other modules or modify options to these modules. As these changes concern only the global PAM configuration files, no manual tweaking of the PAM setup for individual applications is required.

A simple real-world use case for pam-config would involve the following:

1. **Auto-generate a fresh Unix-style PAM configuration.** Let pam-config create the simplest possible setup which you can extend later on. The `pam-config --create` command creates a simple UNIX authentication configuration. Preexisting configuration files not maintained by pam-config are overwritten, but backup copies are kept as `*.pam-config-backup`.

2. **Add a new authentication method.** Adding a new authentication method (e.g. LDAP) to your stack of PAM modules comes down to a simple `pam-config --add --ldap` command. LDAP is added wherever appropriate across all `common-*-pc` PAM configuration files.

3. **Add debugging for test purposes.** To make sure the new authentication procedure works as planned, turn on debugging for all PAM-related operations. The `pam-config --add --ldap-debug` turns on debugging for LDAP-related PAM operations. Find the debugging output in `/var/log/messages`.

4. **Query your setup.** Before you finally apply your new PAM setup, check whether it contains all the options you planned to add. The `pam-config --query --`*`module`* lists both the type and the options for the queried PAM module.

5. **Remove the debug options.** Finally, remove the debug option from your setup when you are entirely satisfied with the performance of it. The `pam-config --delete --ldap-debug` turns of debugging for LDAP authentication. In case you had debugging options added for other modules, use similar commands to turn these off.

When you create your PAM configuration files from scratch using the `pam-config --create` command, it creates symbolic links from the `common-*` to the `common-*-pc` files. pam-config only modifies the `common-*-pc` configuration files. Removing these symbolic links effectively disable pam-config, because pam-config only operates on the `common-*-pc` files and these files are not put into effect without the symbolic links.

For more information on the `pam-config` command and the options available, refer to the manual page of `pam-config`, `pam-config(8).`

18.4 For More Information

In the directory `/usr/share/doc/packages/pam` of your installed system, find the following additional documentation:

READMEs
: In the top level of this directory, there are some general README files. The subdirectory `modules` holds README files about the available PAM modules.

The Linux-PAM System Administrators' Guide
: This document includes everything that a system administrator should know about PAM. It discusses a range of topics, from the syntax of configuration files to the

security aspects of PAM. The document is available as a PDF file, in HTML format, and as plain text.

The Linux-PAM Module Writers' Manual
: This document summarizes the topic from the developer's point of view, with information about how to write standard-compliant PAM modules. It is available as a PDF file, in HTML format, and as plain text.

The Linux-PAM Application Developers' Guide
: This document includes everything needed by an application developer who wants to use the PAM libraries. It is available as a PDF file, in HTML format, and as plain text.

The PAM Manual Pages
: PAM in general as well as the individual modules come with manual pages that provide a good overview of the functionality provided by the respective component.

Thorsten Kukuk has developed a number of PAM modules and made some information available about them at `http://www.suse.de/~kukuk/pam/`.

Working with the Shell 19

Although graphical user interfaces have become very important and user-friendly, using them is not the only way to communicate with your system. A command line interpreter, in Unix/Linux called a *shell*, provides a highly flexible and efficient means for text-oriented communication with your system.

In administration, shell-based applications are especially important for controlling computers over slow network links or if you want to perform tasks as administrator on the command line.

This chapter deals with a couple of basics you need to know for making efficient use of the command line: the directory structure of Linux, the user and permission concept of Linux, an overview of important shell commands, and a short introduction to the vi editor, which is a default editor always available in Unix and Linux systems.

19.1 Using the Bash Shell

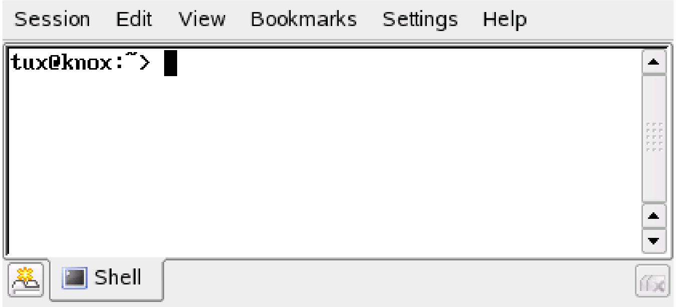
For UNIX or Linux several shells are available which differ slightly in behavior and in the commands they accept. The default shell in openSUSE® is Bash (GNU Bourne-Again Shell).

If you are logged in to a graphical user interface, you can start a (login) shell parallel to the user interface or in a terminal window within the graphical user interface. Press Ctrl + Alt + F2 to leave the graphical user interface and access a login shell. After login, the prompt shows your login name followed by `@` and the hostname of your computer. The hostname is followed by a colon and the path to the current directory. If you have logged in as system administrator, `root`, Bash indicates this with a hash symbol, `#`. Directly after login, the current directory is usually the home directory of the user account with which you have logged in, indicated by the tilde symbol, ~. When you are logged in on a remote computer the information provided by the prompt always shows you which system you are currently working on. You can now enter commands and execute tasks. To log out from the shell, enter `exit` and press Alt + F7 to switch back to the graphical user interface. You will find your desktop and the applications running on it unchanged.

To start a terminal window *within* the graphical user interface in KDE or GNOME press Alt + F2 and enter `xterm` (or click the Konsole or GNOME terminal icon in the panel). This opens a terminal window on your desktop. As you are already logged in to your desktop the prompt shows the usual login and path information. You can now enter commands and execute tasks just like in any shell which runs parallel to your desktop. To close the terminal window press Alt + F4 .

The Konsole or the GNOME Terminal window appears, displaying the prompt at the first line, see Figure 19.1, "Example of a Bash Terminal Window" (page 272). The prompt usually shows your login name (in this example, `tux`), the hostname of your computer (here, `knox`), and the current path (in this case, your home directory, indicated by the tilde symbol, ~). When you are logged in on a remote computer this information always shows you which system you are currently working on. When the cursor placed behind this prompt, you can send commands directly to your computer system.

***Figure 19.1** Example of a Bash Terminal Window*



```
tux@knox:~>
```

Because the shell does not offer a graphical overview of directories and files like the tree view in a file manager, it is useful to have some basic knowledge of the default directory structure in Linux.

19.1.1 The Directory Structure

The following table provides a short overview of the most important higher-level directories you find on a Linux system. Find more detailed information about the directories and important subdirectories in the following list.

Table 19.1 *Overview of a Standard Directory Tree*

Directory	Contents
`/`	Root directory—the starting point of the directory tree.
`/bin`	Essential binary files, such as commands that are needed by both the system administrator and normal users. Usually also contains the shells, such as Bash.
`/boot`	Static files of the boot loader.
`/dev`	Files needed to access host-specific devices.
`/etc`	Host-specific system configuration files.
`/home`	Holds the home directories of all users who have an account on the system. Only `root`'s home directory is not located in `/home` but in `/root`.
`/lib`	Essential shared libraries and kernel modules.
`/media`	Mount points for removable media.
`/mnt`	Mount point for temporarily mounting a file system.
`/opt`	Add-on application software packages.
`/root`	Home directory for the superuser `root`.
`/sbin`	Essential system binaries.
`/srv`	Data for services provided by the system.

Directory	Contents
`/tmp`	Temporary files.
`/usr`	Secondary hierarchy with read-only data.
`/var`	Variable data such as log files
`/windows`	Only available if you have both Microsoft Windows* and Linux installed on your system. Contains the Windows data.

The following list provides more detailed information and gives some examples which files and subdirectories can be found in the directories:

`/bin`
: Contains the basic shell commands that may be used both by `root` and by other users. These commands include `ls`, `mkdir`, `cp`, `mv`, `rm`, and `rmdir`. `/bin` also contains Bash, the default shell in openSUSE.

`/boot`
: Contains data required for booting, such as the boot loader, the kernel, and other data that is used before the kernel begins executing user mode programs.

`/dev`
: Holds device files that represent hardware components.

`/etc`
: Contains local configuration files that control the operation of programs like the X Window System. The `/etc/init.d` subdirectory contains scripts that are executed during the boot process.

`/home/`*`username`*
: Holds the private data of every user who has an account on the system. The files located here can only be modified by their owner or by the system administrator. By default, your e-mail directory and personal desktop configuration are located here in form of hidden files and directories. KDE users find the personal configuration data for their desktop in `.kde`, GNOME users find it in `.gconf`. For information about hidden files, refer to Section “Key Features” (Chapter 7, *Basic Concepts*, ↑Start-Up).

NOTE: Home Directory in a Network Environment

If you are working in a network environment, your home directory may be mapped to a directory in the file system other than `/home`.

`/lib`
: Contains essential shared libraries needed to boot the system and to run the commands in the root file system. The Windows equivalent for shared libraries are DLL files.

`/media`
: Contains mount points for removable media, such as CD-ROMs, USB sticks, and digital cameras (if they use USB). `/media` generally holds any type of drive except the hard drive of your system. As soon as your removable medium has been inserted or connected to the system and has been mounted, you can access it from here.

`/mnt`
: This directory provides a mount point for a temporarily mounted file system. `root` may mount file systems here.

`/opt`
: Reserved for the installation of additional software. Optional software and larger add-on program packages can be found there. KDE3 is located there, whereas KDE4 and GNOME have moved to `/usr` now.

`/root`
: Home directory for the `root` user. Personal data of `root` is located here.

`/sbin`
: As the `s` indicates, this directory holds utilities for the superuser. `/sbin` contains binaries essential for booting, restoring, and recovering the system in addition to the binaries in `/bin`.

`/srv`
: Holds data for services provided by the system, such as FTP and HTTP.

`/tmp`
: This directory is used by programs that require temporary storage of files.

`/usr`

`/usr` has nothing to do with users, but is the acronym for UNIX system resources. The data in `/usr` is static, read-only data that can be shared among various hosts compliant to the Filesystem Hierarchy Standard (FHS). This directory contains all application programs and establishes a secondary hierarchy in the file system. `/usr` holds a number of subdirectories, such as `/usr/bin`, `/usr/sbin`, `/usr/local`, and `/usr/share/doc`.

`/usr/bin`

Contains generally accessible programs.

`/usr/sbin`

Contains programs reserved for the system administrator, such as repair functions.

`/usr/local`

In this directory, the system administrator can install local, distribution-independent extensions.

`/usr/share/doc`

Holds various documentation files and the release notes for your system. In the `manual` subdirectory, find an online version of this manual. If more than one language is installed, this directory may contain versions of the manuals for different languages.

Under `packages`, find the documentation included in the software packages installed on your system. For every package, a subdirectory `/usr/share/doc/packages/`*`packagename`* is created that often holds README files for the package and sometimes examples, configuration files, or additional scripts.

If HOWTOs are installed on your system `/usr/share/doc` also holds the `howto` subdirectory in which to find additional documentation on many tasks relating to the setup and operation of Linux software.

`/var`

Whereas `/usr` holds static, read-only data, `/var` is for data which is written during system operation and thus is variable data, such as log files or spooling data. For example, the log files of your system are in `/var/log/messages` (only accessible for `root`).

`/windows`

Only available if you have both Microsoft Windows and Linux installed on your system. Contains the Windows data available on the Windows partition of your system. Whether you can edit the data in this directory depends on the file system your Windows partition uses. If it is FAT32, you can open and edit the files in this directory. For an NTFS file system, however, you can only read your Windows files from Linux, but not modify them. Learn more in Section “Accessing Files on Different OS on the Same Computer” (Chapter 11, *Copying and Sharing Files*, ↑Start-Up).

19.2 Users and Access Permissions

Since its inception in the early 1990s, Linux has been developed as a multiuser system. Any number of users can work on it simultaneously. Users need to log in to the system before starting a session at their workstations. Each user has a username with a corresponding password. This differentiation of users guarantees that unauthorized users cannot see files for which they do not have permission. Larger changes to the system, such as installing new programs, are also usually impossible or restricted for normal users. Only the root user, or *super user*, has the unrestricted capacity to make changes to the system and unlimited access to all files. Those who use this concept wisely, only logging in with full `root` access when necessary, can cut back the risk of unintentional loss of data. Because under normal circumstances only root can delete system files or format hard disks, the threat from the *Trojan horse effect* or from accidentally entering destructive commands can be significantly reduced.

19.2.1 File System Permissions

Basically, every file in a Linux file system belongs to a user and a group. Both of these proprietary groups and all others can be authorized to write, read, or execute these files.

A group, in this case, can be defined as a set of connected users with certain collective rights. For example, call a group working on a certain project `project3`. Every user in a Linux system is a member of at least one proprietary group, normally `users`. There can be as many groups in a system as needed, but only `root` is able to add groups. Every user can find out, with the command `groups`, of which groups he is a member.

File Access

The organization of permissions in the file system differs for files and directories. File permission information can be displayed with the command `ls -l`. The output could appear as in Example 19.1, “Sample Output Showing File Permissions” (page 278).

Example 19.1 *Sample Output Showing File Permissions*

```
-rw-r----- 1 tux project3 14197 Jun 21  15:03 Roadmap
```

As shown in the third column, this file belongs to user `tux`. It is assigned to the group `project3`. To discover the user permissions of the `Roadmap` file, the first column must be examined more closely.

-	rw-	r--	---
Type	Users Permissions	Group Permissions	Permissions for Other Users

This column consists of one leading character followed by nine characters grouped in threes. The first of the ten letters stands for the type of file system component. The hyphen (`-`) shows that this is a file. A directory (`d`), a link (`l`), a block device (`b`), or a character device could also be indicated.

The next three blocks follow a standard pattern. The first three characters refer to whether the file is readable (`r`) or not (`-`). A `w` in the middle portion symbolizes that the corresponding object can be edited and a hyphen (`-`) means it is not possible to write to the file. An `x` in the third position denotes that the object can be executed. Because the file in this example is a text file and not one that is executable, executable access for this particular file is not needed.

In this example, `tux` has, as owner of the file `Roadmap`, read (`r`) and write access (`w`) to it, but cannot execute it (`x`). The members of the group `project3` can read the file, but they cannot modify it or execute it. Other users do not have any access to this file. Other permissions can be assigned by means of ACLs (access control lists).

Directory Permissions

Access permissions for directories have the type `d`. For directories, the individual permissions have a slightly different meaning.

Example 19.2 *Sample Output Showing Directory Permissions*

```
drwxrwxr-x 1 tux project3 35 Jun 21 15:15  ProjectData
```

In Example 19.2, "Sample Output Showing Directory Permissions" (page 279), the owner (tux) and the owning group (project3) of the directory ProjectData are easy to recognize. In contrast to the file access permissions from File Access (page 278), the set reading permission (r) means that the contents of the directory can be shown. The write permission (w) means that new files can be created. The executable permission (x) means that the user can change to this directory. In the above example, the user tux as well as the members of the group project3 can change to the ProjectData directory (x), view the contents (r), and add or delete files (w). The rest of the users, on the other hand, are given less access. They may enter the directory (x) and browse through it (r), but not insert any new files (w).

19.2.2 Modifying File Permissions

Changing Access Permissions

The access permissions of a file or directory can be changed by the owner and, of course, by root with the command chmod followed by the parameters changing the permissions and one or more filenames. The parameters form different categories:

1. Users concerned

- u (*user*)—owner of the file
- g (*group*)—group that owns the file
- o (*others*)—additional users (if no parameter is given, the changes apply to all categories)

2. A character for deletion (–), setting (=), or insertion (+)

3. The abbreviations

- r—*read*
- w—*write*

- x—*execute*

4. Filename or filenames separated by spaces

If, for example, the user tux in Example 19.2, “Sample Output Showing Directory Permissions” (page 279) also wants to grant other users write (w) access to the directory ProjectData, he can do this using the command chmod o+w ProjectData.

If, however, he wants to deny all users other than himself write permissions, he can do this by entering the command chmod go-w ProjectData. To prohibit all users from adding a new file to the folder ProjectData, enter chmod -w ProjectData. Now, not even the owner can create a new file in the directory without first reestablishing write permissions.

Changing Ownership Permissions

Other important commands to control the ownership and permissions of the file system components are chown (change owner) and chgrp (change group). The command chown can be used to transfer ownership of a file to another user. However, only root is permitted to perform this change.

Suppose the file Roadmap from Example 19.2, “Sample Output Showing Directory Permissions” (page 279) should no longer belong to tux, but to the user geeko. root should then enter chown geeko Roadmap.

chgrp changes the group ownership of the file. However, the owner of the file must be a member of the new group. In this way, the user tux from Example 19.1, “Sample Output Showing File Permissions” (page 278) can switch the group owning the file ProjectData to project4 with the command chgrp project4 ProjectData, as long as he is a member of this new group.

19.3 Important Linux Commands

This section gives insight into the most important commands. There are many more commands than listed in this chapter. Along with the individual commands, parameters are listed and, where appropriate, a typical sample application is introduced. To learn more about the various commands, use the manual pages, accessed with man followed by the name of the command, for example, man ls.

In the man pages, move up and down with PgUp and PgDn. Move between the beginning and the end of a document with Home and End. End this viewing mode by pressing Q. Learn more about the `man` command itself with `man man`.

In the following overview, the individual command elements are written in different typefaces. The actual command and its mandatory options are always printed as `command option`. Specifications or parameters that are not required are placed in `[square brackets]`.

Adjust the settings to your needs. It makes no sense to write `ls file` if no file named `file` actually exists. You can usually combine several parameters, for example, by writing `ls -la` instead of `ls -l -a`.

19.3.1 File Commands

The following section lists the most important commands for file management. It covers anything from general file administration to manipulation of file system ACLs.

File Administration

`ls [options] [files]`
: If you run `ls` without any additional parameters, the program lists the contents of the current directory in short form.

    `-l`
    : Detailed list

    `-a`
    : Displays hidden files

`cp [options] source target`
: Copies `source` to `target`.

    -i
    : Waits for confirmation, if necessary, before an existing `target` is overwritten

    -r
    : Copies recursively (includes subdirectories)

`mv [options] source target`

Copies `source` to `target` then deletes the original `source`.

-b

Creates a backup copy of the `source` before moving

-i

Waits for confirmation, if necessary, before an existing `targetfile` is overwritten

`rm [options] files`

Removes the specified files from the file system. Directories are not removed by `rm` unless the option `-r` is used.

`-r`

Deletes any existing subdirectories

`-i`

Waits for confirmation before deleting each file

`ln [options] source target`

Creates an internal link from `source` to `target`. Normally, such a link points directly to `source` on the same file system. However, if `ln` is executed with the `-s` option, it creates a symbolic link that only points to the directory in which `source` is located, enabling linking across file systems.

-s

Creates a symbolic link

`cd [options] [directory]`

Changes the current directory. `cd` without any parameters changes to the user's home directory.

`mkdir [options] directory`

Creates a new directory.

`rmdir [options] directory`

Deletes the specified directory if it is already empty.

`chown [options] username[:[group]] files`

Transfers ownership of a file to the user with the specified username.

`-R`

Changes files and directories in all subdirectories

`chgrp [options] groupname files`

Transfers the group ownership of a given `file` to the group with the specified group name. The file owner can only change group ownership if a member of both the current and the new group.

`chmod [options] mode files`

Changes the access permissions.

The `mode` parameter has three parts: `group`, `access`, and `access type`. `group` accepts the following characters:

u

User

g

Group

o

Others

For `access`, grant access with `+` and deny it with `-`.

The `access type` is controlled by the following options:

r

Read

w

Write

x

Execute—executing files or changing to the directory

s

Setuid bit—the application or program is started as if it were started by the owner of the file

As an alternative, a numeric code can be used. The four digits of this code are composed of the sum of the values 4, 2, and 1—the decimal result of a binary mask. The first digit sets the set user ID (SUID) (4), the set group ID (2), and the sticky (1) bits. The second digit defines the permissions of the owner of the file. The third digit defines the permissions of the group members and the last digit sets the permissions for all other users. The read permission is set with 4, the write permission with 2, and the permission for executing a file is set with 1. The owner of a file would usually receive a 6 or a 7 for executable files.

`gzip [parameters] files`

This program compresses the contents of files using complex mathematical algorithms. Files compressed in this way are given the extension `.gz` and need to be uncompressed before they can be used. To compress several files or even entire directories, use the `tar` command.

-d

Decompresses the packed gzip files so they return to their original size and can be processed normally (like the command `gunzip`)

`tar options archive files`

`tar` puts one or more files into an archive. Compression is optional. `tar` is a quite complex command with a number of options available. The most frequently used options are:

`-f`

Writes the output to a file and not to the screen as is usually the case

`-c`

Creates a new tar archive

`-r`

Adds files to an existing archive

`-t`

Outputs the contents of an archive

`-u`

Adds files, but only if they are newer than the files already contained in the archive

`-x`
: Unpacks files from an archive (*extraction*)

`-z`
: Packs the resulting archive with `gzip`

`-j`
: Compresses the resulting archive with `bzip2`

`-v`
: Lists files processed

The archive files created by `tar` end with `.tar`. If the tar archive was also compressed using `gzip`, the ending is `.tgz` or `.tar.gz`. If it was compressed using `bzip2`, the ending is `.tar.bz2`.

`locate patterns`
: This command is only available if you have installed the `findutils-locate` package. The `locate` command can find in which directory a specified file is located. If desired, use wild cards to specify filenames. The program is very speedy, because it uses a database specifically created for the purpose (rather than searching through the entire file system). This very fact, however, also results in a major drawback: locate is unable to find any files created after the latest update of its database. The database can be generated by `root` with `updatedb`.

`updatedb [options]`
: This command performs an update of the database used by `locate`. To include files in all existing directories, run the program as `root`. It also makes sense to place it in the background by appending an ampersand (`&`), so you can immediately continue working on the same command line (`updatedb &`). This command usually runs as a daily cron job (see `cron.daily`).

`find [options]`
: With `find`, search for a file in a given directory. The first argument specifies the directory in which to start the search. The option `-name` must be followed by a search string, which may also include wild cards. Unlike `locate`, which uses a database, `find` scans the actual directory.

Commands to Access File Contents

`file [options] [files]`

With `file`, detect the contents of the specified files.

-z

Tries to look inside compressed files

`cat [options] files`

The `cat` command displays the contents of a file, printing the entire contents to the screen without interruption.

-n

Numbers the output on the left margin

`less [options] files`

This command can be used to browse the contents of the specified file. Scroll half a screen page up or down with PgUp and PgDn or a full screen page down with Space. Jump to the beginning or end of a file using Home and End. Press Q to exit the program.

`grep [options] searchstring files`

The grep command finds a specific search string in the specified files. If the search string is found, the command displays the line in which `searchstring` was found along with the filename.

`-i`

Ignores case

`-H`

Only displays the names of the respective files, but not the text lines

`-n`

Additionally displays the numbers of the lines in which it found a hit

`-l`

Only lists the files in which `searchstring` does not occur `diff [options] file1 file2`

The `diff` command compares the contents of any two files. The output produced by the program lists all lines that do not match. This is frequently used by programmers who need only send their program alterations and not the entire source code.

`-q`

Only reports whether the two files differ

`-u`

Produces a “unified” diff, which makes the output more readable

File Systems

`mount [options] [device] mountpoint`

This command can be used to mount any data media, such as hard disks, CD-ROM drives, and other drives, to a directory of the Linux file system.

`-r`

Mount read-only

`-t filesystem`

Specify the file system, commonly `ext2` for Linux hard disks, `msdos` for MS-DOS media, `vfat` for the Windows file system, and `iso9660` for CDs

For hard disks not defined in the file `/etc/fstab`, the device type must also be specified. In this case, only `root` can mount it. If the file system should also be mounted by other users, enter the option `user` in the appropriate line in the `/etc/fstab` file (separated by commas) and save this change. Further information is available in the `mount(1)` man page.

`umount [options] mountpoint`

This command unmounts a mounted drive from the file system. To prevent data loss, run this command before taking a removable data medium from its drive. Normally, only `root` is allowed to run the commands `mount` and `umount`. To enable other users to run these commands, edit the `/etc/fstab` file to specify the option `user` for the respective drive.

19.3.2 System Commands

The following section lists a few of the most important commands needed for retrieving system information and controlling processes and the network.

System Information

`df [options] [directory]`

The `df` (disk free) command, when used without any options, displays information about the total disk space, the disk space currently in use, and the free space on all the mounted drives. If a directory is specified, the information is limited to the drive on which that directory is located.

`-h`

Shows the number of occupied blocks in gigabytes, megabytes, or kilobytes—in human-readable format

`-T`

Type of file system (ext2, nfs, etc.)

`du [options] [path]`

This command, when executed without any parameters, shows the total disk space occupied by files and subdirectories in the current directory.

`-a`

Displays the size of each individual file

`-h`

Output in human-readable form

`-s`

Displays only the calculated total size

`free [options]`

The command `free` displays information about RAM and swap space usage, showing the total and the used amount in both categories. See Section 14.1.6, "The free Command" (page 220) for more information.

`-b`

Output in bytes

`-k`

Output in kilobytes

`-m`

Output in megabytes

`date [options]`

This simple program displays the current system time. If run as `root`, it can also be used to change the system time. Details about the program are available in the date(1) man page.

Processes

`top [options]`

`top` provides a quick overview of the currently running processes. Press H to access a page that briefly explains the main options for customizing the program.

`ps [options] [process ID]`

If run without any options, this command displays a table of all your own programs or processes—those you started. The options for this command are not preceded by hyphen.

aux

Displays a detailed list of all processes, independent of the owner

`kill [options] process ID`

Unfortunately, sometimes a program cannot be terminated in the normal way. In most cases, you should still be able to stop such a runaway program by executing the `kill` command, specifying the respective process ID (see `top` and `ps`). `kill` sends a *TERM* signal that instructs the program to shut itself down. If this does not help, the following parameter can be used:

-9

Sends a *KILL* signal instead of a *TERM* signal, bringing the specified process to an end in almost all cases

`killall [options] processname`

This command is similar to `kill`, but uses the process name (instead of the process ID) as an argument, killing all processes with that name.

Network

`ping [options] hostname or IP address`

The `ping` command is the standard tool for testing the basic functionality of TCP/IP networks. It sends a small data packet to the destination host, requesting an immediate reply. If this works, `ping` displays a message to that effect, which indicates that the network link is basically functioning.

`-c` *`number`*

Determines the total number of packages to send and ends after they have been dispatched (by default, there is no limitation set)

`-f`

flood ping: sends as many data packages as possible; a popular means, reserved for `root`, to test networks

`-i` *`value`*

Specifies the interval between two data packages in seconds (default: one second)

`nslookup`

The domain name system resolves domain names to IP addresses. With this tool, send queries to name servers (DNS servers).

`telnet [options] hostname or IP address [port]`

Telnet is actually an Internet protocol that enables you to work on remote hosts across a network. telnet is also the name of a Linux program that uses this protocol to enable operations on remote computers.

WARNING

Do not use telnet over a network on which third parties can "eavesdrop." Particularly on the Internet, use encrypted transfer methods, such as `ssh`, to avoid the risk of malicious misuse of a password (see the man page for `ssh`).

Miscellaneous

`passwd [options] [username]`
Users may change their own passwords at any time using this command. The administrator `root` can use the command to change the password of any user on the system.

`su [options] [username]`
The `su` command makes it possible to log in under a different username from a running session. Specify a username and the corresponding password. The password is not required from `root`, because `root` is authorized to assume the identity of any user. When using the command without specifying a username, you are prompted for the `root` password and change to the superuser (`root`). Use `su -` to start a login shell for a different user.

`halt [options]`
To avoid loss of data, you should always use this program to shut down your system.

`reboot [options]`
Does the same as `halt` except the system performs an immediate reboot.

`clear`
This command cleans up the visible area of the console. It has no options.

19.3.3 For More Information

There are many more commands than listed in this chapter. For information about other commands or more detailed information, the O'Reilly publication *Linux in a Nutshell* is recommended.

19.4 The vi Editor

Text editors are still used for many system administration tasks as well as for programming. In the world of Unix, vi stands out as an editor that offers comfortable editing functions and is more ergonomic than many editors with mouse support.

19.4.1 Operating Modes

NOTE: Display of Keys

In the following, find several commands that you can enter in vi by just pressing keys. These appear in uppercase as on a keyboard. If you need to enter a key in uppercase, this is stated explicitly by showing a key combination including the Shift key.

Basically, vi makes use of three operating modes: *insert* mode, *command* mode, and *extended* mode. The keys have different functions depending on the mode. On start-up, vi is normally set to the *command* mode. The first thing to learn is how to switch between the modes:

Command Mode to Insert Mode

There are many possibilities, including A for append, I for insert, or O for a new line under the current line.

Insert Mode to Command Mode

Press Esc to exit the *insert* mode. vi cannot be terminated in *insert* mode, so it is important to get used to pressing Esc.

Command Mode to Extended Mode

The *extended* mode of vi can be activated by entering a colon (:). The *extended* or *ex* mode is similar to an independent line-oriented editor that can be used for various simple and more complex tasks.

Extended Mode to Command Mode

After executing a command in *extended* mode, the editor automatically returns to *command* mode. If you decide not to execute any command in *extended* mode, delete the colon with <—. The editor returns to *command* mode.

It is not possible to switch directly from *insert* mode to *extended* mode without first switching to *command* mode.

vi, like other editors, has its own procedure for terminating the program. You cannot terminate vi while in *insert* mode. First, exit *insert* mode by pressing Esc. Subsequently, you have two options:

1. *Exit without saving:* To terminate the editor without saving the changes, enter : – Q – ! in *command* mode. The exclamation mark (!) causes vi to ignore any changes.

2. *Save and exit:* There are several possibilities to save your changes and terminate the editor. In *command* mode, use Shift + Z Shift + Z. To exit the program saving all changes using the *extended* mode, enter : – W – Q. In *extended* mode, `w` stands for write and `q` for quit.

19.4.2 vi in Action

vi can be used as a normal editor. In *insert* mode, enter text and delete text with the <— and Del keys. Use the arrow keys to move the cursor.

However, these control keys often cause problems, because there are many terminal types that use special key codes. This is where the *command* mode comes into play. Press Esc to switch from *insert* mode to *command* mode. In *command* mode, move the cursor with H, J, K, and L. The keys have the following functions:

H
: Move one character to the left

J
: Move one line down

K
: Move one line up

L
: Move one character to the right

The commands in *command* mode allow diverse variations. To execute a command several times, simply enter the number of repetitions before entering the actual command. For example, enter 5 L to move the cursor five characters to the right.

A selection of important commands is shown in Table 19.2, “Simple Commands of the vi Editor” (page 294) This list is far from complete. More complete lists are available in the documentation found in Section 19.4.3, “For More Information” (page 295)

Table 19.2 *Simple Commands of the vi Editor*

Esc	Change to command mode
I	Change to insert mode (characters appear at the current cursor position)
A	Change to insert mode (characters are inserted after the current cursor position)
Shift + A	Change to insert mode (characters are added at the end of the line)
Shift + R	Change to replace mode (overwrite the old text)
R	Replace the character under the cursor
O	Change to insert mode (a new line is inserted after the current one)
Shift + O	Change to insert mode (a new line is inserted before the current one)
X	Delete the current character
D – D	Delete the current line
D – W	Delete up to the end of the current word
C – W	Change to insert mode (the rest of the current word is overwritten by the next entries you make)
U	Undo the last command
Ctrl + R	Redo the change that was undone
Shift + J	Join the following line with the current one
.	Repeat the last command

19.4.3 For More Information

vi supports a wide range of commands. It enables the use of macros, shortcuts, named buffers, and many other useful features. A detailed description of the various options would exceed the scope of this manual. openSUSE comes with vim (vi improved), an improved version of vi. There are numerous information sources for this application:

- vimtutor is an interactive tutor for vim.
- In vim, enter the command `:help` to get help for many subjects.
- A book about vim is available online at http://www.truth.sk/vim/vimbook-OPL.pdf.
- The Web pages of the vim project at http://www.vim.org feature all kinds of news, mailing lists, and other documentation.
- A number of vim sources are available on the Internet: http://www.selflinux.org/selflinux/html/vim.html, and http://www.apmaths.uwo.ca/~xli/vim/vim_tutorial.html. See http://linux-universe.com/HOWTO/Vim-HOWTO/vim-tutorial.html for further links to tutorials.

IMPORTANT: The VIM License

vim is “charityware,” which means that the authors do not charge any money for the software but encourage you to support a nonprofit project with a monetary contribution. This project solicits help for poor children in Uganda. More information is available online at http://iccf-holland.org/index.html, http://www.vim.org/iccf/, and http://www.iccf.nl/.

Part IV. Services

20 Basic Networking

Linux offers the necessary networking tools and features for integration into all types of network structures. The customary Linux protocol, TCP/IP, has various services and special features, which are discussed here. Network access using a network card, modem, or other device can be configured with YaST. Manual configuration is also possible. Only the fundamental mechanisms and the relevant network configuration files are discussed in this chapter.

Linux and other Unix operating systems use the TCP/IP protocol. It is not a single network protocol, but a family of network protocols that offer various services. The protocols listed in Table 20.1, "Several Protocols in the TCP/IP Protocol Family" (page 300) are provided for the purpose of exchanging data between two machines via TCP/IP. Networks combined by TCP/IP, comprising a worldwide network are also referred to, in their entirety, as "the Internet."

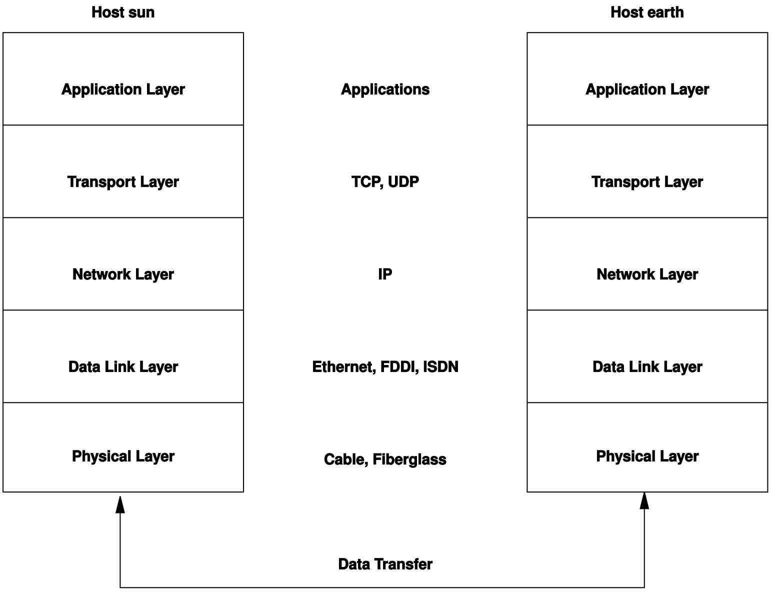
RFC stands for *Request for Comments*. RFCs are documents that describe various Internet protocols and implementation procedures for the operating system and its applications. The RFC documents describe the setup of Internet protocols. To expand your knowledge about any of the protocols, refer to the appropriate RFC documents. They are available online at `http://www.ietf.org/rfc.html`.

Table 20.1 Several Protocols in the TCP/IP Protocol Family

Protocol	Description
TCP	Transmission Control Protocol: A connection-oriented secure protocol. The data to transmit is first sent by the application as a stream of data then converted by the operating system to the appropriate format. The data arrives at the respective application on the destination host in the original data stream format in which it was initially sent. TCP determines whether any data has been lost during the transmission and that there is no mix-up. TCP is implemented wherever the data sequence matters.
UDP	User Datagram Protocol: A connectionless, insecure protocol. The data to transmit is sent in the form of packets generated by the application. The order in which the data arrives at the recipient is not guaranteed and data loss is a possibility. UDP is suitable for record-oriented applications. It features a smaller latency period than TCP.
ICMP	Internet Control Message Protocol: Essentially, this is not a protocol for the end user, but a special control protocol that issues error reports and can control the behavior of machines participating in TCP/IP data transfer. In addition, it provides a special echo mode that can be viewed using the program ping.
IGMP	Internet Group Management Protocol: This protocol controls machine behavior when implementing IP multicast.

As shown in Figure 20.1, “Simplified Layer Model for TCP/IP” (page 301), data exchange takes place in different layers. The actual network layer is the insecure data transfer via IP (Internet protocol). On top of IP, TCP (transmission control protocol) guarantees, to a certain extent, security of the data transfer. The IP layer is supported by the underlying hardware-dependent protocol, such as ethernet.

Figure 20.1 *Simplified Layer Model for TCP/IP*

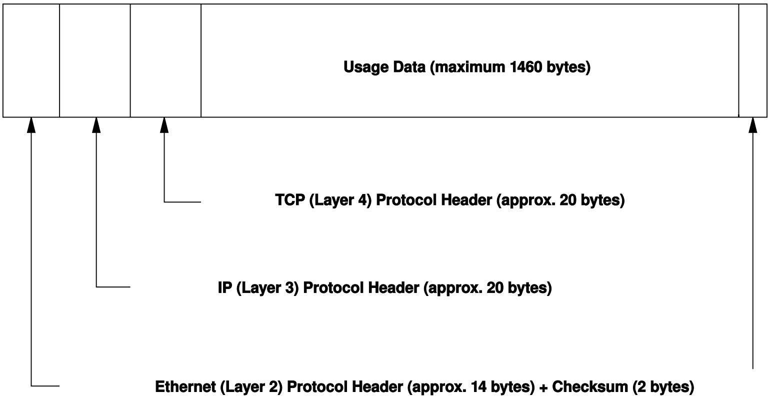


The diagram provides one or two examples for each layer. The layers are ordered according to *abstraction levels*. The lowest layer is very close to the hardware. The uppermost layer, however, is almost a complete abstraction from the hardware. Every layer has its own special function. The special functions of each layer are mostly implicit in their description. The data link and physical layers represent the physical network used, such as ethernet.

Almost all hardware protocols work on a packet-oriented basis. The data to transmit is packaged in *packets*, because it cannot be sent all at once. The maximum size of a TCP/IP packet is approximately 64 KB. Packets are normally quite a bit smaller, because the network hardware can be a limiting factor. The maximum size of a data packet on an ethernet is about fifteen hundred bytes. The size of a TCP/IP packet is limited to this amount when the data is sent over an ethernet. If more data is transferred, more data packets need to be sent by the operating system.

For the layers to serve their designated functions, additional information regarding each layer must be saved in the data packet. This takes place in the *header* of the packet. Every layer attaches a small block of data, called the protocol header, to the front of each emerging packet. A sample TCP/IP data packet traveling over an ethernet cable is illustrated in Figure 20.2, “TCP/IP Ethernet Packet” (page 302). The proof sum is located at the end of the packet, not at the beginning. This simplifies things for the network hardware.

Figure 20.2 *TCP/IP Ethernet Packet*


Usage Data (maximum 1460 bytes)
TCP (Layer 4) Protocol Header (approx. 20 bytes)
IP (Layer 3) Protocol Header (approx. 20 bytes)
Ethernet (Layer 2) Protocol Header (approx. 14 bytes) + Checksum (2 bytes)

When an application sends data over the network, the data passes through each layer, all implemented in the Linux kernel except the physical layer. Each layer is responsible for preparing the data so it can be passed to the next layer. The lowest layer is ultimately responsible for sending the data. The entire procedure is reversed when data is received. Like the layers of an onion, in each layer the protocol headers are removed from the transported data. Finally, the transport layer is responsible for making the data available for use by the applications at the destination. In this manner, one layer only communicates with the layer directly above or below it. For applications, it is irrelevant whether data is transmitted via a 100 Mbit/s FDDI network or via a 56-Kbit/s modem line. Likewise, it is irrelevant for the data line which kind of data is transmitted, as long as packets are in the correct format.

20.1 IP Addresses and Routing

The discussion in this section is limited to IPv4 networks. For information about IPv6 protocol, the successor to IPv4, refer to Section 20.2, "IPv6—The Next Generation Internet" (page 305).

20.1.1 IP Addresses

Every computer on the Internet has a unique 32-bit address. These 32 bits (or 4 bytes) are normally written as illustrated in the second row in Example 20.1, “Writing IP Addresses” (page 303).

Example 20.1 *Writing IP Addresses*

```
IP Address (binary):  11000000 10101000 00000000 00010100
IP Address (decimal):      192.     168.       0.      20
```

In decimal form, the four bytes are written in the decimal number system, separated by periods. The IP address is assigned to a host or a network interface. It cannot be used anywhere else in the world. There are exceptions to this rule, but these are not relevant in the following passages.

The points in IP addresses indicate the hierarchical system. Until the 1990s, IP addresses were strictly categorized in classes. However, this system has proven too inflexible and was discontinued. Now, *classless routing* (CIDR, classless interdomain routing) is used.

20.1.2 Netmasks and Routing

Netmasks are used to define the address range of a subnetwork. If two hosts are in the same subnetwork, they can reach each other directly, if they are not in the same subnetwork, they need the address of a gateway that handles all the traffic between the subnetwork and the rest of the world. To check if two IP addresses are in the same subnet, simply “AND” both addresses with the netmask. If the result is identical, both IP addresses are in the same local network. If there are differences, the remote IP address, and thus the remote interface, can only be reached over a gateway.

To understand how the netmask works, look at Example 20.2, “Linking IP Addresses to the Netmask” (page 304). The netmask consists of 32 bits that identify how much of an IP address belongs to the network. All those bits that are `1` mark the corresponding bit in the IP address as belonging to the network. All bits that are `0` mark bits inside the subnetwork. This means that the more bits are `1`, the smaller the subnetwork is. Because the netmask always consists of several successive `1` bits, it is also possible to just count the number of bits in the netmask. In Example 20.2, “Linking IP Addresses to the Netmask” (page 304) the first net with 24 bits could also be written as `192.168.0.0/24`.

Example 20.2 *Linking IP Addresses to the Netmask*

```
IP address (192.168.0.20):  11000000 10101000 00000000 00010100
Netmask   (255.255.255.0):  11111111 11111111 11111111 00000000
---------------------------------------------------------------
Result of the link:         11000000 10101000 00000000 00000000
In the decimal system:           192.     168.       0.       0

IP address (213.95.15.200): 11010101 10111111 00001111 11001000
Netmask    (255.255.255.0): 11111111 11111111 11111111 00000000
---------------------------------------------------------------
Result of the link:         11010101 10111111 00001111 00000000
In the decimal system:           213.      95.      15.       0
```

To give another example: all machines connected with the same ethernet cable are usually located in the same subnetwork and are directly accessible. Even when the subnet is physically divided by switches or bridges, these hosts can still be reached directly.

IP addresses outside the local subnet can only be reached if a gateway is configured for the target network. In the most common case, there is only one gateway that handles all traffic that is external. However, it is also possible to configure several gateways for different subnets.

If a gateway has been configured, all external IP packets are sent to the appropriate gateway. This gateway then attempts to forward the packets in the same manner—from host to host—until it reaches the destination host or the packet's TTL (time to live) expires.

Table 20.2 *Specific Addresses*

Address Type	Description
Base Network Address	This is the netmask AND any address in the network, as shown in Example 20.2, “Linking IP Addresses to the Netmask” (page 304) under `Result`. This address cannot be assigned to any hosts.
Broadcast Address	This basically says, “Access all hosts in this subnetwork.” To generate this, the netmask is inverted in binary form and linked to the base network address with a logical OR. The above example therefore results in 192.168.0.255. This address cannot be assigned to any hosts.

Address Type	Description
Local Host	The address `127.0.0.1` is assigned to the “loopback device” on each host. A connection can be set up to your own machine with this address.

Because IP addresses must be unique all over the world, you cannot just select random addresses. There are three address domains to use if you want to set up a private IP-based network. These cannot get any connection from the rest of the Internet, because they cannot be transmitted over the Internet. These address domains are specified in RFC 1597 and listed in Table 20.3, “Private IP Address Domains” (page 305).

Table 20.3 Private IP Address Domains

Network/Netmask	Domain
`10.0.0.0/255.0.0.0`	`10.x.x.x`
`172.16.0.0/255.240.0.0`	`172.16.x.x – 172.31.x.x`
`192.168.0.0/255.255.0.0`	`192.168.x.x`

20.2 IPv6—The Next Generation Internet

Due to the emergence of the WWW (World Wide Web), the Internet has experienced explosive growth with an increasing number of computers communicating via TCP/IP in the past fifteen years. Since Tim Berners-Lee at CERN (`http://public.web.cern.ch`) invented the WWW in 1990, the number of Internet hosts has grown from a few thousand to about a hundred million.

As mentioned, an IPv4 address consists of only 32 bits. Also, quite a few IP addresses are lost—they cannot be used due to the way in which networks are organized. The number of addresses available in your subnet is two to the power of the number of bits, minus two. A subnetwork has, for example, 2, 6, or 14 addresses available. To connect 128 hosts to the Internet, for example, you need a subnetwork with 256 IP addresses, from which only 254 are usable, because two IP addresses are needed for the structure of the subnetwork itself: the broadcast and the base network address.

Under the current IPv4 protocol, DHCP or NAT (network address translation) are the typical mechanisms used to circumvent the potential address shortage. Combined with the convention to keep private and public address spaces separate, these methods can certainly mitigate the shortage. The problem with them lies in their configuration, which is a chore to set up and a burden to maintain. To set up a host in an IPv4 network, you need a number of address items, such as the host's own IP address, the subnetmask, the gateway address, and maybe a name server address. All these items need to be known and cannot be derived from somewhere else.

With IPv6, both the address shortage and the complicated configuration should be a thing of the past. The following sections tell more about the improvements and benefits brought by IPv6 and about the transition from the old protocol to the new one.

20.2.1 Advantages

The most important and most visible improvement brought by the new protocol is the enormous expansion of the available address space. An IPv6 address is made up of 128 bit values instead of the traditional 32 bits. This provides for as many as several quadrillion IP addresses.

However, IPv6 addresses are not only different from their predecessors with regard to their length. They also have a different internal structure that may contain more specific information about the systems and the networks to which they belong. More details about this are found in Section 20.2.2, “Address Types and Structure” (page 308).

The following is a list of some other advantages of the new protocol:

Autoconfiguration

IPv6 makes the network “plug and play” capable, which means that a newly set up system integrates into the (local) network without any manual configuration. The new host uses its automatic configuration mechanism to derive its own address from the information made available by the neighboring routers, relying on a protocol called the *neighbor discovery* (ND) protocol. This method does not require any intervention on the administrator's part and there is no need to maintain a central server for address allocation—an additional advantage over IPv4, where automatic address allocation requires a DHCP server.

Mobility

IPv6 makes it possible to assign several addresses to one network interface at the same time. This allows users to access several networks easily, something that could be compared with the international roaming services offered by mobile phone companies: when you take your mobile phone abroad, the phone automatically logs in to a foreign service as soon as it enters the corresponding area, so you can be reached under the same number everywhere and are able to place an outgoing call just like in your home area.

Secure Communication

With IPv4, network security is an add-on function. IPv6 includes IPsec as one of its core features, allowing systems to communicate over a secure tunnel to avoid eavesdropping by outsiders on the Internet.

Backward Compatibility

Realistically, it would be impossible to switch the entire Internet from IPv4 to IPv6 at one time. Therefore, it is crucial that both protocols are able to coexist not only on the Internet, but also on one system. This is ensured by compatible addresses (IPv4 addresses can easily be translated into IPv6 addresses) and through the use of a number of tunnels. See Section 20.2.3, "Coexistence of IPv4 and IPv6" (page 312). Also, systems can rely on a *dual stack IP* technique to support both protocols at the same time, meaning that they have two network stacks that are completely separate, such that there is no interference between the two protocol versions.

Custom Tailored Services through Multicasting

With IPv4, some services, such as SMB, need to broadcast their packets to all hosts in the local network. IPv6 allows a much more fine-grained approach by enabling servers to address hosts through *multicasting*—by addressing a number of hosts as parts of a group (which is different from addressing all hosts through *broadcasting* or each host individually through *unicasting*). Which hosts are addressed as a group may depend on the concrete application. There are some predefined groups to address all name servers (the *all name servers multicast group*), for example, or all routers (the *all routers multicast group*).

20.2.2 Address Types and Structure

As mentioned, the current IP protocol is lacking in two important aspects: there is an increasing shortage of IP addresses and configuring the network and maintaining the routing tables is becoming a more complex and burdensome task. IPv6 solves the first problem by expanding the address space to 128 bits. The second one is countered by introducing a hierarchical address structure, combined with sophisticated techniques to allocate network addresses, as well as *multihoming* (the ability to assign several addresses to one device, giving access to several networks).

When dealing with IPv6, it is useful to know about three different types of addresses:

Unicast
: Addresses of this type are associated with exactly one network interface. Packets with such an address are delivered to only one destination. Accordingly, unicast addresses are used to transfer packets to individual hosts on the local network or the Internet.

Multicast
: Addresses of this type relate to a group of network interfaces. Packets with such an address are delivered to all destinations that belong to the group. Multicast addresses are mainly used by certain network services to communicate with certain groups of hosts in a well-directed manner.

Anycast
: Addresses of this type are related to a group of interfaces. Packets with such an address are delivered to the member of the group that is closest to the sender, according to the principles of the underlying routing protocol. Anycast addresses are used to make it easier for hosts to find out about servers offering certain services in the given network area. All servers of the same type have the same anycast address. Whenever a host requests a service, it receives a reply from the server with the closest location, as determined by the routing protocol. If this server should fail for some reason, the protocol automatically selects the second closest server, then the third one, and so forth.

An IPv6 address is made up of eight four-digit fields, each representing 16 bits, written in hexadecimal notation. They are also separated by colons (`:`). Any leading zero bytes within a given field may be dropped, but zeros within the field or at its end may not. Another convention is that more than four consecutive zero bytes may be collapsed into a double colon. However, only one such `::` is allowed per address. This kind of shorthand notation is shown in Example 20.3, “Sample IPv6 Address” (page 309), where all three lines represent the same address.

Example 20.3 *Sample IPv6 Address*

```
fe80 : 0000 : 0000 : 0000 : 0000 : 10 : 1000 : 1a4
fe80 :    0 :    0 :    0 :    0 : 10 : 1000 : 1a4
fe80 :                           : 10 : 1000 : 1a4
```

Each part of an IPv6 address has a defined function. The first bytes form the prefix and specify the type of address. The center part is the network portion of the address, but it may be unused. The end of the address forms the host part. With IPv6, the netmask is defined by indicating the length of the prefix after a slash at the end of the address. An address, as shown in Example 20.4, “IPv6 Address Specifying the Prefix Length” (page 309), contains the information that the first 64 bits form the network part of the address and the last 64 form its host part. In other words, the `64` means that the netmask is filled with 64 1-bit values from the left. Just like with IPv4, the IP address is combined with AND with the values from the netmask to determine whether the host is located in the same subnetwork or in another one.

Example 20.4 *IPv6 Address Specifying the Prefix Length*

```
fe80::10:1000:1a4/64
```

IPv6 knows about several predefined types of prefixes. Some of these are shown in Table 20.4, “Various IPv6 Prefixes” (page 309).

Table 20.4 *Various IPv6 Prefixes*

Prefix (hex)	Definition
`00`	IPv4 addresses and IPv4 over IPv6 compatibility addresses. These are used to maintain compatibility with IPv4. Their use still requires a router able to translate IPv6 packets into IPv4 packets. Several special addresses, such as the one for the loopback device, have this prefix as well.
`2` or `3` as the first digit	Aggregatable global unicast addresses. As is the case with IPv4, an interface can be assigned to form part of a certain subnetwork. Currently, there are the following address spaces: `2001::/16` (production quality address space) and `2002::/16` (6to4 address space).

Prefix (hex)	Definition
`fe80::/10`	Link-local addresses. Addresses with this prefix should not be routed and should therefore only be reachable from within the same subnetwork.
`fec0::/10`	Site-local addresses. These may be routed, but only within the network of the organization to which they belong. In effect, they are the IPv6 equivalent of the current private network address space, such as `10.x.x.x`.
`ff`	These are multicast addresses.

A unicast address consists of three basic components:

Public Topology
: The first part (which also contains one of the prefixes mentioned above) is used to route packets through the public Internet. It includes information about the company or institution that provides the Internet access.

Site Topology
: The second part contains routing information about the subnetwork to which to deliver the packet.

Interface ID
: The third part identifies the interface to which to deliver the packet. This also allows for the MAC to form part of the address. Given that the MAC is a globally unique, fixed identifier coded into the device by the hardware maker, the configuration procedure is substantially simplified. In fact, the first 64 address bits are consolidated to form the `EUI-64` token, with the last 48 bits taken from the MAC, and the remaining 24 bits containing special information about the token type. This also makes it possible to assign an `EUI-64` token to interfaces that do not have a MAC, such as those based on PPP or ISDN.

On top of this basic structure, IPv6 distinguishes between five different types of unicast addresses:

`::` (unspecified)
: This address is used by the host as its source address when the interface is initialized for the first time—when the address cannot yet be determined by other means.

`::1` (loopback)
: The address of the loopback device.

IPv4 Compatible Addresses
: The IPv6 address is formed by the IPv4 address and a prefix consisting of 96 zero bits. This type of compatibility address is used for tunneling (see Section 20.2.3, "Coexistence of IPv4 and IPv6" (page 312)) to allow IPv4 and IPv6 hosts to communicate with others operating in a pure IPv4 environment.

IPv4 Addresses Mapped to IPv6
: This type of address specifies a pure IPv4 address in IPv6 notation.

Local Addresses
: There are two address types for local use:

    link-local
    : This type of address can only be used in the local subnetwork. Packets with a source or target address of this type should not be routed to the Internet or other subnetworks. These addresses contain a special prefix (`fe80::/10`) and the interface ID of the network card, with the middle part consisting of zero bytes. Addresses of this type are used during automatic configuration to communicate with other hosts belonging to the same subnetwork.

    site-local
    : Packets with this type of address may be routed to other subnetworks, but not to the wider Internet—they must remain inside the organization's own network. Such addresses are used for intranets and are an equivalent of the private address space defined by IPv4. They contain a special prefix (`fec0::/10`), the interface ID, and a 16 bit field specifying the subnetwork ID. Again, the rest is filled with zero bytes.

As a completely new feature introduced with IPv6, each network interface normally gets several IP addresses, with the advantage that several networks can be accessed through the same interface. One of these networks can be configured completely automatically using the MAC and a known prefix with the result that all hosts on the local network can be reached as soon as IPv6 is enabled (using the link-local address). With the MAC forming part of it, any IP address used in the world is unique. The only variable parts of the address are those specifying the *site topology* and the *public topology*, depending on the actual network in which the host is currently operating.

For a host to go back and forth between different networks, it needs at least two addresses. One of them, the *home address*, not only contains the interface ID but also an identifier of the home network to which it normally belongs (and the corresponding prefix). The home address is a static address and, as such, it does not normally change. Still, all packets destined to the mobile host can be delivered to it, regardless of whether it operates in the home network or somewhere outside. This is made possible by the completely new features introduced with IPv6, such as *stateless autoconfiguration* and *neighbor discovery*. In addition to its home address, a mobile host gets one or more additional addresses that belong to the foreign networks where it is roaming. These are called *care-of* addresses. The home network has a facility that forwards any packets destined to the host when it is roaming outside. In an IPv6 environment, this task is performed by the *home agent*, which takes all packets destined to the home address and relays them through a tunnel. On the other hand, those packets destined to the care-of address are directly transferred to the mobile host without any special detours.

20.2.3 Coexistence of IPv4 and IPv6

The migration of all hosts connected to the Internet from IPv4 to IPv6 is a gradual process. Both protocols will coexist for some time to come. The coexistence on one system is guaranteed where there is a *dual stack* implementation of both protocols. That still leaves the question of how an IPv6 enabled host should communicate with an IPv4 host and how IPv6 packets should be transported by the current networks, which are predominantly IPv4 based. The best solutions offer tunneling and compatibility addresses (see Section 20.2.2, "Address Types and Structure" (page 308)).

IPv6 hosts that are more or less isolated in the (worldwide) IPv4 network can communicate through tunnels: IPv6 packets are encapsulated as IPv4 packets to move them across an IPv4 network. Such a connection between two IPv4 hosts is called a *tunnel*. To achieve this, packets must include the IPv6 destination address (or the corresponding prefix) as well as the IPv4 address of the remote host at the receiving end of the tunnel. A basic tunnel can be configured manually according to an agreement between the hosts' administrators. This is also called *static tunneling*.

However, the configuration and maintenance of static tunnels is often too labor-intensive to use them for daily communication needs. Therefore, IPv6 provides for three different methods of *dynamic tunneling*:

6over4
: IPv6 packets are automatically encapsulated as IPv4 packets and sent over an IPv4 network capable of multicasting. IPv6 is tricked into seeing the whole network (Internet) as a huge local area network (LAN). This makes it possible to determine the receiving end of the IPv4 tunnel automatically. However, this method does not scale very well and is also hampered by the fact that IP multicasting is far from widespread on the Internet. Therefore, it only provides a solution for smaller corporate or institutional networks where multicasting can be enabled. The specifications for this method are laid down in RFC 2529.

6to4
: With this method, IPv4 addresses are automatically generated from IPv6 addresses, enabling isolated IPv6 hosts to communicate over an IPv4 network. However, a number of problems have been reported regarding the communication between those isolated IPv6 hosts and the Internet. The method is described in RFC 3056.

IPv6 Tunnel Broker
: This method relies on special servers that provide dedicated tunnels for IPv6 hosts. It is described in RFC 3053.

20.2.4 Configuring IPv6

To configure IPv6, you normally do not need to make any changes on the individual workstations. IPv6 is enabled by default. You can disable it during the installation in the network configuration step described in Section “Network” (Chapter 1, *Installation with YaST*, ↑Start-Up). To disable or enable IPv6 on an installed system, start YaST, open the *Network Devices* > *Network Card* dialog and in the *Global Options* tab, check or uncheck *Enable IPv6*. To enable IPv6 manually, enter `modprobe ipv6` as `root`.

Because of the autoconfiguration concept of IPv6, the network card is assigned an address in the *link-local* network. Normally, no routing table management takes place on a workstation. The network routers can be queried by the workstation, using the *router advertisement protocol*, for what prefix and gateways should be implemented. The radvd program can be used to set up an IPv6 router. This program informs the workstations which prefix to use for the IPv6 addresses and which routers. Alternatively, use zebra for automatic configuration of both addresses and routing.

Consult the ifup(8) man page to get information about how to set up various types of tunnels using the `/etc/sysconfig/network` files.

20.2.5 For More Information

The above overview does not cover the topic of IPv6 comprehensively. For a more in-depth look at the new protocol, refer to the following online documentation and books:

`http://www.ipv6.org/`
The starting point for everything about IPv6.

`http://www.ipv6day.org`
All information needed to start your own IPv6 network.

`http://www.ipv6-to-standard.org/`
The list of IPv6-enabled products.

`http://www.bieringer.de/linux/IPv6/`
Here, find the Linux IPv6-HOWTO and many links related to the topic.

RFC 2640
The fundamental RFC about IPv6.

IPv6 Essentials
A book describing all the important aspects of the topic is *IPv6 Essentials* by Silvia Hagen (ISBN 0-596-00125-8).

20.3 Name Resolution

DNS assists in assigning an IP address to one or more names and assigning a name to an IP address. In Linux, this conversion is usually carried out by a special type of software known as bind. The machine that takes care of this conversion is called a *name server*. The names make up a hierarchical system in which each name component is separated by dots. The name hierarchy is, however, independent of the IP address hierarchy described above.

Consider a complete name, such as `earth.example.com`, written in the format `hostname.domain`. A full name, referred to as a *fully qualified domain name* (FQDN), consists of a hostname and a domain name (`example.com`). The latter also includes the *top level domain* or TLD (`com`).

TLD assignment has become quite confusing for historical reasons. Traditionally, three-letter domain names are used in the USA. In the rest of the world, the two-letter ISO national codes are the standard. In addition to that, longer TLDs were introduced in 2000 that represent certain spheres of activity (for example, `.info`, `.name`, `.museum`).

In the early days of the Internet (before 1990), the file `/etc/hosts` was used to store the names of all the machines represented over the Internet. This quickly proved to be impractical in the face of the rapidly growing number of computers connected to the Internet. For this reason, a decentralized database was developed to store the hostnames in a widely distributed manner. This database, similar to the name server, does not have the data pertaining to all hosts in the Internet readily available, but can dispatch requests to other name servers.

The top of the hierarchy is occupied by *root name servers*. These root name servers manage the top level domains and are run by the Network Information Center (NIC). Each root name server knows about the name servers responsible for a given top level domain. Information about top level domain NICs is available at `http://www.internic.net`.

DNS can do more than just resolve hostnames. The name server also knows which host is receiving e-mails for an entire domain—the *mail exchanger (MX)*.

For your machine to resolve an IP address, it must know about at least one name server and its IP address. Easily specify such a name server with the help of YaST. If you have a modem dial-up connection, you may not need to configure a name server manually at all. The dial-up protocol provides the name server address as the connection is made. The configuration of name server access with openSUSE® is described in Chapter 22, *The Domain Name System* (page 353).

The protocol `whois` is closely related to DNS. With this program, quickly find out who is responsible for any given domain.

20.4 Configuring a Network Connection with YaST

There are many supported networking types on Linux. Most of them use different device names and the configuration files are spread over several locations in the file system.

For a detailed overview of the aspects of manual network configuration, see Section 20.5, “Configuring a Network Connection Manually” (page 332).

During installation, YaST can be used to configure automatically all interfaces that have been detected. Additional hardware can be configured any time after installation in the installed system. The following sections describe the network configuration for all types of network connections supported by openSUSE.

20.4.1 Configuring the Network Card with YaST

To configure your wired or wireless network card in YaST, select *Network Devices* > *Network Card*. After starting the module, YaST displays the Network Settings dialog with four tabs: *Global Options*, *Overview*, *Hostname/DNS*, and *Routing*.

The *Overview* tab contains information about installed network cards. Any properly detected network card is listed with its name. You can manually add new cards, remove them or change their configuration in this dialog. If you want to manually add and configure a card that was not automatically detected, read Section “Configuring an Undetected Network Card” (page 323).

The *Global Options* tab allows to set important global networking options. In the *Network Setup Method* choose the way network connections are managed. If you want a NetworkManager desktop applet to manage connections for all interfaces, choose *User Controlled with NetworkManager*. This option is well suited for switching between multiple wired and wireless networks. If you do not run a desktop environment (GNOME or KDE) or if you need to use multiple interfaces at once, use the *Traditional Method with ifup*. For more information on NetworkManager, see Chapter 10, *Managing Network Connections with NetworkManager* (↑Start-Up).

In the *IPv6 Protocol Settings* choose whether you want to use IPv6 protocol. It is possible to use IPv6 together with IPv4. By default, IPv6 is activated. However, in networks not using IPv6 protocol, response times can be faster with IPv6 protocol disabled. If you want to disable IPv6, uncheck the *Enable IPv6* option. This disables autoload of the kernel module for IPv6. The change will be applied after reboot.

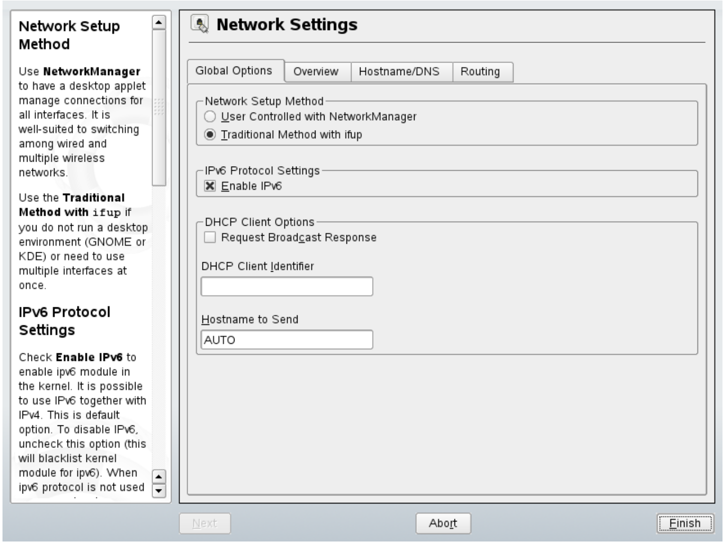
In the *DHCP Client Options* configure options for the DHCP client. If you want the DHCP client to ask the server to always broadcast its responses, check *Request*

Broadcast Response. It may be needed if your machine is moving between different networks.

DHCP Client Identifier must be different for each DHCP client on a single network. If left empty, it defaults to the hardware address of the network interface. However, if you are running several virtual machines using the same network interface and therefore the same hardware address, specify a unique free-form identifier here.

The Hostname to Send specifies a string used for the hostname option field when dhcpcd sends messages to DHCP server. Some DHCP servers update name server zones (forward and reverse records) according to this hostname (dynamic DNS). Also, some DHCP servers require the Hostname to Send option field to contain a specific string in the DHCP messages from clients. Leave AUTO to send the current hostname (i.e. the one defined in `/etc/HOSTNAME)`. Leave the option field empty for not sending a hostname.

Figure 20.3 *Configuring Network Settings*



Network Setup Method

Use **NetworkManager** to have a desktop applet manage connections for all interfaces. It is well-suited to switching among wired and multiple wireless networks.

Use the **Traditional Method with `ifup`** if you do not run a desktop environment (GNOME or KDE) or need to use multiple interfaces at once.

IPv6 Protocol Settings

Check **Enable IPv6** to enable ipv6 module in the kernel. It is possible to use IPv6 together with IPv4. This is default option. To disable IPv6, uncheck this option (this will blacklist kernel module for ipv6). When ipv6 protocol is not used

Network Settings

Network Setup Method
○ User Controlled with NetworkManager
● Traditional Method with ifup

IPv6 Protocol Settings
☒ Enable IPv6

DHCP Client Options
☐ Request Broadcast Response

DHCP Client Identifier:

Hostname to Send: AUTO

The *Hostname/DNS* tab allows to set the hostname of the computer and name servers to be used. For more information about these options see Section “Configuring Hostname and DNS” (page 319). In the *Routing* tab, you can set the default gateway and routing details. See Section “Configuring Routing” (page 320) for more information.

Changing the Configuration of a Network Card

To change the configuration of a network card, select a card from the list of the detected cards in the *Overview* tab of the YaST network card configuration module and click *Configure*. The *Network Card* dialog appears in which to adjust the card configuration using the *General*, *Address*, and *Hardware* tabs. For information about wireless card configuration, see Section 33.1.3, “Configuration with YaST” (page 548).

Configuring IP Addresses

If possible, wired network cards that are available during the installation are automatically configured to use automatic address setup, DHCP.

DHCP should also be used if you are using a DSL line but with no static IP assigned by the ISP (Internet Service Provider). If you decide to use DHCP, configure the details in *DHCP Client Options* on *Global Options* tab of the *Network Settings* dialog of the YaST network card configuration module. Specify whether the DHCP client should ask the server to always broadcast its responses in *Request Broadcast response*. This option may be needed if your machine is a mobile client moving between networks. If you have a virtual host setup where different hosts communicate through the same interface, an *DHCP Client Identifier* is necessary to distinguish them.

DHCP is a good choice for client configuration but it is not ideal for server configuration. To set a static IP address, proceed as follows:

1. Select a card from the list of detected cards in the *Overview* tab of the YaST network card configuration module and click *Configure*.

2. In the *Address* tab, choose *Statically assigned IP adress*.

3. Enter *IP Address* and *Subnet Mask*.

4. Click *Next*.

5. To activate the configuration, click *Finish*.

If you use the static address, the name servers and default gateway are not configured automatically. To configure name servers, proceed as described in Section “Configuring Hostname and DNS” (page 319). To configure a gateway, proceed as described in Section “Configuring Routing” (page 320).

Configuring Aliases

One network device can have multiple IP addresses, called aliases. To set an alias for your network card, proceed as follows:

1. Select a card from the list of detected cards in the *Overview* tab of the YaST network card configuration module and click *Configure*.

2. In the *Additional Addresses* part of the *Address* tab, click *Add*.

3. Enter *Alias Name*, *IP Address*, and *Netmask*. Do not include the interface name in the alias name.

4. Click *OK*.

5. Click *Next*.

6. To activate the configuration, click *Finish*.

Configuring Hostname and DNS

If you did not change the network configuration during installation and the wired card was available, a hostname was automatically generated for your computer and DHCP was activated. The same applies to the name service information your host needs to integrate into a network environment. If DHCP is used for network address setup, the list of domain name servers is automatically filled with the appropriate data. If a static setup is preferred, set these values manually.

To change the name of your computer and adjust the name server search list, proceed as follows:

1. Go to the *Hostname/DNS* tab of the YaST network card configuration.

2. Enter *Hostname* and, if needed, the *Domain Name*. Note that the hostname is global and applies to all set network interfaces.

   If you are using DHCP to get an IP adress, the hostname of your computer will be automatically set by the DHCP. You may want to disable this behavior by unchecking *Change Hostname via DHCP* if you connect to different networks which may assign different hostnames, because changing the hostname at runtime may confuse the graphical desktop.

If you are using DHCP to get an IP address, your hostname will be written to `/etc/hosts` by default and be resolvable as a `127.0.0.2` IP address. If you want to disable this, uncheck *Write Hostname to /etc/hosts* but note, that your hostname will not be resolvable without an active network.

3. Enter the name servers and domain search list.

4. To activate the configuration, click *Finish*.

Configuring Routing

To make your machine communicate with other machines and other networks, routing information must be given to make network traffic take the correct path. If DHCP is used, this information is automatically provided. If a static setup is used, this data must be added manually.

1. Go to the *Routing* tab of the YaST network card configuration module.

2. Enter the IP of the *Default Gateway*. The default gateway matches every possible destination, but poorly. If any other entry exists that matches the required address, it is used instead of the default route.

3. If you need to add more entries into the *Routing Table*, check *Expert Configuration*. Then add an entry with *Add*. Enter *Destination*, *Netmask* and optionally select the *Device* to be used.

4. If the system is a router, enable the *IP Forwarding* option.

5. To activate the configuration, click *Finish*.

Changing Hardware Options

It is possible to change the device name of the network card when it is used if you need to. It is also possible to switch to a different driver. To set these options with YaST, proceed as follows:

1. Select a card from the list of detected cards in the YaST network card configuration module and click *Configure*.

2. Go to the *Hardware* tab. The *Matching rule* under *Udev rules* is not editable. It is the hardware address (MAC) or bus ID udev uses to identify the network card. You can, however, change the device name of this card by editing it in *Device Name*.

3. In *Driver name* you can set the driver to be used for the network card.

4. Click *Next*.

5. To activate configuration, click *Finish*.

Starting the Device

If you use the traditional method with ifup, you can configure your device to start during boot, on cable connection, on card detection, manually, or never. To change device start-up, proceed as follows:

1. Select a card from the list of detected cards in the YaST network card configuration module and click *Configure*.

2. In the *General* tab, select the desired entry from *Device Activation*.

   Choose *At Boot Time* to start the device during the system boot. With *On Cable Connection*, the interface is watched for any existing physical connection. With *On Hotplug*, the interface is set as soon as available. Choose *Manually* to control the interface manually with `ifup`. Choose *Never* to not start the device at all. The *On NFSroot* is similar to *At Boot Time*, but the interface is never shut down with the command `rcnetwork stop`. Use this if you use a nfs or iscsi root filesystem.

3. Click *Next*.

4. To activate the configuration, click *Finish*.

Configuring the Firewall

Without having to enter the detailed firewall setup as described in Section 35.4.1, "Configuring the Firewall with YaST" (page 583), you can determine the basic firewall setup for your device as part of the device setup. Proceed as follows:

1. Select a card from the list of detected cards in the YaST network card configuration module and click *Configure*.

2. Enter the *General* tab of the network configuration dialog.

3. Determine the firewall zone to which your interface should be assigned. The following options are available:

   Firewall Disabled
   : The firewall does not run at all. Only use this option if your machine is part of a greater network that is protected by an outer firewall.

   Internal Zone (Unprotected)
   : The firewall is running, but does not enforce any rules to protect this interface. Only use this option, if your machine part is part of a greater network that is protected by an outer firewall.

   Demilitarized Zone
   : A demilitarized zone is an additional line of defense in front of an internal network and the (hostile) Internet. Hosts assigned to this zone can be reached from the internal network and from the Internet, but cannot access the internal network.

   External Zone
   : The firewall is running on this interface and fully protects it against other (presumably hostile) network traffic. This is the default option.

4. Click *Next*.

5. Activate the configuration by clicking *Finish*.

Configuring an Undetected Network Card

Your card may not be detected correctly. In this case, the card is not included in the list of the detected cards. If you are sure that your system includes a driver for your card, you can configure it manually. To configure an undetected network card, proceed as follows:

1. In the *Overview* tab of the YaST Network Card module click *Add*.

2. In the *Hardware* dialog, set the *Device Type* of the interface from the available options and *Configuration Name*. If the network card is a PCMCIA or USB device, activate the respective check box and exit this dialog with *Next*. Otherwise, select your network card model from *Select from List*. YaST then automatically selects the appropriate kernel module for the card. If your card is not in the list, select the *Module Name* manually.

3. Click *Next*.

4. In the *Address* tab, set the device type of the interface, the configuration name, and IP address. To use a static address, choose *Statically assigned IP Address* then complete *IP Address* and *Subnet Mask*. Here, you can also select to configure the hostname and aditional addresses of the interface (see Section "Configuring Aliases" (page 319)).

   If you selected *Wireless* as the device type of the interface, configure the wireless connection in the next dialog. Detailed information about wireless device configuration is available in Section 33.1, "Wireless LAN" (page 543).

5. In the *General* tab, set the *Firewall Zone* and *Device Activation*. With *Enable Device Control for Non-root User via KInternet*, grant connection control to ordinary users.

6. Click *Next*.

7. To activate the new network configuration, click *Finish*.

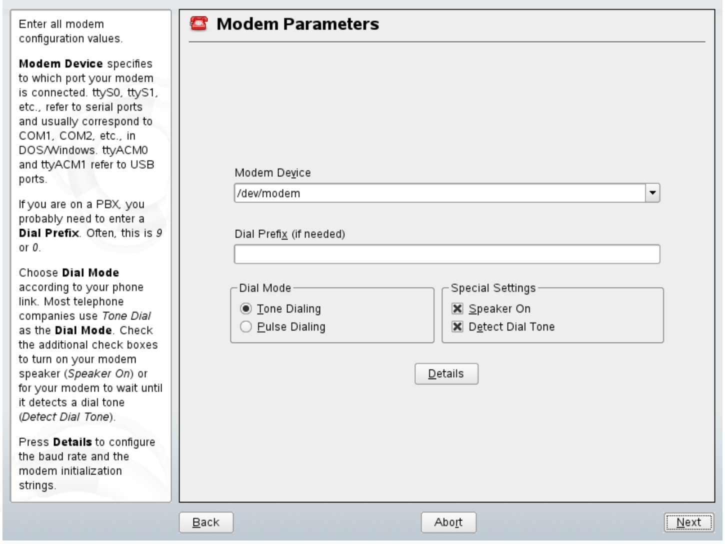
20.4.2 Modem

In the YaST Control Center, access the modem configuration under *Network Devices* > *Modem*. If your modem was not automatically detected, open the dialog for manual configuration by clicking *Add*. Enter the interface to which the modem is connected under *Modem Device*.

TIP: CDMA and GPRS Modems

Configure supported CDMA and GPRS modems with the YaST modem module just as you would configure regular modems.

Figure 20.4 *Modem Configuration*



Enter all modem configuration values.

Modem Device specifies to which port your modem is connected. ttyS0, ttyS1, etc., refer to serial ports and usually correspond to COM1, COM2, etc., in DOS/Windows. ttyACM0 and ttyACM1 refer to USB ports.

If you are on a PBX, you probably need to enter a **Dial Prefix**. Often, this is *9* or *0*.

Choose **Dial Mode** according to your phone link. Most telephone companies use *Tone Dial* as the **Dial Mode**. Check the additional check boxes to turn on your modem speaker (*Speaker On*) or for your modem to wait until it detects a dial tone (*Detect Dial Tone*).

Press **Details** to configure the baud rate and the modem initialization strings.

Modem Parameters

Modem Device: /dev/modem

Dial Prefix (if needed):

Dial Mode

● Tone Dialing
○ Pulse Dialing

Special Settings

☒ Speaker On
☒ Detect Dial Tone

If you are behind a private branch exchange (PBX), you may need to enter a dial prefix. This is often a zero. Consult the instructions that came with the PBX to find out. Also select whether to use tone or pulse dialing, whether the speaker should be on, and whether the modem should wait until it detects a dial tone. The last option should not be enabled if the modem is connected to an exchange.

Under *Details*, set the baud rate and the modem initialization strings. Only change these settings if your modem was not detected automatically or if it requires special settings for data transmission to work. This is mainly the case with ISDN terminal adapters. Leave this dialog by clicking *OK*. To delegate control over the modem to the normal user without root permissions, activate *Enable Device Control for Non-root User via Kinternet*. In this way, a user without administrator permissions can activate or deactivate an interface. Under *Dial Prefix Regular Expression*, specify a regular expression. The *Dial Prefix* in KInternet, which can be modified by the normal user, must match this regular expression. If this field is left empty, the user cannot set a different *Dial Prefix* without administrator permissions.

In the next dialog, select the ISP. To choose from a predefined list of ISPs operating in your country, select *Country*. Alternatively, click *New* to open a dialog in which to provide the data for your ISP. This includes a name for the dial-up connection and ISP as well as the login and password provided by your ISP. Enable *Always Ask for Password* to be prompted for the password each time you connect.

In the last dialog, specify additional connection options:

Dial on Demand
If you enable dial on demand, set at least one name server. Use this feature only if your Internet connection is inexpensive, because there are programs that periodically request data from the Internet.

Modify DNS when Connected
This option is enabled by default, with the effect that the name server address is updated each time you connect to the Internet.

Automatically Retrieve DNS
If the provider does not transmit its domain name server after connecting, disable this option and enter the DNS data manually.

Automatically Reconnect
If this options is enabled, the connection is automatically reestablished after failure.

Ignore prompts
This option disables the detection of any prompts from the dial-up server. If the connection build-up is slow or does not work at all, try this option.

External Firewall Interface
Selecting this option activates the SUSEfirewall2 and sets the interface as external. This way, you are protected from outside attacks for the duration of your Internet connection.

Idle Time-Out (seconds)
With this option, specify a period of network inactivity after which the modem disconnects automatically.

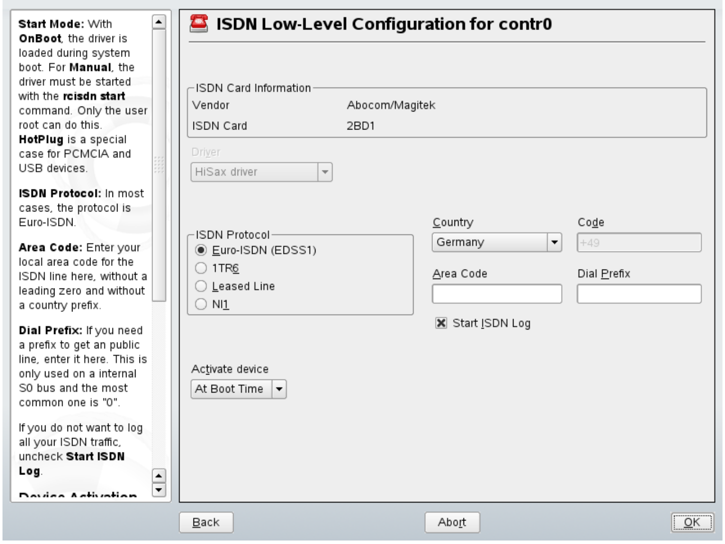
IP Details
This opens the address configuration dialog. If your ISP does not assign a dynamic IP address to your host, disable *Dynamic IP Address* then enter your host's local IP address and the remote IP address. Ask your ISP for this information. Leave *Default Route* enabled and close the dialog by selecting *OK*.

Selecting *Next* returns to the original dialog, which displays a summary of the modem configuration. Close this dialog with *Finish*.

20.4.3 ISDN

Use this module to configure one or several ISDN cards for your system. If YaST did not detect your ISDN card, click on *Add* in the *ISDN Devices* tab and manually select your card. Multiple interfaces are possible, but several ISPs can be configured for one interface. In the subsequent dialogs, set the ISDN options necessary for the proper functioning of the card.

Figure 20.5 *ISDN Configuration*



Start Mode: With **OnBoot**, the driver is loaded during system boot. For **Manual**, the driver must be started with the **rcisdn start** command. Only the user root can do this. **HotPlug** is a special case for PCMCIA and USB devices.

ISDN Protocol: In most cases, the protocol is Euro-ISDN.

Area Code: Enter your local area code for the ISDN line here, without a leading zero and without a country prefix.

Dial Prefix: If you need a prefix to get an public line, enter it here. This is only used on a internal S0 bus and the most common one is "0".

If you do not want to log all your ISDN traffic, uncheck **Start ISDN Log**.

Device Activation

ISDN Low-Level Configuration for contr0

ISDN Card Information
Vendor Abocom/Magitek
ISDN Card 2BD1

Driver: HiSax driver

ISDN Protocol
● Euro-ISDN (EDSS1)
○ 1TR6
○ Leased Line
○ NI1

Country: Germany
Code: +49
Area Code:
Dial Prefix:
☒ Start ISDN Log

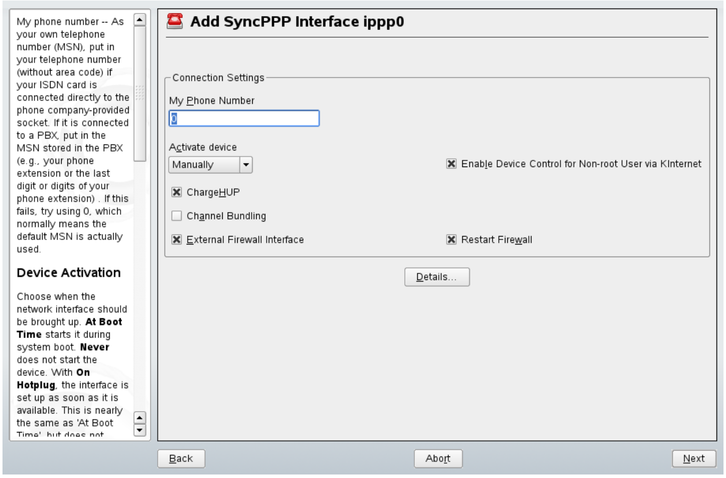
Activate device: At Boot Time

In the next dialog, shown in Figure 20.5, "ISDN Configuration" (page 326), select the protocol to use. The default is *Euro-ISDN (EDSS1)*, but for older or larger exchanges, select *1TR6*. If you are in the US, select *NI1*. Select your country in the relevant field. The corresponding country code then appears in the field next to it. Finally, provide your *Area Code* and the *Dial Prefix* if necessary.

Activate device defines how the ISDN interface should be started: *At Boot Time* causes the ISDN driver to be initialized each time the system boots. *Manually* requires you to load the ISDN driver as `root` with the command `rcisdn start`. *On Hotplug*, used for PCMCIA or USB devices, loads the driver after the device is plugged in. When finished with these settings, select *OK*.

In the next dialog, specify the interface type for your ISDN card and add ISPs to an existing interface. Interfaces may be either the `SyncPPP` or the `RawIP` type, but most ISPs operate in the `SyncPPP` mode, which is described below.

Figure 20.6 *ISDN Interface Configuration*



My phone number -- As your own telephone number (MSN), put in your telephone number (without area code) if your ISDN card is connected directly to the phone company-provided socket. If it is connected to a PBX, put in the MSN stored in the PBX (e.g., your phone extension or the last digit or digits of your phone extension) . If this fails, try using 0, which normally means the default MSN is actually used.

Device Activation

Choose when the network interface should be brought up. **At Boot Time** starts it during system boot. **Never** does not start the device. With **On Hotplug**, the interface is set up as soon as it is available. This is nearly the same as 'At Boot Time', but does not

Add SyncPPP Interface ippp0

Connection Settings

My Phone Number: 0

Activate device: Manually

☒ Enable Device Control for Non-root User via KInternet

☒ ChargeHUP

□ Channel Bundling

☒ External Firewall Interface

☒ Restart Firewall

The number to enter for *My Phone Number* depends on your particular setup:

ISDN Card Directly Connected to Phone Outlet

A standard ISDN line provides three phone numbers (called multiple subscriber numbers, or MSNs). If the subscriber asked for more, there may be up to 10. One of these MSNs must be entered here, but without your area code. If you enter the wrong number, your phone operator automatically falls back to the first MSN assigned to your ISDN line.

ISDN Card Connected to a Private Branch Exchange
Again, the configuration may vary depending on the equipment installed:

1. Smaller private branch exchanges (PBX) built for home purposes mostly use the Euro-ISDN (EDSS1) protocol for internal calls. These exchanges have an internal S0 bus and use internal numbers for the equipment connected to them.

   Use one of the internal numbers as your MSN. You should be able to use at least one of the exchange's MSNs that have been enabled for direct outward dialing. If this does not work, try a single zero. For further information, consult the documentation delivered with your phone exchange.

2. Larger phone exchanges designed for businesses normally use the 1TR6 protocol for internal calls. Their MSN is called EAZ and usually corresponds to the direct-dial number. For the configuration under Linux, it should be sufficient to enter the last digit of the EAZ. As a last resort, try each of the digits from 1 to 9.

For the connection to be terminated just before the next charge unit is due, enable *ChargeHUP*. However, remember that may not work with every ISP. You can also enable channel bundling (multilink PPP) by selecting the corresponding option. Finally, you can enable SuSEfirewall2 for your link by selecting *External Firewall Interface* and *Restart Firewall*. To enable the normal user without administrator permissions to activate or deactivate the interface, select the *Enable Device Control for Non-root user via KInternet*.

Details opens a dialog in which to implement more complex connection schemes, which are not relevant for normal home users. Leave the *Details* dialog by selecting *OK*.

In the next dialog, make IP address settings. If you have not been given a static IP by your provider, select *Dynamic IP Address*. Otherwise, use the fields provided to enter your host's local IP address and the remote IP address according to the specifications of your ISP. If the interface should be the default route to the Internet, select *Default Route*. Each host can only have one interface configured as the default route. Leave this dialog by selecting *Next*.

The following dialog allows you to set your country and select an ISP. The ISPs included in the list are call-by-call providers only. If your ISP is not in the list, select *New*. This opens the *Provider Parameters* dialog in which to enter all the details for your ISP.

When entering the phone number, do not include any blanks or commas among the digits. Finally, enter your login and the password as provided by the ISP. When finished, select *Next*.

To use *Dial on Demand* on a stand-alone workstation, also specify the name server (DNS server). Most ISPs support dynamic DNS, which means the IP address of a name server is sent by the ISP each time you connect. For a single workstation, however, you still need to provide a placeholder address like `192.168.22.99`. If your ISP does not support dynamic DNS, specify the name server IP addresses of the ISP. If desired, specify a time-out for the connection—the period of network inactivity (in seconds) after which the connection should be automatically terminated. Confirm your settings with *Next*. YaST displays a summary of the configured interfaces. To activate these settings, select *Finish*.

20.4.4 Cable Modem

In some countries it is quite common to access the Internet through the TV cable network. The TV cable subscriber usually gets a modem that is connected to the TV cable outlet on one side and to a computer network card on the other (using a 10Base-TG twisted pair cable). The cable modem then provides a dedicated Internet connection with a fixed IP address.

Depending on the instructions provided by your ISP, when configuring the network card either select *Dynamic Address* or *Statically assigned IP address*. Most providers today use DHCP. A static IP address often comes as part of a special business account.

For further information about the configuration of cable modems, read the Support Database article on the topic, which is available online at http://en.opensuse.org/SDB:Setting_Up_an_Internet_Connection_via_Cable_Modem_with_SuSE_Linux_8.0_or_Higher.

20.4.5 DSL

To configure your DSL device, select the *DSL* module from the YaST *Network Devices* section. This YaST module consists of several dialogs in which to set the parameters of DSL links based on one of the following protocols:

- PPP over Ethernet (PPPoE)

- PPP over ATM (PPPoATM)

- CAPI for ADSL (Fritz Cards)

- Point-to-Point Tunneling Protocol (PPTP)—Austria

In the *DSL Devices* tab of the *DSL Configuration* dialog, you will find a list of installed DSL devices. To change the configuration of a DSL device, select it in the list and click *Edit*. If you click *Add*, you can manually configure a new DSL device.

The configuration of a DSL connection based on PPPoE or PPTP requires that the corresponding network card has already been set up in the correct way. If you have not done so yet, first configure the card by selecting *Configure Network Cards* (see Section 20.4.1, "Configuring the Network Card with YaST" (page 316)). In the case of a DSL link, addresses may be assigned automatically but not via DHCP, which is why you should not enable the option *Dynamic Address*. Instead, enter a static dummy address for the interface, such as `192.168.22.1`. In *Subnet Mask*, enter `255.255.255.0`. If you are configuring a stand-alone workstation, leave *Default Gateway* empty.

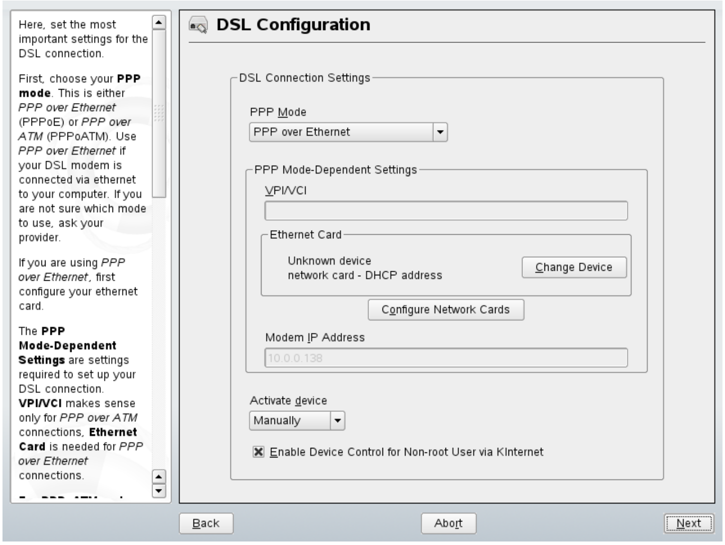
TIP

Values in *IP Address* and *Subnet Mask* are only placeholders. They are only needed to initialize the network card and do not represent the DSL link as such.

In the first DSL configuration dialog (see Figure 20.7, "DSL Configuration" (page 331)), select the *PPP Mode* and the *Ethernet Card* to which the DSL modem is connected (in most cases, this is `eth0`). Then use *Activate Device* to specify whether the DSL link should be established during the boot process. Click *Enable Device Control for Non-root User via KInternet* to authorize the normal user without root permissions to activate or deactivate the interface with KInternet.

In the next dialog you are able to select your country as well, and choose from a number of ISPs operating in it. The details of any subsequent dialogs of the DSL configuration depend on the options set so far, which is why they are only briefly mentioned in the following paragraphs. For details on the available options, read the detailed help available from the dialogs.

Figure 20.7 DSL Configuration



Here, set the most important settings for the DSL connection.

First, choose your **PPP mode**. This is either *PPP over Ethernet* (PPPoE) or *PPP over ATM* (PPPoATM). Use *PPP over Ethernet* if your DSL modem is connected via ethernet to your computer. If you are not sure which mode to use, ask your provider.

If you are using *PPP over Ethernet*, first configure your ethernet card.

The **PPP Mode-Dependent Settings** are settings required to set up your DSL connection. **VPI/VCI** makes sense only for *PPP over ATM* connections, **Ethernet Card** is needed for *PPP over Ethernet* connections.

DSL Configuration

DSL Connection Settings

PPP Mode: PPP over Ethernet

PPP Mode-Dependent Settings

VPI/VCI:

Ethernet Card: Unknown device network card - DHCP address

Modem IP Address: 10.0.0.138

Activate device: Manually

☒ Enable Device Control for Non-root User via KInternet

To use *Dial on Demand* on a stand-alone workstation, also specify the name server (DNS server). Most ISPs support dynamic DNS—the IP address of a name server is sent by the ISP each time you connect. For a single workstation, however, provide a placeholder address like `192.168.22.99`. If your ISP does not support dynamic DNS, enter the name server IP address provided by your ISP.

Idle Time-Out (seconds) defines a period of network inactivity after which to terminate the connection automatically. A reasonable time-out value is between 60 and 300 seconds. If *Dial on Demand* is disabled, it may be useful to set the time-out to zero to prevent automatic hang-up.

The configuration of T-DSL is very similar to the DSL setup. Just select *T-Online* as your provider and YaST opens the T-DSL configuration dialog. In this dialog, provide some additional information required for T-DSL—the line ID, the T-Online number, the user code, and your password. All of these should be included in the information you received after subscribing to T-DSL.

20.5 Configuring a Network Connection Manually

Manual configuration of the network software should always be the last alternative. Using YaST is recommended. However, this background information about the network configuration can also assist your work with YaST.

When the kernel detects a network card and creates a corresponding network interface, it assigns the device a name depending on the order of device discovery, or order of the loading of the kernel modules. The default kernel device names are only predictable in very simple or tightly controlled hardware environments. Systems which allow adding or removing hardware during runtime, or support automatic configuration of devices cannot expect stable network device names assigned by the kernel across reboots.

However, all system configuration tools rely on persistent interface names. The problem is solved by udev. udev maintains a database of known network interfaces and renames interfaces from their kernel assigned names to persistent names stored in the database. The udev database of network interfaces is stored in the file `/etc/udev/rules.d/70-persistent-net.rules`. Every line in the file describes one network interface and specifies its persistent name. System administrators can change the assigned names by editing the `NAME=""` entries. After the network device has been renamed to the configured name by udev, the `ifup` command applies the system configuration to the interface.

Table 20.5, “Manual Network Configuration Scripts” (page 332) summarizes the most important scripts involved in the network configuration.

***Table 20.5** Manual Network Configuration Scripts*

Command	Function
`if{up,down,status}`	The `if*` scripts start existing network interfaces or return the status of the specified interface. More information is available in the manual page of `ifup`.
`rcnetwork`	The `rcnetwork` script can be used to start, stop, or restart all network interfaces or just a specified one. Use `rcnetwork stop` to stop network interfaces,

Command	Function
	`rcnetwork start` to start network interfaces, and `rcnetwork restart` to restart them. If you want to stop, start or restart just one interface, use the command followed by the interface name, for example `rcnetwork restart eth0`. If no interface is specified, the firewall is stopped, started, or restarted along with the network interfaces. The `rcnetwork status` command displays the state of the interfaces, their IP addresses, and whether an DHCP client is running. With `rcnetwork stop-all-dhcp-clients` and `rcnetwork restart-all-dhcp-clients` you can stop or restart DHCP clients running on network interfaces.

More information about udev and persistent device names is available in Chapter 15, *Dynamic Kernel Device Management with udev* (page 227).

20.5.1 Configuration Files

This section provides an overview of the network configuration files and explains their purpose and the format used.

/etc/sysconfig/network/ifcfg-*

These files contain the configurations for network interfaces. They include information such as the start mode and the IP address. Possible parameters are described in the manual page of `ifup`. Additionally, all variables from the files `dhcp`, `wireless`, and `config` can be used in the `ifcfg-*` files if a general setting should be used for only one interface.

/etc/sysconfig/network/config, dhcp, wireless

The file config contains general settings for the behavior of ifup, ifdown, and ifstatus. dhcp contains settings for DHCP and wireless for wireless LAN cards. The variables in all three configuration files are commented and can also be used in ifcfg-* files, where they are treated with higher priority.

/etc/sysconfig/network/routes,ifroute-*

The static routing of TCP/IP packets is determined here. All the static routes required by the various system tasks can be entered in the /etc/sysconfig/network/routes file: routes to a host, routes to a host via a gateway, and routes to a network. For each interface that needs individual routing, define an additional configuration file: /etc/sysconfig/network/ifroute-*. Replace * with the name of the interface. The entries in the routing configuration files look like this:

```
# Destination     Dummy/Gateway     Netmask            Device
#
127.0.0.0         0.0.0.0           255.255.255.0      lo
204.127.235.0     0.0.0.0           255.255.255.0      eth0
default           204.127.235.41    0.0.0.0            eth0
207.68.156.51     207.68.145.45     255.255.255.255    eth1
192.168.0.0       207.68.156.51     255.255.0.0        eth1
```

The route's destination is in the first column. This column may contain the IP address of a network or host or, in the case of *reachable* name servers, the fully qualified network or hostname.

The second column contains the default gateway or a gateway through which a host or network can be accessed. The third column contains the netmask for networks or hosts behind a gateway. For example, the mask is 255.255.255.255 for a host behind a gateway.

The fourth column is only relevant for networks connected to the local host such as loopback, Ethernet, ISDN, PPP, and dummy device. The device name must be entered here.

An (optional) fifth column can be used to specify the type of a route. Columns that are not needed should contain a minus sign - to ensure that the parser correctly interprets the command. For details, refer to the routes(5) man page.

/etc/resolv.conf

The domain to which the host belongs is specified in this file (keyword search). Also listed is the status of the name server address to access (keyword nameserver). Multiple domain names can be specified. When resolving a name that is not fully qualified, an attempt is made to generate one by attaching the individual search entries. Use multiple name servers by entering several lines, each beginning with nameserver. Precede comments with # signs. YaST enters the specified name server in this file. Example 20.5, "/etc/resolv.conf" (page 335) shows what /etc/resolv.conf could look like.

Example 20.5 */etc/resolv.conf*

```
# Our domain
search example.com
#
# We use sun (192.168.0.20) as nameserver
nameserver 192.168.0.20
```

Some services, like pppd (wvdial), ipppd (isdn), dhcp (dhcpcd and dhclient), and pcmcia modify the file /etc/resolv.conf by means of the script modify_resolvconf. If the file /etc/resolv.conf has been temporarily modified by this script, it contains a predefined comment giving information about the service that modified it, the location where the original file has been backed up, and how to turn off the automatic modification mechanism. If /etc/resolv.conf is modified several times, the file includes modifications in a nested form. These can be reverted in a clean way even if this reversal takes place in an order different from the order in which modifications were introduced. Services that may need this flexibility include isdn and pcmcia.

If a service was not terminated in a normal, clean way, modify_resolvconf can be used to restore the original file. Also, on system boot, a check is performed to see whether there is an uncleaned, modified resolv.conf, for example, after a system crash, in which case the original (unmodified) resolv.conf is restored.

YaST uses the command modify_resolvconf check to find out whether resolv.conf has been modified and subsequently warns the user that changes will be lost after restoring the file. Apart from this, YaST does not rely on modify_resolvconf, which means that the impact of changing resolv.conf through YaST is the same as that of any manual change. In both cases, changes have a permanent effect. Modifications requested by the mentioned services are only temporary.

/etc/hosts

In this file, shown in Example 20.6, "/etc/hosts" (page 336), IP addresses are assigned to hostnames. If no name server is implemented, all hosts to which an IP connection will be set up must be listed here. For each host, enter a line consisting of the IP address, the fully qualified hostname, and the hostname into the file. The IP address must be at the beginning of the line and the entries separated by blanks and tabs. Comments are always preceded by the # sign.

Example 20.6 */etc/hosts*

```
127.0.0.1 localhost
192.168.0.20 sun.example.com sun
192.168.0.1 earth.example.com earth
```

/etc/networks

Here, network names are converted to network addresses. The format is similar to that of the hosts file, except the network names precede the addresses. See Example 20.7, "/etc/networks" (page 336).

Example 20.7 */etc/networks*

```
loopback     127.0.0.0
localnet     192.168.0.0
```

/etc/host.conf

Name resolution—the translation of host and network names via the *resolver* library—is controlled by this file. This file is only used for programs linked to libc4 or libc5. For current glibc programs, refer to the settings in /etc/nsswitch.conf. A parameter must always stand alone in its own line. Comments are preceded by a # sign. Table 20.6, "Parameters for /etc/host.conf" (page 336) shows the parameters available. A sample /etc/host.conf is shown in Example 20.8, " /etc/host.conf " (page 337).

Table 20.6 *Parameters for /etc/host.conf*

order *hosts*, *bind*	Specifies in which order the services are accessed for the name resolution. Available arguments are (separated by blank spaces or commas):

hosts: Searches the `/etc/hosts` file

bind: Accesses a name server

nis: Uses NIS

multi *on*/*off*
: Defines if a host entered in `/etc/hosts` can have multiple IP addresses.

nospoof *on*
spoofalert *on*/*off*
: These parameters influence the name server *spoofing*, but, apart from that, do not exert any influence on the network configuration.

trim *domainname*
: The specified domain name is separated from the hostname after hostname resolution (as long as the hostname includes the domain name). This option is useful if only names from the local domain are in the `/etc/hosts` file, but should still be recognized with the attached domain names.

Example 20.8 */etc/host.conf*

```
# We have named running
order hosts bind
# Allow multiple address
multi on
```

/etc/nsswitch.conf

The introduction of the GNU C Library 2.0 was accompanied by the introduction of the *Name Service Switch* (NSS). Refer to the `nsswitch.conf(5)` man page and *The GNU C Library Reference Manual* for details.

The order for queries is defined in the file `/etc/nsswitch.conf`. A sample `nsswitch.conf` is shown in Example 20.9, "`/etc/nsswitch.conf`" (page 338). Comments are introduced by # signs. In this example, the entry under the `hosts` database means that a request is sent to `/etc/hosts` (`files`) via DNS.

Example 20.9 */etc/nsswitch.conf*

```
passwd:     compat
group:      compat

hosts:      files dns
networks:   files dns

services:   db files
protocols:  db files

netgroup:   files
automount:  files nis
```

The “databases” available over NSS are listed in Table 20.7, “Databases Available via /etc/nsswitch.conf” (page 338). In addition, `automount`, `bootparams`, `netmasks`, and `publickey` are expected in the near future. The configuration options for NSS databases are listed in Table 20.8, “Configuration Options for NSS “Databases”” (page 339).

Table 20.7 *Databases Available via /etc/nsswitch.conf*

`aliases`	Mail aliases implemented by `sendmail`; see `man 5 aliases`.
`ethers`	Ethernet addresses.
`group`	For user groups, used by `getgrent`. See also the man page for `group`.
`hosts`	For hostnames and IP addresses, used by `gethostbyname` and similar functions.
`netgroup`	Valid host and user lists in the network for the purpose of controlling access permissions; see the `netgroup(5)` man page.
`networks`	Network names and addresses, used by `getnetent`.
`passwd`	User passwords, used by `getpwent`; see the `passwd(5)` man page.

protocols	Network protocols, used by getprotoent; see the protocols(5) man page.
rpc	Remote procedure call names and addresses, used by getrpcbyname and similar functions.
services	Network services, used by getservent.
shadow	Shadow passwords of users, used by getspnam; see the shadow(5) man page.

Table 20.8 *Configuration Options for NSS "Databases"*

files	directly access files, for example, /etc/aliases
db	access via a database
nis, nisplus	NIS, see also Chapter 25, *Using NIS* (page 401)
dns	can only be used as an extension for hosts and networks
compat	can only be used as an extension for passwd, shadow, and group

/etc/nscd.conf

This file is used to configure nscd (name service cache daemon). See the nscd(8) and nscd.conf(5) man pages. By default, the system entries of passwd and groups are cached by nscd. This is important for the performance of directory services, like NIS and LDAP, because otherwise the network connection needs to be used for every access to names or groups. hosts is not cached by default, because the mechanism in nscd to cache hosts makes the local system unable to trust forward and reverse lookup checks. Instead of asking nscd to cache names, set up a caching DNS server.

If the caching for passwd is activated, it usually takes about fifteen seconds until a newly added local user is recognized. Reduce this waiting time by restarting nscd with the command rcnscd restart.

/etc/HOSTNAME

This contains the hostname without the domain name attached. This file is read by several scripts while the machine is booting. It may only contain one line in which the hostname is set.

20.5.2 Testing the Configuration

Before you write your configuration to the configuration files, you can test it. To set up a test configuration, use the `ip` command. To test the connection, use the `ping` command. Older configuration tools, `ifconfig` and `route`, are also available.

The commands `ip`, `ifconfig`, and `route` change the network configuration directly without saving it in the configuration file. Unless you enter your configuration in the correct configuration files, the changed network configuration is lost on reboot.

Configuring a Network Interface with ip

`ip` is a tool to show and configure routing, network devices, policy routing, and tunnels. It was designed as a replacement for the older tools ifconfig and route.

`ip` is very complex tool. Its common syntax is `ip` *`options object command`*. You can work with the following objects:

link
: This object represents a network device.

address
: This object represents the IP address of device.

neighbour
: This object represents a ARP or NDISC cache entry.

route
: This object represents the routing table entry.

rule
: This object represents a rule in the routing policy database.

maddress
This object represents a multicast address.

mroute
This object represents a multicast routing cache entry.

tunnel
This object represents a tunnel over IP.

If no command is given, the default command is used, usually `list`.

Change the state of a device with the command `ip link set` *`device_name command`*. For example, to deactivate device eth0, enter `ip link seteth0 down`. To activate it again, use `ip link seteth0 up`.

After activating a device, you can configure it. To set the IP address, use `ip addr add` *`ip_address`* `+ dev` *`device_name`*. For example, to set the address of the interface eth0 to 192.168.12.154/30 with standard broadcast (option `brd`), enter `ip addr add 192.168.12.154/30 brd + dev eth0`.

To have a working connection, you must also configure the default gateway. To set a gateway for your system, enter `ip route add gateway_ip_address`. To translate one IP address to another, use `nat`: `ip route add nat ip_address via other_ip_address`.

To display all devices, use `ip link ls`. To display the running interfaces only, use `ip link ls up`. To print interface statistics for a device, enter `ip -s link ls device_name`. To view addresses of your devices, enter `ip addr`. In the output of the `ip addr`, also find information about MAC addresses of your devices. To show all routes, use `ip route show`.

For more information about using ip, enter `ip help` or see the `ip(8)` man page. The `help` option is also available for all ip objects. If, for example, you want to read help for `ip addr`, enter `ip addr help`. Find the ip manual in `/usr/share/doc/packages/iproute2/ip-cref.pdf`.

Testing a Connection with ping

The `ping` command is the standard tool for testing whether a TCP/IP connection works. It uses the ICMP protocol to send a small data packet, ECHO_REQUEST datagram, to the destination host, requesting an immediate reply. If this works, ping displays a message to that effect, which indicates that the network link is basically functioning.

ping does more than test only the function of the connection between two computers: it also provides some basic information about the quality of the connection. In Example 20.10, "Output of the Command ping" (page 342), you can see an example of the ping output. The second-to-last line contains information about number of transmitted packets, packet loss, and total time of ping running.

As the destination, you can use a hostname or IP address, for example, ping venus.example.com or ping 192.168.2.101. The program sends packets until you press Ctrl + C.

If you only need to check the functionality of the connection, you can limit the number of the packets with the -c option. For example to limit ping to three packets, enter ping -c 3 192.168.2.101.

Example 20.10 Output of the Command ping

```
ping -c 3 venus.example.com
PING venus.example.com (192.168.2.101) 56(84) bytes of data.
64 bytes from venus.example.com (192.168.2.101): icmp_seq=1 ttl=49 time=188
ms
64 bytes from venus.example.com (192.168.2.101): icmp_seq=2 ttl=49 time=184
ms
64 bytes from venus.example.com (192.168.2.101): icmp_seq=3 ttl=49 time=183
ms
--- venus.example.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 183.417/185.447/188.259/2.052 ms
```

The default interval between two packets is one second. To change the interval, ping provides option -i. For example to increase ping interval to ten seconds, enter ping -i 10 192.168.2.101.

In a system with multiple network devices, it is sometimes useful to send the ping through a specific interface address. To do so, use the -I option with the name of the selected device, for example, ping -I wlan1 192.168.2.101.

For more options and information about using ping, enter ping -h or see the ping (8) man page.

Configuring the Network with ifconfig

`ifconfig` is a traditional network configuration tool. In contrast to ip, you can use it only for interface configuration. If you want to configure routing, use `route`.

NOTE: ifconfig and ip

The program ifconfig is obsolete. Use ip instead.

Without arguments, ifconfig displays the status of the currently active interfaces. As you can see in Example 20.11, "Output of the ifconfig Command" (page 343), ifconfig has very well-arranged and detailed output. The output also contains information about the MAC address of your device, the value of `HWaddr`, in the first line.

Example 20.11 *Output of the ifconfig Command*

```
eth0      Link encap:Ethernet  HWaddr 00:08:74:98:ED:51
          inet6 addr: fe80::208:74ff:fe98:ed51/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:634735 errors:0 dropped:0 overruns:4 frame:0
          TX packets:154779 errors:0 dropped:0 overruns:0 carrier:1
          collisions:0 txqueuelen:1000
          RX bytes:162531992 (155.0 Mb)  TX bytes:49575995 (47.2 Mb)
          Interrupt:11 Base address:0xec80

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8559 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8559 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:533234 (520.7 Kb)  TX bytes:533234 (520.7 Kb)

wlan1     Link encap:Ethernet  HWaddr 00:0E:2E:52:3B:1D
          inet addr:192.168.2.4  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::20e:2eff:fe52:3b1d/64 Scope:Link
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:50828 errors:0 dropped:0 overruns:0 frame:0
          TX packets:43770 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:45978185 (43.8 Mb)  TX bytes:7526693 (7.1 Mb)
```

For more options and information about using ifconfig, enter `ifconfig -h` or see the `ifconfig (8)` man page.

Configuring Routing with route

route is a program for manipulating the IP routing table. You can use it to view your routing configuration and add or remove of routes.

NOTE: route and ip

The program route is obsolete. Use ip instead.

route is especially useful if you need quick and comprehensible information about your routing configuration to determine problems with routing. To view your current routing configuration, enter route -n as root.

Example 20.12 *Output of the route -n Command*

```
route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.20.0.0       *               255.255.248.0   U         0 0          0 eth0
link-local      *               255.255.0.0     U         0 0          0 eth0
loopback        *               255.0.0.0       U         0 0          0 lo
default         styx.exam.com   0.0.0.0         UG        0 0          0 eth0
```

For more options and information about using route, enter route -h or see the route (8) man page.

20.5.3 Start-Up Scripts

Apart from the configuration files described above, there are also various scripts that load the network programs while the machine is booting. These are started as soon as the system is switched to one of the *multiuser runlevels*. Some of these scripts are described in Table 20.9, “Some Start-Up Scripts for Network Programs” (page 344).

Table 20.9 *Some Start-Up Scripts for Network Programs*

/etc/init.d/network	This script handles the configuration of the network interfaces. If the network service was not started, no network interfaces are implemented.

`/etc/init.d/xinetd`	Starts xinetd. xinetd can be used to make server services available on the system. For example, it can start vsftpd whenever an FTP connection is initiated.
`/etc/init.d/portmap`	Starts the portmapper needed for the RPC server, such as an NFS server.
`/etc/init.d/nfsserver`	Starts the NFS server.
`/etc/init.d/postfix`	Controls the postfix process.
`/etc/init.d/ypserv`	Starts the NIS server.
`/etc/init.d/ypbind`	Starts the NIS client.

20.6 smpppd as Dial-up Assistant

Some home users do not have a dedicated line connecting them to the Internet. Instead, they use dial-up connections. Depending on the dial-up method (ISDN or DSL), the connection is controlled by ipppd or pppd. Basically, all that needs to be done to go online is to start these programs correctly.

If you have a flat-rate connection that does not generate any additional costs for the dial-up connection, simply start the respective daemon. Control the dial-up connection with a KDE applet or a command-line interface. If the Internet gateway is not the host you are using, you might want to control the dial-up connection by way of a network host.

This is where smpppd is involved. It provides a uniform interface for auxiliary programs and acts in two directions. First, it programs the required pppd or ipppd and controls its dial-up properties. Second, it makes various providers available to the user programs and transmits information about the current status of the connection. As smpppd can also be controlled by way of the network, it is suitable for controlling dial-up connections to the Internet from a workstation in a private subnetwork.

20.6.1 Configuring smpppd

The connections provided by smpppd are automatically configured by YaST. The actual dial-up programs KInternet and cinternet are also preconfigured. Manual settings are only required to configure additional features of smpppd, such as remote control.

The configuration file of smpppd is `/etc/smpppd.conf`. By default, it does not enable remote control. The most important options of this configuration file are:

open-inet-socket = *`yes|no`*
: To control smpppd via the network, this option must be set to `yes`. The port on which smpppd listens is `3185`. If this parameter is set to `yes`, the parameters `bind-address`, `host-range`, and `password` should also be set accordingly.

bind-address = *`ip address`*
: If a host has several IP addresses, use this parameter to determine at which IP address smpppd should accept connections. The default is to listen at all addresses.

host-range = *`min ip`* *`max ip`*
: The parameter `host-range` defines a network range. Hosts whose IP addresses are within this range are granted access to smpppd. All hosts not within this range are denied access.

password = *`password`*
: By assigning a password, limit the clients to authorized hosts. As this is a plaintext password, you should not overrate the security it provides. If no password is assigned, all clients are permitted to access smpppd.

slp-register = *`yes|no`*
: With this parameter, the smpppd service can be announced in the network via SLP.

More information about smpppd is available in the `smpppd(8)` and `smpppd.conf(5)` man pages.

20.6.2 Configuring KInternet, cinternet, and qinternet for Remote Use

KInternet, cinternet, and qinternet can be used to control a local or remote smpppd. cinternet is the command-line counterpart of the graphical KInternet. qinternet is basically the same as KInternet, but does not use the KDE libraries, so it can be used without KDE and must be installed separately. To prepare these utilities for use with a remote smpppd, edit the configuration file `/etc/smpppd-c.conf` manually or using KInternet. This file only uses three options:

sites = *`list of sites`*
: Here, tell the front-ends where to search for smpppd. The front-ends test the options in the order specified here. The `local` option orders the establishment of a connection to the local smpppd. `gateway` points to an smpppd on the gateway. The connection should be established as specified under `server` in `config-file`. `slp` orders the front-ends to connect to an smpppd found via SLP.

server = *`server`*
: Here, specify the host on which smpppd runs.

password = *`password`*
: Insert the password selected for smpppd.

If smpppd is active, you can now try to access it, for example, with `cinternet --verbose --interface-list`. If you experience difficulties at this point, refer to the `smpppd-c.conf(5)` and `cinternet(8)` man pages.

SLP Services in the Network 21

The *service location protocol* (SLP) was developed to simplify the configuration of networked clients within a local network. To configure a network client, including all required services, the administrator traditionally needs detailed knowledge of the servers available in the network. SLP makes the availability of selected services known to all clients in the local network. Applications that support SLP can use the information distributed and be configured automatically.

openSUSE® supports installation using installation sources provided with SLP and contains many system services with integrated support for SLP. YaST and Konqueror both have appropriate front-ends for SLP. You can use SLP to provide networked clients with central functions, such as an installation server, file server, or print server on your system.

IMPORTANT: SLP Support in openSUSE

Services that offer SLP support include cupsd, rsyncd, ypserv, openldap2, openwbem (CIM), ksysguardd, saned, kdm vnc login, smpppd, rpasswd, postfix, and sshd (via fish).

21.1 Installation

Only an SLP client and slptools are installed by default. If you want to provide services via SLP, install the package `openslp-server`. To install the package, start YaST and select *Software* > *Software Management*. Now choose *Filter* > *Patterns* and click

Misc. Server. Select `openslp-server`. Confirm the installation of the required packages to finish the installation process.

21.2 Activating SLP

slpd must run on your system to offer services with SLP. It is not necessary to start this daemon simply to make service inquiries. Like most system services in openSUSE, the slpd daemon is controlled by means of a separate `init` script. The daemon is inactive by default. To activate it for the duration of a session, run `rcslpd start` as `root` to start it and `rcslpd stop` to stop it. Perform a restart or status check with `restart` or `status`. If slpd should be active by default, enable slpd in YaST *System* > *System Services (Runlevel)* or run the `insserv slpd` command once as `root`. This automatically includes slpd in the set of services to start when the system boots.

21.3 SLP Front-Ends in openSUSE

To find services provided via SLP in your network, use an SLP front-end. openSUSE contains several front-ends:

slptool

slptool is a simple command line program that can be used to announce SLP inquiries in the network or announce proprietary services. `slptool --help` lists all available options and functions. slptool can also be called from scripts that process SLP information.

Konqueror

When used as a network browser, Konqueror can display all SLP services available in the local network at `slp:/`. Click the icons in the main window to obtain more detailed information about the relevant service. If you use Konqueror with `service:/`, click the relevant icon once in the browser window to set up a connection with the selected service.

21.4 Installation over SLP

If you offer an installation server with openSUSE installation media within your network, this can be registered with SLP. For details, see Section 1.2.1, “Setting Up an Installation Server Using YaST” (page 12). If SLP installation is selected, linuxrc starts an SLP inquiry after the system has booted from the selected boot medium and displays the sources found.

21.5 Providing Services via SLP

Many applications in openSUSE already have integrated SLP support through the use of the `libslp` library. If a service has not been compiled with SLP support, use one of the following methods to make it available via SLP:

Static Registration with `/etc/slp.reg.d`

Create a separate registration file for each new service. The following is an example of a file for registering a scanner service:

```
## Register a saned service on this system
## en means english language
## 65535 disables the timeout, so the service registration does
## not need refreshes
service:scanner.sane://$HOSTNAME:6566,en,65535
watch-port-tcp=6566
description=SANE scanner daemon
```

The most important line in this file is the *service URL*, which begins with `service:`. This contains the service type (`scanner.sane`) and the address under which the service is available on the server. *`$HOSTNAME`* is automatically replaced with the full hostname. The name of the TCP port on which the relevant service can be found follows, separated by a colon. Then enter the language in which the service should appear and the duration of registration in seconds. These should be separated from the service URL by commas. Set the value for the duration of registration between `0` and `65535`. `0` prevents registration. `65535` removes all restrictions.

The registration file also contains the two variables `watch-port-tcp` and `description`. `watch-port-tcp` links the SLP service announcement to whether the relevant service is active by having slpd check the status of the service.

The second variable contains a more precise description of the service that is displayed in suitable browsers.

Static Registration with `/etc/slp.reg`

The only difference from the procedure with `/etc/slp.reg.d` is the grouping of all services within a central file.

Dynamic Registration with slptool

If a service should be registered for SLP from proprietary scripts, use the slptool command line front-end.

21.6 For More Information

The following sources provide further information about SLP:

RFC 2608, 2609, 2610

RFC 2608 generally deals with the definition of SLP. RFC 2609 deals with the syntax of the service URLs used in greater detail and RFC 2610 deals with DHCP via SLP.

`http://www.openslp.org/`

The home page of the OpenSLP project.

`/usr/share/doc/packages/openslp`

This directory contains all available documentation for SLP, including a `README.SuSE` containing the openSUSE details, the RFCs, and two introductory HTML documents. Programmers who want to use the SLP functions should install the `openslp-devel` package to consult its supplied *Programmers Guide*.

The Domain Name System 22

DNS (domain name system) is needed to resolve the domain names and hostnames into IP addresses. In this way, the IP address 192.168.2.100 is assigned to the hostname `jupiter`, for example. Before setting up your own name server, read the general information about DNS in Section 20.3, "Name Resolution" (page 314). The following configuration examples refer to BIND.

22.1 DNS Terminology

Zone

The domain namespace is divided into regions called zones. For instance, if you have `example.com`, you have the `example` section, or zone, of the `com` domain.

DNS server

The DNS server is a server that maintains the name and IP information for a domain. You can have a primary DNS server for master zone, a secondary server for slave zone, or a slave server without any zones for caching.

Master zone DNS server

The master zone includes all hosts from your network and a DNS server master zone stores up-to-date records for all the hosts in your domain.

Slave zone DNS server

A slave zone is a copy of the master zone. The slave zone DNS server obtains its zone data with zone transfer operations from its master server. The slave zone DNS server responds authoritatively for the zone as long as it has valid (not expired) zone data. If the slave cannot obtain a new copy of the zone data, it stops responding for the zone.

Forwarder
: Forwarders are DNS servers to which your DNS server should send queries it cannot answer.

Record
: The record is information about name and IP address. Supported records and their syntax are described in BIND documentation. Some special records are:

    NS record
    : An NS record tells name servers which machines are in charge of a given domain zone.

    MX record
    : The MX (mail exchange) records describe the machines to contact for directing mail across the Internet.

    SOA record
    : SOA (Start of Authority) record is the first record in a zone file. The SOA record is used when using DNS to synchronize data between multiple computers.

22.2 Installation

To install a DNS server, start YaST and select *Software* > *Software Management*. Choose *Filter* > *Patterns* and select *DHCP and DNS Server*. Confirm the installation of the dependent packages to finish the installation process.

22.3 Configuration with YaST

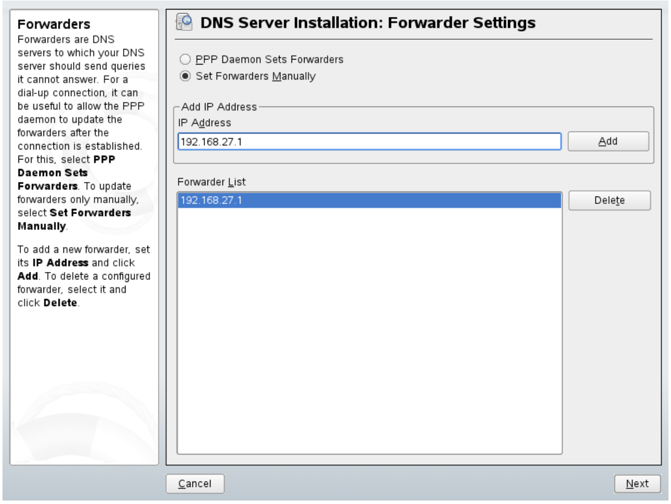
You can use the DNS module of YaST to configure a DNS server for your local network. When starting the module for the first time, a wizard starts, prompting you to make just a few basic decisions concerning administration of the server. Completing this initial setup produces a very basic server configuration that should be functioning in its essential aspects. The expert mode can be used to deal with more advanced configuration tasks.

22.3.1 Wizard Configuration

The wizard consists of three steps or dialogs. At the appropriate places in the dialogs, you are given the opportunity to enter the expert configuration mode.

1. When starting the module for the first time, the *Forwarder Settings* dialog, shown in Figure 22.1, "DNS Server Installation: Forwarder Settings" (page 355), opens. In it, decide whether the PPP daemon should provide a list of forwarders on dial-up via DSL or ISDN (*PPP Daemon Sets Forwarders*) or whether you want to supply your own list (*Set Forwarders Manually*).

Figure 22.1 *DNS Server Installation: Forwarder Settings*



Forwarders

Forwarders are DNS servers to which your DNS server should send queries it cannot answer. For a dial-up connection, it can be useful to allow the PPP daemon to update the forwarders after the connection is established. For this, select **PPP Daemon Sets Forwarders**. To update forwarders only manually, select **Set Forwarders Manually**.

To add a new forwarder, set its **IP Address** and click **Add**. To delete a configured forwarder, select it and click **Delete**.

DNS Server Installation: Forwarder Settings

○ PPP Daemon Sets Forwarders
● Set Forwarders Manually

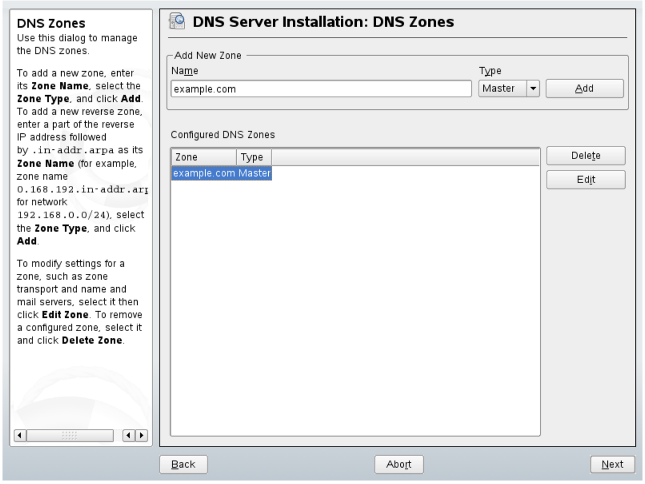
Add IP Address

IP Address: 192.168.27.1

Forwarder List: 192.168.27.1

2. The *DNS Zones* dialog consists of several parts and is responsible for the management of zone files, described in Section 22.6, "Zone Files" (page 367). For a new zone, provide a name for it in *Zone Name*. To add a reverse zone, the name must end in `.in-addr.arpa`. Finally, select the *Zone Type* (master or slave). See Figure 22.2, "DNS Server Installation: DNS Zones" (page 356). Click *Edit Zone* to configure other settings of an existing zone. To remove a zone, click *Delete Zone*.

Figure 22.2 *DNS Server Installation: DNS Zones*



DNS Zones

Use this dialog to manage the DNS zones.

To add a new zone, enter its **Zone Name**, select the **Zone Type**, and click **Add**. To add a new reverse zone, enter a part of the reverse IP address followed by `.in-addr.arpa` as its **Zone Name** (for example, zone name `0.168.192.in-addr.arp` for network `192.168.0.0/24`), select the **Zone Type**, and click **Add**.

To modify settings for a zone, such as zone transport and name and mail servers, select it then click **Edit Zone**. To remove a configured zone, select it and click **Delete Zone**.

DNS Server Installation: DNS Zones

Add New Zone

Name: example.com

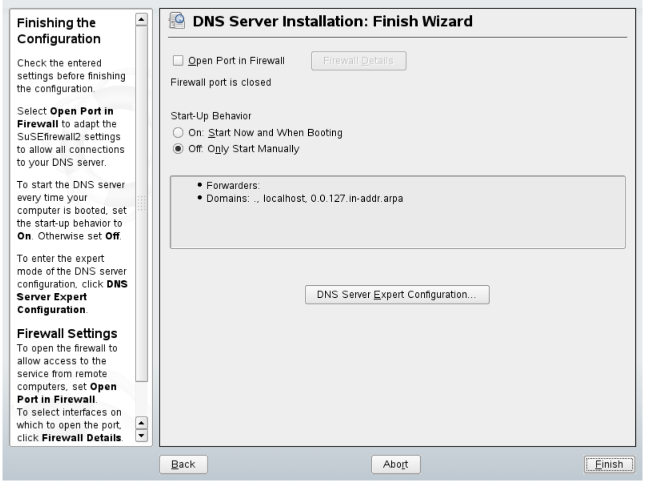
Type: Master

Configured DNS Zones

Zone	Type
example.com	Master

3 In the final dialog, you can open the ports for the DNS service in the firewall that is activated during the installation and decide whether DNS should be started. The expert configuration can also be accessed from this dialog. See Figure 22.3, “DNS Server Installation: Finish Wizard” (page 356).

Figure 22.3 *DNS Server Installation: Finish Wizard*



Finishing the Configuration

Check the entered settings before finishing the configuration.

Select **Open Port in Firewall** to adapt the SuSEfirewall2 settings to allow all connections to your DNS server.

To start the DNS server every time your computer is booted, set the start-up behavior to **On**. Otherwise set **Off**.

To enter the expert mode of the DNS server configuration, click **DNS Server Expert Configuration**.

Firewall Settings

To open the firewall to allow access to the service from remote computers, set **Open Port in Firewall**. To select interfaces on which to open the port, click **Firewall Details**.

DNS Server Installation: Finish Wizard

□ Open Port in Firewall

Firewall port is closed

Start-Up Behavior

○ On: Start Now and When Booting

● Off: Only Start Manually

- Forwarders:
- Domains: ., localhost, 0.0.127.in-addr.arpa

22.3.2 Expert Configuration

After starting the module, YaST opens a window displaying several configuration options. Completing it results in a DNS server configuration with the basic functions in place:

Starting the DNS Server

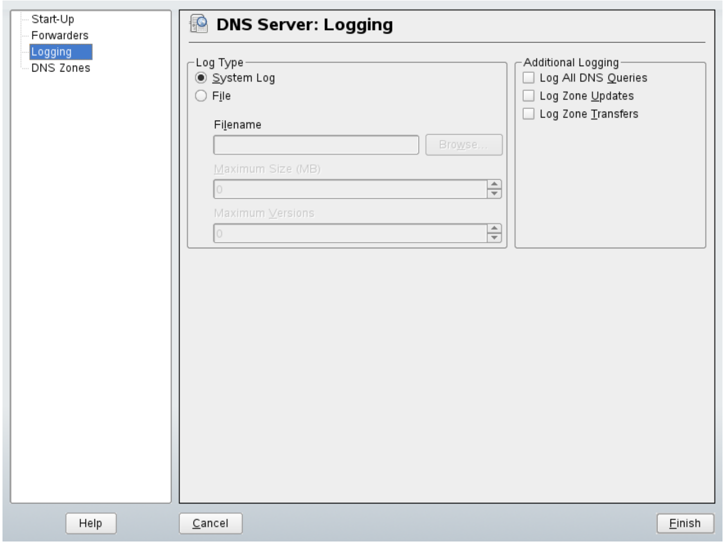
Under *Service Start*, define whether the DNS server should be started when the system boots (during booting the system) or manually. To start the DNS server immediately, select *Start DNS Server Now*. To stop the DNS server, select *Stop DNS Server Now*. To save the current settings, select *Save Settings and Restart DNS Server Now*. You can open the DNS port in the firewall with *Open Port in Firewall* and modify the firewall settings with *Firewall Details*.

Logging

To set what the DNS server should log and how, select *Logging*. Under *Log Type*, specify where the DNS server should write the log data. Use the systemwide log file `/var/log/messages` by selecting *System Log* or specify a different file by selecting *File*. In the latter case, additionally specify a name, the maximum file size in megabytes and the number of versions of log files to store.

Further options are available under *Additional Logging*. Enabling *Log All DNS Queries* causes *every* query to be logged, in which case the log file could grow extremely large. For this reason, it is not a good idea to enable this option for other than debugging purposes. To log the data traffic during zone updates between DHCP and DNS server, enable *Log Zone Updates*. To log the data traffic during a zone transfer from master to slave, enable *Log Zone Transfer*. See Figure 22.4, “DNS Server: Logging” (page 358).

Figure 22.4 DNS Server: Logging



DNS Server: Logging

Log Type
● System Log
○ File
Filename
Browse...
Maximum Size (MB) 0
Maximum Versions 0

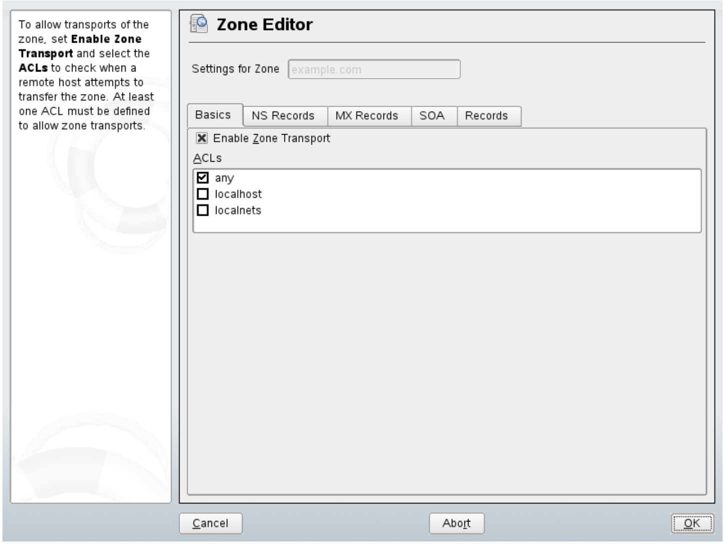
Additional Logging
□ Log All DNS Queries
□ Log Zone Updates
□ Log Zone Transfers

Adding a Slave Zone

To add a slave zone, select *DNS Zones*, choose the zone type *Slave*, write the name of the new zone, and click *Add*.

In the *Zone Editor* under *Master DNS Server IP*, specify the master from which the slave should fetch its data. To limit access to the server, select one of the ACLs from the list. See Figure 22.5, “DNS Server: Slave Zone Editor” (page 359).

Figure 22.5 *DNS Server: Slave Zone Editor*



To allow transports of the zone, set **Enable Zone Transport** and select the **ACLs** to check when a remote host attempts to transfer the zone. At least one ACL must be defined to allow zone transports.

Settings for Zone: example.com

☒ Enable Zone Transport

ACLs

☑ any
□ localhost
□ localnets

Adding a Master Zone

To add a master zone, select *DNS Zones*, choose the zone type *Master*, write the name of the new zone, and click *Add*.

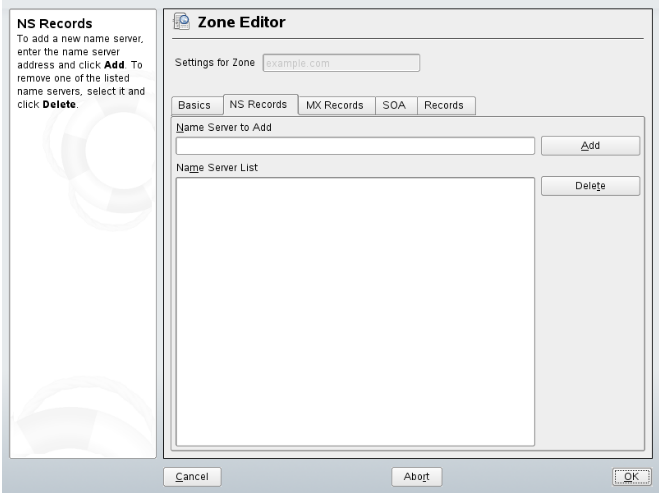
Editing a Master Zone

To edit a master zone, select *DNS Zones*, select the master zone from the table, and click *Edit*. The dialog consists of several pages: *Basics* (the one opened first), *NS Records*, *MX Records*, *SOA*, and *Records*.

Zone Editor (NS Records)

> This dialog allows you to define alternative name servers for the zones specified. Make sure that your own name server is included in the list. To add a record, enter its name under *Name Server to Add* then confirm with *Add*. See Figure 22.6, "DNS Server: Zone Editor (NS Records)" (page 360).

Figure 22.6 *DNS Server: Zone Editor (NS Records)*



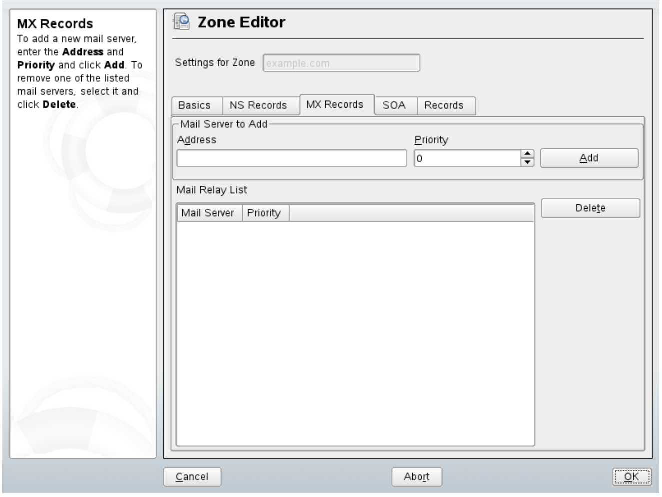
NS Records
To add a new name server, enter the name server address and click **Add**. To remove one of the listed name servers, select it and click **Delete**.

Settings for Zone: example.com

Zone Editor (MX Records)

To add a mail server for the current zone to the existing list, enter the corresponding address and priority value. After doing so, confirm by selecting *Add*. See Figure 22.7, "DNS Server: Zone Editor (MX Records)" (page 360).

Figure 22.7 *DNS Server: Zone Editor (MX Records)*



MX Records
To add a new mail server, enter the **Address** and **Priority** and click **Add**. To remove one of the listed mail servers, select it and click **Delete**.

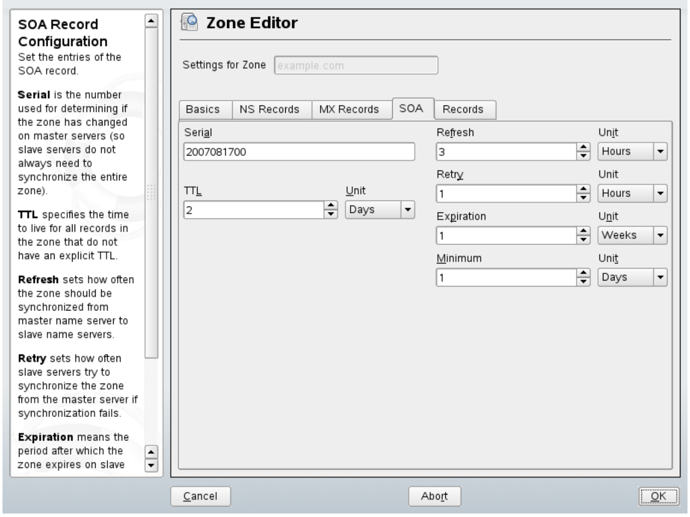
Settings for Zone: example.com

Priority: 0

Zone Editor (SOA)

This page allows you to create SOA (start of authority) records. For an explanation of the individual options, refer to Example 22.6, “File /var/lib/named/example.com.zone” (page 368).

Figure 22.8 DNS Server: Zone Editor (SOA)



SOA Record Configuration
Set the entries of the SOA record.

Serial is the number used for determining if the zone has changed on master servers (so slave servers do not always need to synchronize the entire zone).

TTL specifies the time to live for all records in the zone that do not have an explicit TTL.

Refresh sets how often the zone should be synchronized from master name server to slave name servers.

Retry sets how often slave servers try to synchronize the zone from the master server if synchronization fails.

Expiration means the period after which the zone expires on slave

Zone Editor

Settings for Zone: example.com

Serial: 2007081700
TTL: 2 Unit: Days
Refresh: 3 Unit: Hours
Retry: 1 Unit: Hours
Expiration: 1 Unit: Weeks
Minimum: 1 Unit: Days

Zone Editor (Records)

This dialog manages name resolution. In *Record Key*, enter the hostname then select its type. *A-Record* represents the main entry. The value for this should be an IP address. *CNAME* is an alias. Use the types *NS* and *MX* for detailed or partial records that expand on the information provided in the *NS Records* and *MX Records* tabs. These three types resolve to an existing A record. *PTR* is for reverse zones. It is the opposite of an A record.

22.4 Starting the Name Server BIND

On an openSUSE® system, the name server BIND (*Berkeley Internet name domain*) comes preconfigured so it can be started right after installation without any problem. If you already have a functioning Internet connection and have entered `127.0.0.1` as the name server address for `localhost` in `/etc/resolv.conf`, you normally already have a working name resolution without needing to know the DNS of the provider. BIND carries out name resolution via the root name server, a notably slower process. Normally, the DNS of the provider should be entered with its IP address in the configuration file /etc/named.conf under forwarders to ensure effective and secure name resolution. If this works so far, the name server runs as a pure *caching-only* name server. Only when you configure its own zones will it become a proper DNS. A simple example of this is included in the documentation in /usr/share/doc/packages/bind/config.

TIP: Automatic Adaptation of the Name Server Information

Depending on the type of Internet connection or the network connection, the name server information can automatically be adapted to the current conditions. To do this, set the variable MODIFY_NAMED_CONF_DYNAMICALLY in the file /etc/sysconfig/network/config to yes.

However, do not set up any official domains until assigned one by the responsible institution. Even if you have your own domain and it is managed by the provider, you are better off not using it, because BIND would otherwise not forward requests for this domain. The Web server at the provider, for example, would not be accessible for this domain.

To start the name server, enter the command rcnamed start as root. If “done” appears to the right in green, named, as the name server process is called, has been started successfully. Test the name server immediately on the local system with the host or dig programs, which should return localhost as the default server with the address 127.0.0.1. If this is not the case, /etc/resolv.conf probably contains an incorrect name server entry or the file does not exist at all. For the first test, enter host 127.0.0.1, which should always work. If you get an error message, use rcnamed status to see whether the server is actually running. If the name server does not start or behaves unexpectedly, you can usually find the cause in the log file /var/log/messages.

To use the name server of the provider or one already running on your network as the forwarder, enter the corresponding IP address or addresses in the options section under forwarders. The addresses included in Example 22.1, “Forwarding Options in named.conf” (page 363) are just examples. Adjust these entries to your own setup.

Example 22.1 *Forwarding Options in named.conf*

```
options {
        directory "/var/lib/named";
        forwarders { 10.11.12.13; 10.11.12.14; };
        listen-on { 127.0.0.1; 192.168.1.116; };
        allow-query { 127/8; 192.168/16 };
        notify no;
        };
```

The `options` entry is followed by entries for the zone, `localhost`, and `0.0.127.in-addr.arpa`. The `type hint` entry under "." should always be present. The corresponding files do not need to be modified and should work as they are. Also make sure that each entry is closed with a ";" and that the curly braces are in the correct places. After changing the configuration file `/etc/named.conf` or the zone files, tell BIND to reread them with `rcnamed reload`. Achieve the same by stopping and restarting the name server with `rcnamed restart`. Stop the server at any time by entering `rcnamed stop`.

22.5 The Configuration File /etc/named.conf

All the settings for the BIND name server itself are stored in the file `/etc/named.conf`. However, the zone data for the domains to handle, consisting of the hostnames, IP addresses, and so on, are stored in separate files in the `/var/lib/named` directory. The details of this are described later.

`/etc/named.conf` is roughly divided into two areas. One is the `options` section for general settings and the other consists of `zone` entries for the individual domains. A `logging` section and `acl` (access control list) entries are optional. Comment lines begin with a # sign or `//`. A minimal `/etc/named.conf` is shown in Example 22.2, "A Basic /etc/named.conf" (page 364).

Example 22.2 *A Basic /etc/named.conf*

```
options {
        directory "/var/lib/named";
        forwarders { 10.0.0.1; };
        notify no;
};

zone "localhost" in {
       type master;
       file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" in {
        type master;
        file "127.0.0.zone";
};

zone "." in {
        type hint;
        file "root.hint";
};
```

22.5.1 Important Configuration Options

directory "*filename*";

Specifies the directory in which BIND can find the files containing the zone data. Usually, this is `/var/lib/named`.

forwarders { *ip-address*; };

Specifies the name servers (mostly of the provider) to which DNS requests should be forwarded if they cannot be resolved directly. Replace *ip-address* with an IP address like `10.0.0.1`.

forward first;

Causes DNS requests to be forwarded before an attempt is made to resolve them via the root name servers. Instead of `forward first`, `forward only` can be written to have all requests forwarded and none sent to the root name servers. This makes sense for firewall configurations.

listen-on port 53 { 127.0.0.1; *ip-address*; };

Tells BIND on which network interfaces and port to accept client queries. `port 53` does not need to be specified explicitly, because `53` is the default port. Enter

127.0.0.1 to permit requests from the local host. If you omit this entry entirely, all interfaces are used by default.

listen-on-v6 port 53 {any; };

Tells BIND on which port it should listen for IPv6 client requests. The only alternative to any is none. As far as IPv6 is concerned, the server only accepts a wild card address.

query-source address * port 53;

This entry is necessary if a firewall is blocking outgoing DNS requests. This tells BIND to post requests externally from port 53 and not from any of the high ports above 1024.

query-source-v6 address * port 53;

Tells BIND which port to use for IPv6 queries.

allow-query { 127.0.0.1; *net*; };

Defines the networks from which clients can post DNS requests. Replace *net* with address information like 192.168/16. The /16 at the end is an abbreviated expression for the netmask, in this case, 255.255.0.0.

allow-transfer ! *;;

Controls which hosts can request zone transfers. In the example, such requests are completely denied with ! *. Without this entry, zone transfers can be requested from anywhere without restrictions.

statistics-interval 0;

In the absence of this entry, BIND generates several lines of statistical information per hour in /var/log/messages. Set it to 0 to suppress these statistics completely or set an interval in minutes.

cleaning-interval 720;

This option defines at which time intervals BIND clears its cache. This triggers an entry in /var/log/messages each time it occurs. The time specification is in minutes. The default is 60 minutes.

interface-interval 0;

BIND regularly searches the network interfaces for new or nonexistent interfaces. If this value is set to 0, this is not done and BIND only listens at the interfaces detected at start-up. Otherwise, the interval can be defined in minutes. The default is sixty minutes.

notify no;
: no prevents other name servers from being informed when changes are made to the zone data or when the name server is restarted.

22.5.2 Logging

What, how, and where logging takes place can be extensively configured in BIND. Normally, the default settings should be sufficient. Example 22.3, "Entry to Disable Logging" (page 366) shows the simplest form of such an entry and completely suppresses any logging.

Example 22.3 *Entry to Disable Logging*

```
logging {
        category default { null; };
};
```

22.5.3 Zone Entries

Example 22.4 *Zone Entry for example.com*

```
zone "example.com" in {
      type master;
      file "example.com.zone";
      notify no;
};
```

After zone, specify the name of the domain to administer (example.com) followed by in and a block of relevant options enclosed in curly braces, as shown in Example 22.4, "Zone Entry for example.com" (page 366). To define a *slave zone*, switch the type to slave and specify a name server that administers this zone as master (which, in turn, may be a slave of another master), as shown in Example 22.5, "Zone Entry for example.net" (page 366).

Example 22.5 *Zone Entry for example.net*

```
zone "example.net" in {
      type slave;
      file "slave/example.net.zone";
      masters { 10.0.0.1; };
};
```

The zone options:

type master;
: By specifying `master`, tell BIND that the zone is handled by the local name server. This assumes that a zone file has been created in the correct format.

type slave;
: This zone is transferred from another name server. It must be used together with `masters`.

type hint;
: The zone `.` of the `hint` type is used to set the root name servers. This zone definition can be left as is.

file `example.com.zone` or file “slave/example.net.zone”;
: This entry specifies the file where zone data for the domain is located. This file is not required for a slave, because this data is fetched from another name server. To differentiate master and slave files, use the directory `slave` for the slave files.

masters { *`server-ip-address`*; };
: This entry is only needed for slave zones. It specifies from which name server the zone file should be transferred.

allow-update {! *; };
: This option controls external write access, which would allow clients to make a DNS entry—something not normally desirable for security reasons. Without this entry, zone updates are not allowed at all. The above entry achieves the same because `! *` effectively bans any such activity.

22.6 Zone Files

Two types of zone files are needed. One assigns IP addresses to hostnames and the other does the reverse: it supplies a hostname for an IP address.

TIP: Using the Dot in Zone Files

The . has an important meaning in the zone files. If hostnames are given without a final ., the zone is appended. Complete hostnames specified with a full domain name must end with a . to avoid having the domain added to it again. A missing or wrongly placed dot is probably the most frequent cause of name server configuration errors.

The first case to consider is the zone file `example.com.zone`, responsible for the domain `example.com`, shown in Example 22.6, “File /var/lib/named/example.com.zone” (page 368).

***Example 22.6** File /var/lib/named/example.com.zone*

```
1.  $TTL 2D
2.  example.com. IN SOA      dns  root.example.com. (
3.               2003072441  ; serial
4.               1D          ; refresh
5.               2H          ; retry
6.               1W          ; expiry
7.               2D )        ; minimum
8.
9.               IN NS       dns
10.              IN MX       10 mail
11.
12. gate         IN A        192.168.5.1
13.              IN A        10.0.0.1
14. dns          IN A        192.168.1.116
15. mail         IN A        192.168.3.108
16. jupiter      IN A        192.168.2.100
17. venus        IN A        192.168.2.101
18. saturn       IN A        192.168.2.102
19. mercury      IN A        192.168.2.103
20. ntp          IN CNAME    dns
```

Line 1:

`$TTL` defines the default time to live that should apply to all the entries in this file. In this example, entries are valid for a period of two days (`2 D`).

Line 2:

This is where the SOA (start of authority) control record begins:

- The name of the domain to administer is `example.com` in the first position. This ends with ., because otherwise the zone would be appended a second time.

Alternatively, `@` can be entered here, in which case the zone would be extracted from the corresponding entry in `/etc/named.conf`.

- After `IN SOA` is the name of the name server in charge as master for this zone. The name is expanded from `dns` to `dns.example.com`, because it does not end with a `.`.
- An e-mail address of the person in charge of this name server follows. Because the `@` sign already has a special meaning, `.` is entered here instead. For `root@example.com` the entry must read `root.example.com.`. The `.` must be included at the end to prevent the zone from being added.
- The `(` includes all lines up to `)` into the SOA record.

Line 3:
The `serial number` is an arbitrary number that is increased each time this file is changed. It is needed to inform the secondary name servers (slave servers) of changes. For this, a 10 digit number of the date and run number, written as YYYYMMDDNN, has become the customary format.

Line 4:
The `refresh rate` specifies the time interval at which the secondary name servers verify the zone `serial number`. In this case, one day.

Line 5:
The `retry rate` specifies the time interval at which a secondary name server, in case of error, attempts to contact the primary server again. Here, two hours.

Line 6:
The `expiration time` specifies the time frame after which a secondary name server discards the cached data if it has not regained contact to the primary server. Here, it is a week.

Line 7:
The last entry in the SOA record specifies the `negative caching TTL`—the time for which results of unresolved DNS queries from other servers may be cached.

Line 9:
The `IN NS` specifies the name server responsible for this domain. `dns` is extended to `dns.example.com` because it does not end with a `.`. There can be several lines like this—one for the primary and one for each secondary name server. If

`notify` is not set to `no` in `/etc/named.conf`, all the name servers listed here are informed of the changes made to the zone data.

Line 10:
: The MX record specifies the mail server that accepts, processes, and forwards e-mails for the domain `example.com`. In this example, this is the host `mail.example.com`. The number in front of the hostname is the preference value. If there are multiple MX entries, the mail server with the smallest value is taken first and, if mail delivery to this server fails, an attempt is made with the next higher value.

Lines 12–19:
: These are the actual address records where one or more IP addresses are assigned to hostnames. The names are listed here without a `.` because they do not include their domain, so `example.com` is added to all of them. Two IP addresses are assigned to the host `gate`, because it has two network cards. Wherever the host address is a traditional one (IPv4), the record is marked with `AAAA`. If the address is an IPv6 address, the entry is marked with `AAAA 0`. The previous token for IPv6 addresses was only `AAAA`, which is now obsolete.

NOTE: IPv6 Syntax

The IPv6 record has a slightly different syntax than IPv4. Because of the fragmentation possibility, it is necessary to provide information about missed bits before the address. You must provide this information even if you want to use a completely unfragmented address. For the IPv4 record with the syntax

```
pluto IN           AAAA 2345:00C1:CA11:0001:1234:5678:9ABC:DEF0
pluto IN           AAAA 2345:00D2:DA11:0001:1234:5678:9ABC:DEF0
```

You need to add information about missing bits in IPv6 format. Because the example above is complete (does not miss any bits), the IPv6 format of this record is:

```
pluto  IN           AAAA 0 2345:00C1:CA11:0001:1234:5678:9ABC:DEF0
pluto  IN           AAAA 0 2345:00D2:DA11:0001:1234:5678:9ABC:DEF0
```

Do not use IPv4 addresses with IPv6 mapping.

Line 20:
: The alias `ntp` can be used to address `dns` (`CNAME` means *canonical name*).

The pseudodomain in-addr.arpa is used for the reverse lookup of IP addresses into hostnames. It is appended to the network part of the address in reverse notation. So 192.168 is resolved into 168.192.in-addr.arpa. See Example 22.7, “Reverse Lookup” (page 371).

Example 22.7 *Reverse Lookup*

```
1.  $TTL 2D
2.  168.192.in-addr.arpa.   IN SOA dns.example.com. root.example.com. (
3.                          2003072441      ; serial
4.                          1D              ; refresh
5.                          2H              ; retry
6.                          1W              ; expiry
7.                          2D )            ; minimum
8.
9.                          IN NS           dns.example.com.
10.
11. 1.5                     IN PTR          gate.example.com.
12. 100.3                   IN PTR          www.example.com.
13. 253.2                   IN PTR          cups.example.com.
```

Line 1:

$TTL defines the standard TTL that applies to all entries here.

Line 2:

The configuration file should activate reverse lookup for the network 192.168. Given that the zone is called 168.192.in-addr.arpa, should not be added to the hostnames. Therefore, all hostnames are entered in their complete form—with their domain and with a . at the end. The remaining entries correspond to those described for the previous example.com example.

Lines 3–7:

See the previous example for example.com.

Line 9:

Again this line specifies the name server responsible for this zone. This time, however, the name is entered in its complete form with the domain and a . at the end.

Lines 11–13:

These are the pointer records hinting at the IP addresses on the respective hosts. Only the last part of the IP address is entered at the beginning of the line, without the . at the end. Appending the zone to this (without the .in-addr.arpa) results in the complete IP address in reverse order.

Normally, zone transfers between different versions of BIND should be possible without any problem.

22.7 Dynamic Update of Zone Data

The term *dynamic update* refers to operations by which entries in the zone files of a master server are added, changed, or deleted. This mechanism is described in RFC 2136. Dynamic update is configured individually for each zone entry by adding an optional `allow-update` or `update-policy` rule. Zones to update dynamically should not be edited by hand.

Transmit the entries to update to the server with the command `nsupdate`. For the exact syntax of this command, check the manual page for nsupdate (`man 8 nsupdate`). For security reasons, any such update should be performed using TSIG keys as described in Section 22.8, "Secure Transactions" (page 372).

22.8 Secure Transactions

Secure transactions can be made with the help of transaction signatures (TSIGs) based on shared secret keys (also called TSIG keys). This section describes how to generate and use such keys.

Secure transactions are needed for communication between different servers and for the dynamic update of zone data. Making the access control dependent on keys is much more secure than merely relying on IP addresses.

Generate a TSIG key with the following command (for details, see `man dnssec-keygen`):

```
dnssec-keygen -a hmac-md5 -b 128 -n HOST host1-host2
```

This creates two files with names similar to these:

```
Khost1-host2.+157+34265.private Khost1-host2.+157+34265.key
```

The key itself (a string like `ejIkuCyyGJwwuN3xAteKgg==`) is found in both files. To use it for transactions, the second file (`Khost1-host2.+157+34265.key`) must be transferred to the remote host, preferably in a secure way (using scp, for example). On the remote server, the key must be included in the file /etc/named.conf to enable a secure communication between host1 and host2:

```
key host1-host2. {
 algorithm hmac-md5;
 secret ";ejIkuCyyGJwwuN3xAteKgg==;
};
```

WARNING: File Permissions of /etc/named.conf

Make sure that the permissions of /etc/named.conf are properly restricted. The default for this file is 0640, with the owner being root and the group named. As an alternative, move the keys to an extra file with specially limited permissions, which is then included from /etc/named.conf. To include an external file, use:

```
include  "filename"
```

Replace filename with an absolute path to your file with keys.

To enable the server host1 to use the key for host2 (which has the address 10.1.2.3 in this example), the server's /etc/named.conf must include the following rule:

```
server 10.1.2.3 {
  keys { host1-host2. ;};
};
```

Analogous entries must be included in the configuration files of host2.

Add TSIG keys for any ACLs (access control lists, not to be confused with file system ACLs) that are defined for IP addresses and address ranges to enable transaction security. The corresponding entry could look like this:

```
allow-update { key host1-host2. ;};
```

This topic is discussed in more detail in the *BIND Administrator Reference Manual* under update-policy.

22.9 DNS Security

DNSSEC, or DNS security, is described in RFC 2535. The tools available for DNSSEC are discussed in the BIND Manual.

A zone considered secure must have one or several zone keys associated with it. These are generated with `dnssec-keygen`, just like the host keys. The DSA encryption algorithm is currently used to generate these keys. The public keys generated should be included in the corresponding zone file with an `$INCLUDE` rule.

With the command `dnssec-makekeyset`, all keys generated are packaged into one set, which must then be transferred to the parent zone in a secure manner. On the parent, the set is signed with `dnssec-signkey`. The files generated by this command are then used to sign the zones with `dnssec-signzone`, which in turn generates the files to include for each zone in `/etc/named.conf`.

22.10 For More Information

For additional information, refer to the *BIND Administrator Reference Manual* from package `bind-doc`, which is installed under `/usr/share/doc/packages/bind/`. Consider additionally consulting the RFCs referenced by the manual and the manual pages included with BIND. `/usr/share/doc/packages/bind/README.SuSE` contains up-to-date information about BIND in openSUSE.

23 DHCP

The purpose of the *dynamic host configuration protocol* (DHCP) is to assign network settings centrally from a server rather than configuring them locally on each and every workstation. A host configured to use DHCP does not have control over its own static address. It is enabled to configure itself completely and automatically according to directions from the server. If you use the NetworkManager on the client side, you do not need to configure the client at all. This is useful if you have changing environments and only one interface active at a time. Never use NetworkManager on a machine that runs a DHCP server.

One way to configure a DHCP server is to identify each client using the hardware address of its network card (which should be fixed), then supply that client with identical settings each time it connects to the server. DHCP can also be configured to assign addresses to each interested client dynamically from an address pool set up for that purpose. In the latter case, the DHCP server tries to assign the same address to the client each time it receives a request, even over longer periods. This works only if the network does not have more clients than addresses.

DHCP makes life easier for system administrators. Any changes, even bigger ones, related to addresses and the network configuration in general can be implemented centrally by editing the server's configuration file. This is much more convenient than reconfiguring numerous workstations. Also it is much easier to integrate machines, particularly new machines, into the network, because they can be given an IP address from the pool. Retrieving the appropriate network settings from a DHCP server is especially useful in the case of laptops regularly used in different networks.

In this chapter, the DHCP server will run in the same subnet as the workstations, 192.168.2.0/24 with 192.168.2.1 as gateway. It has the fixed IP address 192.168.2.254

and serves two address ranges, 192.168.2.10 to 192.168.2.20 and 192.168.2.100 192.168.2.200;.

A DHCP server supplies not only the IP address and the netmask, but also the hostname, domain name, gateway, and name server addresses for the client to use. In addition to that, DHCP allows a number of other parameters to be configured in a centralized way, for example, a time server from which clients may poll the current time or even a print server.

23.1 Configuring a DHCP Server with YaST

In this version of openSUSE, the YaST DHCP module can be set up to store the server configuration locally (on the host that runs the DHCP server) or to have its configuration data managed by an LDAP server. If you want to use LDAP, setup your LDAP environment before configuring the DHCP server.

The YaST DHCP module allows you to set up your own DHCP server for the local network. The module can run in simple mode or expert mode.

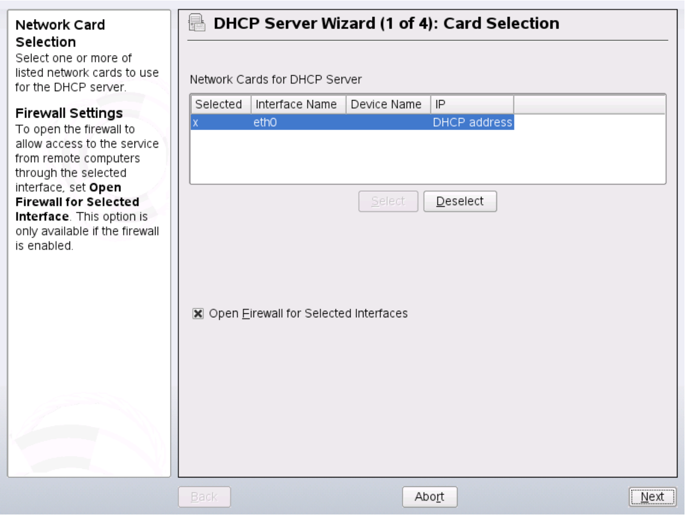
23.1.1 Initial Configuration (Wizard)

When the module is started for the first time, a wizard starts, prompting you to make a few basic decision concerning server administration. Completing this initial setup produces a very basic server configuration that should function in essential aspects. The expert mode can be used to deal with more advanced configuration tasks.

Card Selection

In the first step, YaST looks for the network interfaces available on your system then displays them in a list. From the list, select the interface on which the DHCP server should listen and click *Add*. After this, select *Open Firewall for Selected Interfaces* to open the firewall for this interface. See Figure 23.1, “DHCP Server: Card Selection” (page 377).

Figure 23.1 *DHCP Server: Card Selection*



Network Card Selection
Select one or more of listed network cards to use for the DHCP server.

Firewall Settings
To open the firewall to allow access to the service from remote computers through the selected interface, set **Open Firewall for Selected Interface**. This option is only available if the firewall is enabled.

DHCP Server Wizard (1 of 4): Card Selection

Network Cards for DHCP Server

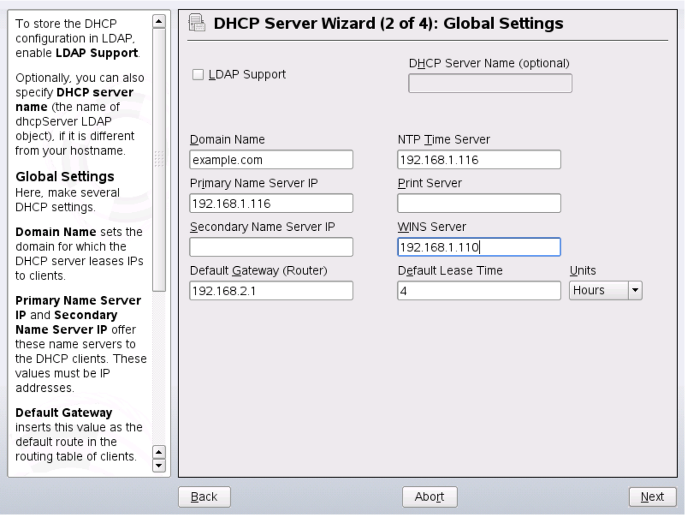
Selected	Interface Name	Device Name	IP
x	eth0		DHCP address

☒ Open Firewall for Selected Interfaces

Global Settings

Use the check box to determine whether your DHCP settings should be automatically stored by an LDAP server. In the entry fields, provide the network specifics for all clients the DHCP server should manage. These specifics are the domain name, address of a time server, addresses of the primary and secondary name server, addresses of a print and a WINS server (for a mixed network with both Windows and Linux clients), gateway address, and lease time. See Figure 23.2, “DHCP Server: Global Settings” (page 378).

Figure 23.2 *DHCP Server: Global Settings*



To store the DHCP configuration in LDAP, enable **LDAP Support**.

Optionally, you can also specify **DHCP server name** (the name of dhcpServer LDAP object), if it is different from your hostname.

Global Settings
Here, make several DHCP settings.

Domain Name sets the domain for which the DHCP server leases IPs to clients.

Primary Name Server IP and **Secondary Name Server IP** offer these name servers to the DHCP clients. These values must be IP addresses.

Default Gateway inserts this value as the default route in the routing table of clients.

DHCP Server Wizard (2 of 4): Global Settings

□ LDAP Support

DHCP Server Name (optional):

Domain Name: example.com

NTP Time Server: 192.168.1.116

Primary Name Server IP: 192.168.1.116

Print Server:

Secondary Name Server IP:

WINS Server: 192.168.1.110

Default Gateway (Router): 192.168.2.1

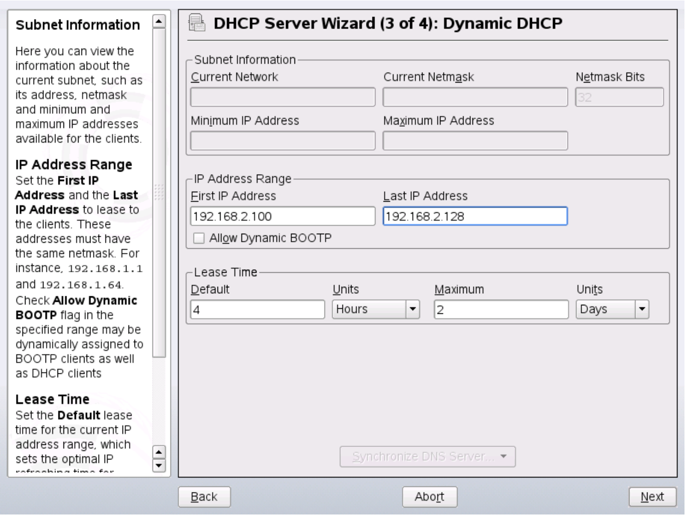
Default Lease Time: 4

Units: Hours

Dynamic DHCP

In this step, configure how dynamic IP addresses should be assigned to clients. To do so, specify an IP range from which the server can assign addresses to DHCP clients. All these addresses must be covered by the same netmask. Also specify the lease time during which a client may keep its IP address without needing to request an extension of the lease. Optionally, specify the maximum lease time—the period during which the server reserves an IP address for a particular client. See Figure 23.3, “DHCP Server: Dynamic DHCP” (page 379).

Figure 23.3 DHCP Server: Dynamic DHCP



Subnet Information

Here you can view the information about the current subnet, such as its address, netmask and minimum and maximum IP addresses available for the clients.

IP Address Range
Set the **First IP Address** and the **Last IP Address** to lease to the clients. These addresses must have the same netmask. For instance, `192.168.1.1` and `192.168.1.64`. Check **Allow Dynamic BOOTP** flag in the specified range may be dynamically assigned to BOOTP clients as well as DHCP clients

Lease Time
Set the **Default** lease time for the current IP address range, which sets the optimal IP

DHCP Server Wizard (3 of 4): Dynamic DHCP

Subnet Information
Current Network:
Current Netmask:
Netmask Bits: 32
Minimum IP Address:
Maximum IP Address:

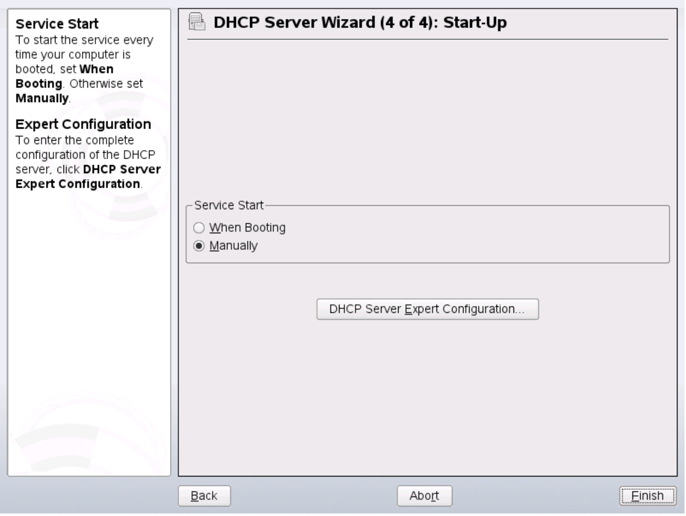
IP Address Range
First IP Address: 192.168.2.100
Last IP Address: 192.168.2.128
□ Allow Dynamic BOOTP

Lease Time
Default: 4
Units: Hours
Maximum: 2
Units: Days

Finishing the Configuration and Setting the Start Mode

After the third part of the configuration wizard, a last dialog is shown in which you can define how the DHCP server should be started. Here, specify whether to start the DHCP server automatically when the system is booted or manually when needed (for example, for test purposes). Click *Finish* to complete the configuration of the server. See Figure 23.4, “DHCP Server: Start-Up” (page 380). Alternatively, you can select *Host Management* from the tree structure on the left to configure special host management features in addition to the basic configuration (see Figure 23.5, “DHCP Server: Host Management” (page 381)).

Figure 23.4 *DHCP Server: Start-Up*



Service Start
To start the service every time your computer is booted, set **When Booting**. Otherwise set **Manually**.

Expert Configuration
To enter the complete configuration of the DHCP server, click **DHCP Server Expert Configuration**.

DHCP Server Wizard (4 of 4): Start-Up

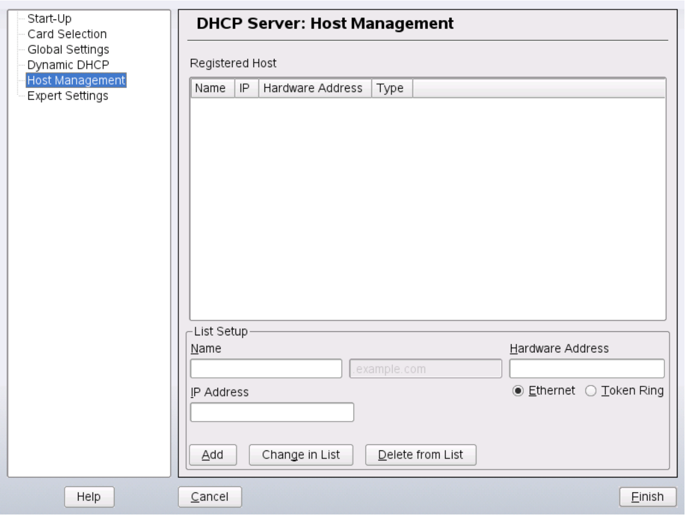
Service Start

○ When Booting
● Manually

Host Management

Instead of using dynamic DHCP in the way described in the preceding sections, you can also configure the server to assign addresses in quasi-static fashion. To do so, use the entry fields provided in the lower part to specify a list of the clients to manage in this way. Specifically, provide the *Name* and the *IP Address* to give to such a client, the *Hardware Address*, and the *Network Type* (token ring or ethernet). Modify the list of clients, which is shown in the upper part, with *Add*, *Edit*, and *Delete from List*. See Figure 23.5, “DHCP Server: Host Management” (page 381).

Figure 23.5 *DHCP Server: Host Management*



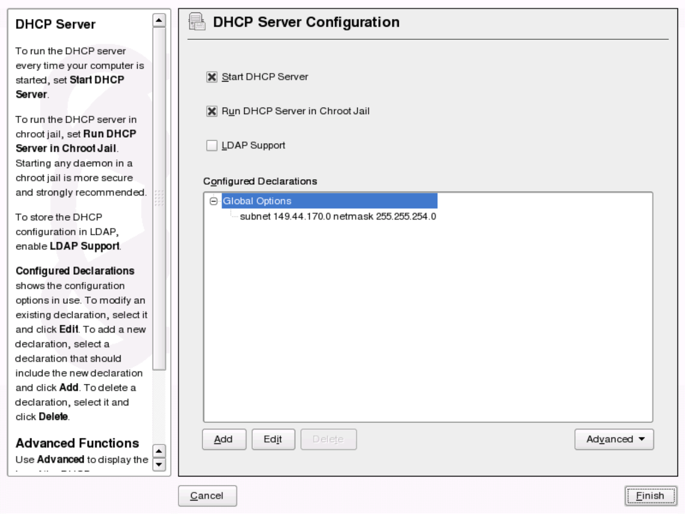
23.1.2 Expert Configuration

In addition to the configuration method discussed earlier, there is also an expert configuration mode that allows you to tweak the DHCP server setup in every detail. Start the expert configuration by selecting *Expert Settings* in the tree view in the left part of the dialog.

Chroot Environment and Declarations

In this first dialog, make the existing configuration editable by selecting *Start DHCP Server*. An important feature of the behavior of the DHCP server is its ability to run in a chroot environment, or chroot jail, to secure the server host. If the DHCP server should ever be compromised by an outside attack, the attacker will still be behind bars in the chroot jail, which prevents him from touching the rest of the system. The lower part of the dialog displays a tree view with the declarations that have already been defined. Modify these with *Add*, *Delete*, and *Edit*. Selecting *Advanced* takes you to additional expert dialogs. See Figure 23.6, "DHCP Server: Chroot Jail and Declarations" (page 382). After selecting *Add*, define the type of declaration to add. With *Advanced*, view the log file of the server, configure TSIG key management, and adjust the configuration of the firewall according to the setup of the DHCP server.

Figure 23.6 DHCP Server: Chroot Jail and Declarations



DHCP Server

To run the DHCP server every time your computer is started, set **Start DHCP Server**.

To run the DHCP server in chroot jail, set **Run DHCP Server in Chroot Jail**. Starting any daemon in a chroot jail is more secure and strongly recommended.

To store the DHCP configuration in LDAP, enable **LDAP Support**.

Configured Declarations shows the configuration options in use. To modify an existing declaration, select it and click **Edit**. To add a new declaration, select a declaration that should include the new declaration and click **Add**. To delete a declaration, select it and click **Delete**.

Advanced Functions

Use **Advanced** to display the

DHCP Server Configuration

☒ Start DHCP Server

☒ Run DHCP Server in Chroot Jail

□ LDAP Support

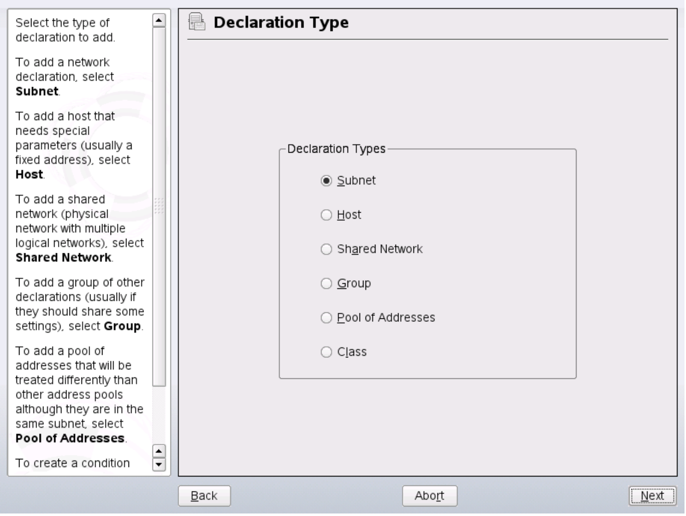
Configured Declarations

⊖ Global Options
  subnet 149.44.170.0 netmask 255.255.254.0

Selecting the Declaration Type

The *Global Options* of the DHCP server are made up of a number of declarations. This dialog lets you set the declaration types *Subnet*, *Host*, *Shared Network*, *Group*, *Pool of Addresses*, and *Class*. This example shows the selection of a new subnetwork (see Figure 23.7, “DHCP Server: Selecting a Declaration Type” (page 383)).

Figure 23.7 DHCP Server: Selecting a Declaration Type



Select the type of declaration to add.

To add a network declaration, select **Subnet**.

To add a host that needs special parameters (usually a fixed address), select **Host**.

To add a shared network (physical network with multiple logical networks), select **Shared Network**.

To add a group of other declarations (usually if they should share some settings), select **Group**.

To add a pool of addresses that will be treated differently than other address pools although they are in the same subnet, select **Pool of Addresses**.

To create a condition

Declaration Type

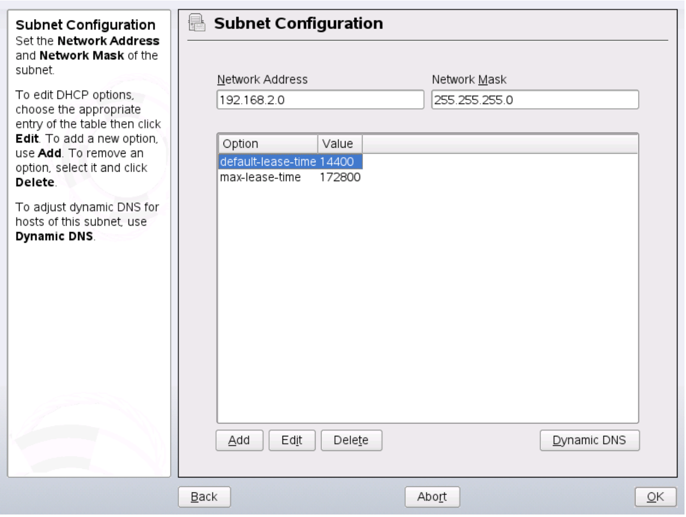
Declaration Types

- ● Subnet
- ○ Host
- ○ Shared Network
- ○ Group
- ○ Pool of Addresses
- ○ Class

Subnet Configuration

This dialog allows you specify a new subnet with its IP address and netmask. In the middle part of the dialog, modify the DHCP server start options for the selected subnet using *Add*, *Edit*, and *Delete*. To set up dynamic DNS for the subnet, select *Dynamic DNS*.

Figure 23.8 *DHCP Server: Configuring Subnets*



Subnet Configuration
Set the **Network Address** and **Network Mask** of the subnet.

To edit DHCP options, choose the appropriate entry of the table then click **Edit**. To add a new option, use **Add**. To remove an option, select it and click **Delete**.

To adjust dynamic DNS for hosts of this subnet, use **Dynamic DNS**.

Subnet Configuration

Network Address: 192.168.2.0

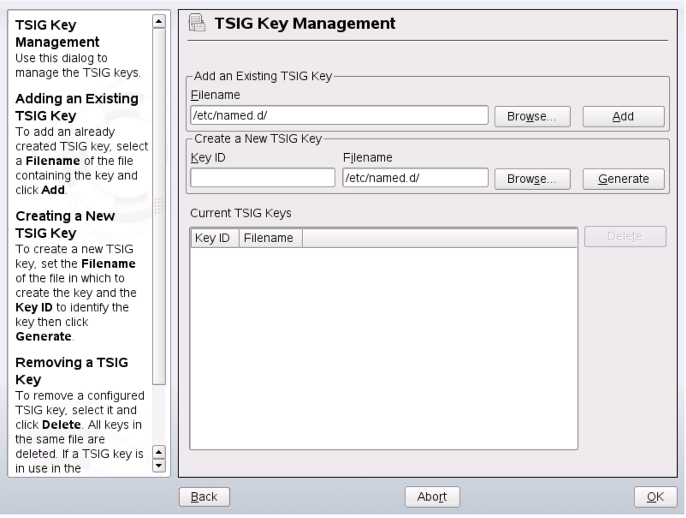
Network Mask: 255.255.255.0

Option	Value
default-lease-time	14400
max-lease-time	172800

TSIG Key Management

If you chose to configure dynamic DNS in the previous dialog, you can now configure the key management for a secure zone transfer. Selecting *OK* takes you to another dialog in which to configure the interface for dynamic DNS (see Figure 23.10, “DHCP Server: Interface Configuration for Dynamic DNS” (page 386)).

Figure 23.9 *DHCP Server: TSIG Configuration*



TSIG Key Management
Use this dialog to manage the TSIG keys.

Adding an Existing TSIG Key
To add an already created TSIG key, select a **Filename** of the file containing the key and click **Add**.

Creating a New TSIG Key
To create a new TSIG key, set the **Filename** of the file in which to create the key and the **Key ID** to identify the key then click **Generate**.

Removing a TSIG Key
To remove a configured TSIG key, select it and click **Delete**. All keys in the same file are deleted. If a TSIG key is in use in the

TSIG Key Management

Add an Existing TSIG Key
Filename: /etc/named.d/

Create a New TSIG Key
Key ID:
Filename: /etc/named.d/

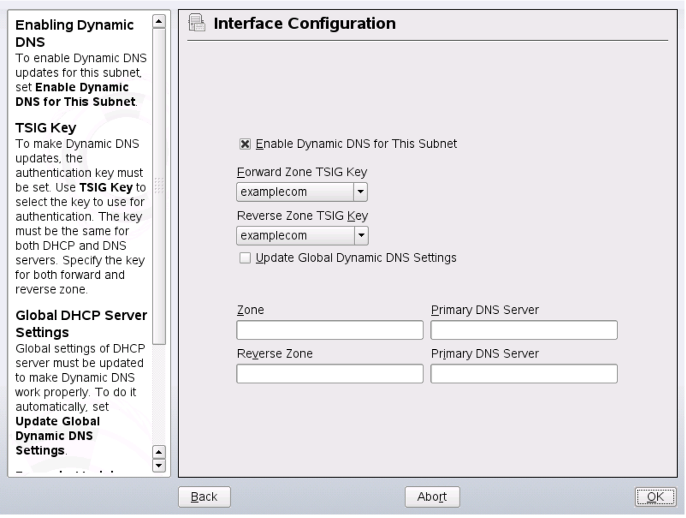
Current TSIG Keys

Key ID	Filename

Dynamic DNS: Interface Configuration

You can now activate dynamic DNS for the subnet by selecting *Enable Dynamic DNS for This Subnet*. After doing so, use the drop-down list to choose the TSIG keys for forward and reverse zones, making sure that keys are the same for the DNS and the DHCP server. With *Update Global Dynamic DNS Settings*, enable the automatic update and adjustment of the global DHCP server settings according to the dynamic DNS environment. Finally, define which forward and reverse zones should be updated per dynamic DNS, specifying the name of the primary name server for each of the two zones. If the name server runs on the same host as the DHCP server, you can leave these fields blank. Selecting *Ok* returns to the subnet configuration dialog (see Figure 23.8, "DHCP Server: Configuring Subnets" (page 384)). Selecting *Ok* again returns to the original expert configuration dialog.

Figure 23.10 *DHCP Server: Interface Configuration for Dynamic DNS*



Enabling Dynamic DNS
To enable Dynamic DNS updates for this subnet, set **Enable Dynamic DNS for This Subnet**

TSIG Key
To make Dynamic DNS updates, the authentication key must be set. Use **TSIG Key** to select the key to use for authentication. The key must be the same for both DHCP and DNS servers. Specify the key for both forward and reverse zone.

Global DHCP Server Settings
Global settings of DHCP server must be updated to make Dynamic DNS work properly. To do it automatically, set **Update Global Dynamic DNS Settings**.

Interface Configuration

☒ Enable Dynamic DNS for This Subnet

Forward Zone TSIG Key: examplecom

Reverse Zone TSIG Key: examplecom

☐ Update Global Dynamic DNS Settings

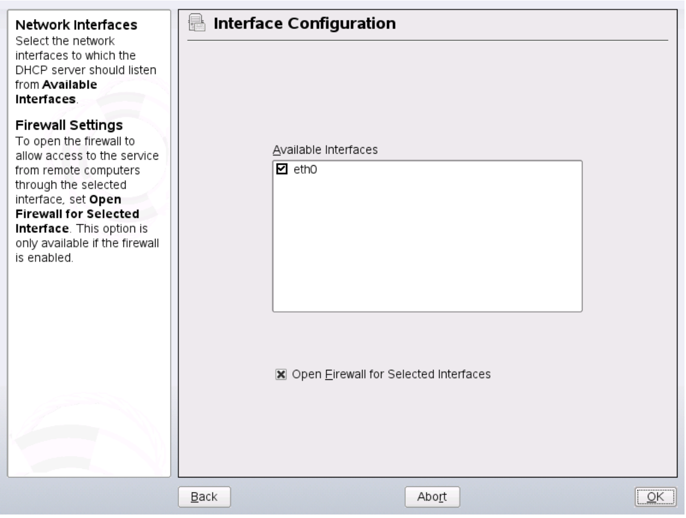
Zone:
Primary DNS Server:

Reverse Zone:
Primary DNS Server:

Network Interface Configuration

To define the interfaces where the DHCP server should listen and to adjust the firewall configuration, select *Advanced* > *Interface Configuration* from the expert configuration dialog. From the list of interfaces displayed, select one or more that should be attended by the the DHCP server. If clients in all of the subnets should be able to communicate with the server and the server host also runs a firewall, adjust the firewall accordingly. To do so, select *Adapt Firewall Settings*. YaST then adjusts the rules of SuSEfirewall2 to the new conditions (see Figure 23.11, “DHCP Server: Network Interface and Firewall” (page 387)), after which you can return to the original dialog by selecting *Ok*.

Figure 23.11 DHCP Server: Network Interface and Firewall



Network Interfaces
Select the network interfaces to which the DHCP server should listen from **Available Interfaces**.

Firewall Settings
To open the firewall to allow access to the service from remote computers through the selected interface, set **Open Firewall for Selected Interface**. This option is only available if the firewall is enabled.

Interface Configuration

Available Interfaces
☑ eth0

☒ Open Firewall for Selected Interfaces

After completing all configuration steps, close the dialog with *Ok*. Now the server is started with its new configuration. configuration.

23.2 DHCP Software Packages

Both a DHCP server and DHCP clients are available for openSUSE. The DHCP server available is dhcpd (published by the Internet Systems Consortium). On the client side, choose between two different DHCP client programs: `dhcp-client` (also from ISC) and the DHCP client daemon in the `dhcpcd` package.

openSUSE installs dhcpcd by default. The program is very easy to handle and is launched automatically on each system boot to watch for a DHCP server. It does not need a configuration file to do its job and works out of the box in most standard setups. For more complex situations, use the ISC dhcp-client, which is controlled by means of the configuration file `/etc/dhclient.conf`.

23.3 The DHCP Server dhcpd

The core of any DHCP system is the dynamic host configuration protocol daemon. This server *leases* addresses and watches how they are used, according to the settings defined in the configuration file `/etc/dhcpd.conf`. By changing the parameters and values in this file, a system administrator can influence the program's behavior in numerous ways. Look at the basic sample `/etc/dhcpd.conf` file in Example 23.1, “The Configuration File /etc/dhcpd.conf” (page 388).

***Example 23.1** The Configuration File /etc/dhcpd.conf*

```
default-lease-time 600;         # 10 minutes
max-lease-time 7200;            # 2  hours

option domain-name "example.com";
option domain-name-servers 192.168.1.116;
option broadcast-address 192.168.2.255;
option routers 192.168.2.1;
option subnet-mask 255.255.255.0;

subnet 192.168.2.0 netmask 255.255.255.0
 {
  range 192.168.2.10 192.168.2.20;
  range 192.168.2.100 192.168.2.200;
 }
```

This simple configuration file should be sufficient to get the DHCP server to assign IP addresses in the network. Make sure that a semicolon is inserted at the end of each line, because otherwise dhcpd is not started.

The sample file can be divided into three sections. The first one defines how many seconds an IP address is leased to a requesting client by default (`default-lease-time`) before it should apply for renewal. The section also includes a statement of the maximum period for which a machine may keep an IP address assigned by the DHCP server without applying for renewal (`max-lease-time`).

In the second part, some basic network parameters are defined on a global level:

- The line `option domain-name` defines the default domain of your network.

- With the entry `option domain-name-servers`, specify up to three values for the DNS servers used to resolve IP addresses into hostnames and vice versa. Ideally, configure a name server on your machine or somewhere else in your network before setting up DHCP. That name server should also define a hostname for each dynamic address and vice versa. To learn how to configure your own name server, read Chapter 22, *The Domain Name System* (page 353).

- The line `option broadcast-address` defines the broadcast address the requesting client should use.

- With `option routers`, set where the server should send data packets that cannot be delivered to a host on the local network (according to the source and target host address and the subnet mask provided). In most cases, especially in smaller networks, this router is identical to the Internet gateway.

- With `option subnet-mask`, specify the netmask assigned to clients.

The last section of the file defines a network, including a subnet mask. To finish, specify the address range that the DHCP daemon should use to assign IP addresses to interested clients. In Example 23.1, "The Configuration File /etc/dhcpd.conf" (page 388), clients may be given any address between `192.168.2.10` and `192.168.2.20` as well as `192.168.2.100` and `192.168.2.200`.

After editing these few lines, you should be able to activate the DHCP daemon with the command `rcdhcpd start`. It will be ready for use immediately. Use the command `rcdhcpd check-syntax` to perform a brief syntax check. If you encounter any unexpected problems with your configuration—the server aborts with an error or does not return `done` on start—you should be able to find out what has gone wrong by looking for information either in the main system log `/var/log/messages` or on console 10 (Ctrl + Alt + F10).

On a default openSUSE system, the DHCP daemon is started in a chroot environment for security reasons. The configuration files must be copied to the chroot environment so the daemon can find them. Normally, there is no need to worry about this because the command `rcdhcpd start` automatically copies the files.

23.3.1 Clients with Fixed IP Addresses

DHCP can also be used to assign a predefined, static address to a specific client. Addresses assigned explicitly always take priority over dynamic addresses from the pool. A static address never expires in the way a dynamic address would, for example, if there were not enough addresses available and the server needed to redistribute them among clients.

To identify a client configured with a static address, dhcpd uses the hardware address, which is a globally unique, fixed numerical code consisting of six octet pairs for the identification of all network devices (for example, `00:30:6E:08:EC:80`). If the respective lines, like the ones in Example 23.2, “Additions to the Configuration File” (page 390), are added to the configuration file of Example 23.1, “The Configuration File /etc/dhcpd.conf” (page 388), the DHCP daemon always assigns the same set of data to the corresponding client.

Example 23.2 *Additions to the Configuration File*

```
host jupiter {
hardware ethernet 00:30:6E:08:EC:80;
fixed-address 192.168.2.100;
}
```

The name of the respective client (`host` *`hostname`*, here `jupiter`) is entered in the first line and the MAC address in the second line. On Linux hosts, find the MAC address with the command `ip link show` followed by the network device (for example, `eth0`). The output should contain something like

```
link/ether 00:30:6E:08:EC:80
```

In the preceding example, a client with a network card having the MAC address `00:30:6E:08:EC:80` is assigned the IP address `192.168.2.100` and the hostname `jupiter` automatically. The type of hardware to enter is `ethernet` in nearly all cases, although `token-ring`, which is often found on IBM systems, is also supported.

23.3.2 The openSUSE Version

To improve security, the openSUSE version of the ISC's DHCP server comes with the non-root/chroot patch by Ari Edelkind applied. This enables dhcpd to run with the user ID `nobody` and run in a chroot environment (`/var/lib/dhcp`). To make this possible, the configuration file `dhcpd.conf` must be located in `/var/lib/dhcp/etc`. The init script automatically copies the file to this directory when starting.

Control the server's behavior regarding this feature by means of entries in the file `/etc/sysconfig/dhcpd`. To run dhcpd without the chroot environment, set the variable `DHCPD_RUN_CHROOTED` in `/etc/sysconfig/dhcpd` to “no”.

To enable dhcpd to resolve hostnames even from within the chroot environment, some other configuration files must be copied as well:

- /etc/localtime
- /etc/host.conf
- /etc/hosts
- /etc/resolv.conf

These files are copied to /var/lib/dhcp/etc/ when starting the init script. Take these copies into account for any changes that they require if they are dynamically modified by scripts like /etc/ppp/ip-up. However, there should be no need to worry about this if the configuration file only specifies IP addresses (instead of hostnames).

If your configuration includes additional files that should be copied into the chroot environment, set these under the variable DHCPD_CONF_INCLUDE_FILES in the file /etc/sysconfig/dhcpd. To ensure that the DHCP logging facility keeps working even after a restart of the syslog-ng daemon, there is an additional entry SYSLOGD_ADDITIONAL_SOCKET_DHCP in the file /etc/sysconfig/syslog.

23.4 For More Information

More information about DHCP is available at the Web site of the *Internet Systems Consortium* (http://www.isc.org/products/DHCP/). Information is also available in the dhcpd, dhcpd.conf, dhcpd.leases, and dhcp-options man pages.

Time Synchronization with NTP

24

The NTP (network time protocol) mechanism is a protocol for synchronizing the system time over the network. First, a machine can obtain the time from a server that is a reliable time source. Second, a machine can itself act as a time source for other computers in the network. The goal is twofold—maintaining the absolute time and synchronizing the system time of all machines within a network.

Maintaining an exact system time is important in many situations. The built-in hardware (BIOS) clock does often not meet the requirements of applications like databases. Manual correction of the system time would lead to severe problems because, for example, a backward leap can cause malfunction of critical applications. Within a network, it is usually necessary to synchronize the system time of all machines, but manual time adjustment is a bad approach. xntp provides a mechanism to solve these problems. It continuously adjusts the system time with the help of reliable time servers in the network. It further enables the management of local reference clocks, such as radio-controlled clocks.

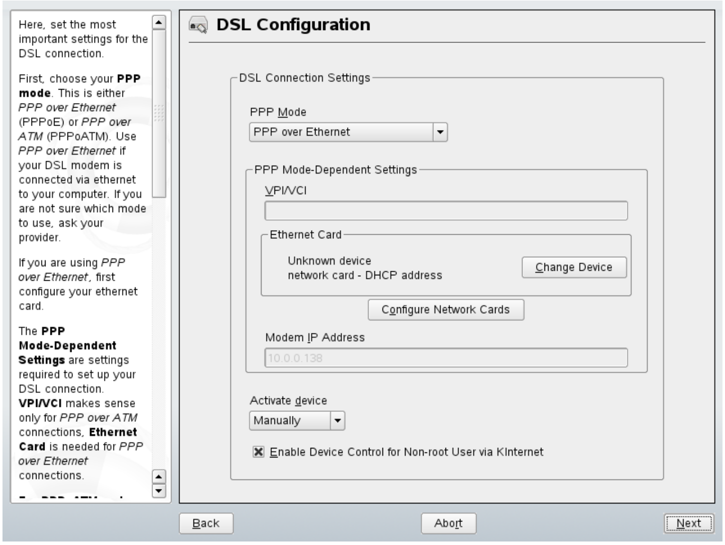
24.1 Configuring an NTP Client with YaST

xntp is preset to use the local computer clock as a time reference. Using the (BIOS) clock, however, only serves as a fallback for the case that no time source of greater precision is available. YaST facilitates the configuration of an NTP client. For a system that is not running a firewall, use either the quick or advanced configuration. For a firewall-protected system, the advanced configuration can open the required ports in SuSEfirewall2.

24.1.1 Quick NTP Client Configuration

The quick NTP client configuration (*Network Services* > *NTP Configuration*) consists of two dialogs. Set the start mode of xntpd and the server to query in the first dialog. To start xntpd automatically when the system is booted, click *During Boot*. Then specify the *NTP Server Configuration*. Either click *Use Random Servers from pool.ntp.org* if you cannot use a local time server or click *Select* to access a second dialog in which to select a suitable time server for your network.

Figure 24.1 *YaST: Configuring an NTP Client*



Here, set the most important settings for the DSL connection.

First, choose your **PPP mode**. This is either *PPP over Ethernet* (PPPoE) or *PPP over ATM* (PPPoATM). Use *PPP over Ethernet* if your DSL modem is connected via ethernet to your computer. If you are not sure which mode to use, ask your provider.

If you are using *PPP over Ethernet*, first configure your ethernet card.

The **PPP Mode-Dependent Settings** are settings required to set up your DSL connection. **VPI/VCI** makes sense only for *PPP over ATM* connections, **Ethernet Card** is needed for *PPP over Ethernet* connections.

DSL Configuration

DSL Connection Settings

PPP Mode: PPP over Ethernet

PPP Mode-Dependent Settings

VPI/VCI:

Ethernet Card

Unknown device
network card - DHCP address

Modem IP Address: 10.0.0.138

Activate device: Manually

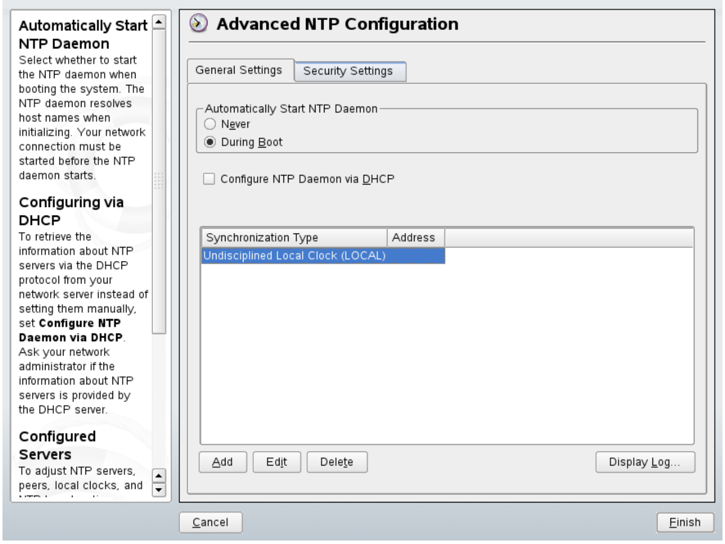
☒ Enable Device Control for Non-root User via KInternet

In the detailed server selection dialog, determine whether to implement time synchronization using a time server from your local network (*Local NTP Server*) or an Internet-based time server that takes care of your time zone (*Public NTP Server*). For a local time server, click *Lookup* to start an SLP query for available time servers in your network. Select the most suitable time server from the list of search results and exit the dialog with *OK*. For a public time server, select your country (time zone) and a suitable server from the list under *Public NTP Server* then exit the dialog with *OK*. In the main dialog, test the availability of the selected server with *Test* and quit the dialog with *Finish*.

24.1.2 Advanced NTP Client Configuration

The advanced configuration of an NTP client can be accessed under *Advanced Configuration* from the main dialog of the *NTP Configuration* module, shown in Figure 24.1, “YaST: Configuring an NTP Client” (page 394), after selecting the start-up mode as described in the quick configuration.

Figure 24.2 *Advanced NTP Client Configuration: General Settings*



Automatically Start NTP Daemon
Select whether to start the NTP daemon when booting the system. The NTP daemon resolves host names when initializing. Your network connection must be started before the NTP daemon starts.

Configuring via DHCP
To retrieve the information about NTP servers via the DHCP protocol from your network server instead of setting them manually, set **Configure NTP Daemon via DHCP**. Ask your network administrator if the information about NTP servers is provided by the DHCP server.

Configured Servers
To adjust NTP servers, peers, local clocks, and

Advanced NTP Configuration

Automatically Start NTP Daemon
○ Never
● During Boot

□ Configure NTP Daemon via DHCP

Synchronization Type	Address
Undisciplined Local Clock (LOCAL)	

On the *General Settings* tab, configure the mode of operation of xntpd. *Configure NTP Daemon via DHCP* sets up the NTP client to get a list of the NTP servers available in your network via DHCP.

The servers and other time sources for the client to query are listed in the lower part. Modify this list as needed with *Add*, *Edit*, and *Delete*. *Display Log* provides the possibility to view the log files of your client.

Click *Add* to add a new source of time information. In the following dialog, select the type of source with which the time synchronization should be made. The following options are available:

Server
: Another dialog enables you to select an NTP server (as described in Section 24.1.1, "Quick NTP Client Configuration" (page 394)). Activate *Use for Initial Synchronization* to trigger the synchronization of the time information between the server and the client when the system is booted. *Options* allows you to specify additional options for xntpd.

  Using *Access Control Options*, you can restrict the actions that the remote computer can perform with the daemon running on your computer. This field is enabled only after checking *Restrict NTP Service to Configured Servers Only* on the *Security Settings* tab. The options correspond to the `restrict` clauses in `/etc/ntp.conf`. For example, `nomodify notrap noquery` disallows the server to modify NTP settings of your computer and to use the trap facility (a remote event logging feature) of your NTP daemon. Using these restrictions is recommended for servers out of your control (e.g., on the Internet).

  Refer to `/usr/share/doc/packages/xntp-doc` (part of the `xntp-doc` package) for detailed information.

Peer
: A peer is a machine to which a symmetric relationship is established: it acts both as a time server and as a client. To use a peer in the same network instead of a server, enter the address of the system. The rest of the dialog is identical to the *Server* dialog.

Radio Clock
: To use a radio clock in your system for the time synchronization, enter the clock type, unit number, device name, and other options in this dialog. Click *Driver Calibration* to fine-tune the driver. Detailed information about the operation of a local radio clock is available in `/usr/share/doc/packages/xntp-doc/refclock.html`.

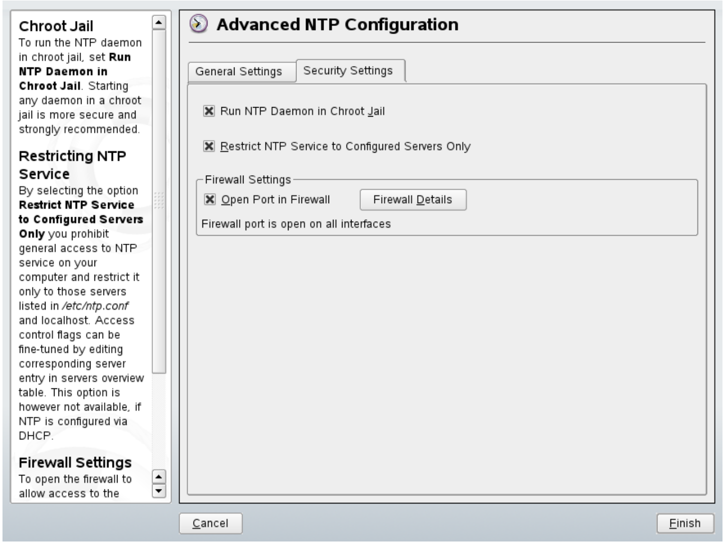
Outgoing Broadcast
: Time information and queries can also be transmitted by broadcast in the network. In this dialog, enter the address to which such broadcasts should be sent. Do not

activate broadcasting unless you have a reliable time source like a radio controlled clock.

Incoming Broadcast

If you want your client to receive its information via broadcast, enter the address from which the respective packets should be accepted in this fields.

Figure 24.3 *Advanced NTP Client Configuration: Security Settings*



Chroot Jail

To run the NTP daemon in chroot jail, set **Run NTP Daemon in Chroot Jail**. Starting any daemon in a chroot jail is more secure and strongly recommended.

Restricting NTP Service

By selecting the option **Restrict NTP Service to Configured Servers Only** you prohibit general access to NTP service on your computer and restrict it only to those servers listed in */etc/ntp.conf* and localhost. Access control flags can be fine-tuned by editing corresponding server entry in servers overview table. This option is however not available, if NTP is configured via DHCP.

Firewall Settings

To open the firewall to allow access to the

Advanced NTP Configuration

☒ Run NTP Daemon in Chroot Jail

☒ Restrict NTP Service to Configured Servers Only

Firewall Settings

☒ Open Port in Firewall

Firewall port is open on all interfaces

On the *Security Settings* tab, determine whether xntpd should be started in a chroot jail. By default, *Run NTP Daemon in Chroot Jail* is activated. This increases the security in the event of an attack over xntpd, because it prevents the attacker from compromising the entire system.

Restrict NTP Service to Configured Servers Only increases the security of your system by disallowing remote computers to view and modify NTP settings of your computer and to use the trap facility for remote event logging. Once enabled, these restrictions apply to all remote computers, unless you override the access control options for individual computers in the list of time sources on the *General Settings* tab. For all other remote computers, only querying for local time is allowed.

Enable *Open Port in Firewall* if SuSEfirewall2 is active, which it is by default. If you leave the port closed, it is not possible to establish a connection to the time server.

24.2 Configuring xntp in the Network

The easiest way to use a time server in the network is to set server parameters. For example, if a time server called ntp.example.com is reachable from the network, add its name to the file /etc/ntp.conf by adding the following line:

```
server ntp.example.com
```

To add more time servers, insert additional lines with the keyword server. After initializing xntpd with the command rcntpd start, it takes about one hour until the time is stabilized and the drift file for correcting the local computer clock is created. With the drift file, the systematic error of the hardware clock can be computed as soon as the computer is powered on. The correction is used immediately, resulting in a higher stability of the system time.

There are two possible ways to use the NTP mechanism as a client: First, the client can query the time from a known server in regular intervals. With many clients, this approach can cause a high load on the server. Second, the client can wait for NTP broadcasts sent out by broadcast time servers in the network. This approach has the disadvantage that the quality of the server is unknown and a server sending out wrong information can cause severe problems.

If the time is obtained via broadcast, you do not need the server name. In this case, enter the line broadcastclient in the configuration file /etc/ntp.conf. To use one or more known time servers exclusively, enter their names in the line starting with servers.

24.3 Setting Up a Local Reference Clock

The software package xntp contains drivers for connecting local reference clocks. A list of supported clocks is available in the xntp-doc package in the file /usr/share/doc/packages/xntp-doc/refclock.html. Every driver is associated with a number. In xntp, the actual configuration takes place by means of pseudo IP addresses. The clocks are entered in the file /etc/ntp.conf as though they existed in the network. For this purpose, they are assigned special IP addresses in the form

`127.127.`*`t`*`.`*`u`*. Here, *`t`* stands for the type of the clock and determines which driver is used and *`u`* for the unit, which determines the interface used.

Normally, the individual drivers have special parameters that describe configuration details. The file `/usr/share/doc/packages/xntp-doc/drivers/driver`*`NN`*`.html` (where *`NN`* is the number of the driver) provides information about the particular type of clock. For example, the "type 8" clock (radio clock over serial interface) requires an additional mode that specifies the clock more precisely. The Conrad DCF77 receiver module, for example, has mode 5. To use this clock as a preferred reference, specify the keyword `prefer`. The complete `server` line for a Conrad DCF77 receiver module would be:

```
server 127.127.8.0 mode 5 prefer
```

Other clocks follow the same pattern. Following the installation of the `xntp-doc` package, the documentation for xntp is available in the directory `/usr/share/doc/packages/xntp-doc`. The file `/usr/share/doc/packages/xntp-doc/refclock.html` provides links to the driver pages describing the driver parameters.

Using NIS 25

As soon as multiple UNIX systems in a network want to access common resources, it becomes important that all user and group identities are the same for all machines in that network. The network should be transparent to users: whatever machines they use, they always find themselves in exactly the same environment. This can be done by means of NIS and NFS services. NFS distributes file systems over a network and is discussed in Chapter 28, *Sharing File Systems with NFS* (page 455).

NIS (Network Information Service) can be described as a database-like service that provides access to the contents of `/etc/passwd`, `/etc/shadow`, and `/etc/group` across networks. NIS can also be used for other purposes (making the contents of files like `/etc/hosts` or `/etc/services` available, for example), but this is beyond the scope of this introduction. People often refer to NIS as *YP*, because it works like the network's “yellow pages.”

25.1 Configuring NIS Servers

To distribute NIS information across networks, you can either have one single server (a *master*) that serves all clients or you can have NIS slave servers requesting this information from the master and relaying it to their respective clients.

- To configure just one NIS server for your network, proceed with Section 25.1.1, “Configuring a NIS Master Server” (page 402).

- If your NIS master server should export its data to slave servers, set up the master server as described in Section 25.1.1, “Configuring a NIS Master Server” (page 402) and set up slave servers in the subnets as described in Section 25.1.2, "Configuring a NIS Slave Server" (page 406).

25.1.1 Configuring a NIS Master Server

To configure a NIS master server for your network, proceed as follows:

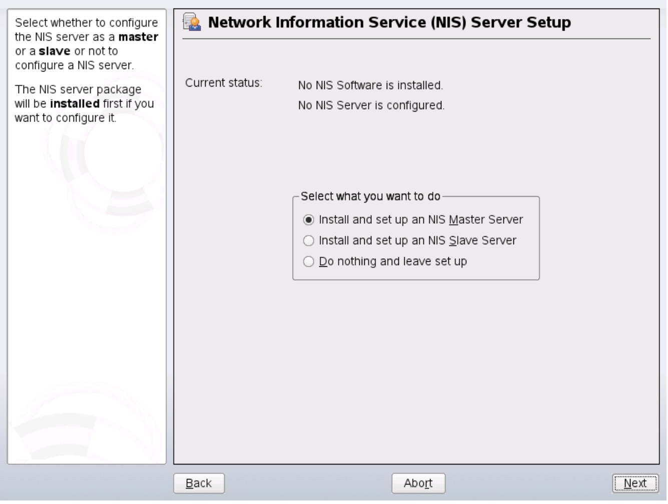
1. Start *YaST* > *Network Services* > *NIS Server*.

2. If you need just one NIS server in your network or if this server is to act as the master for further NIS slave servers, select *Install and set up NIS Master Server*. YaST installs the required packages.

   TIP

   If NIS server software is already installed on your machine, initiate the creation of a NIS master server by clicking *Create NIS Master Server*.

Figure 25.1 *NIS Server Setup*



Select whether to configure the NIS server as a **master** or a **slave** or not to configure a NIS server.

The NIS server package will be **installed** first if you want to configure it.

Network Information Service (NIS) Server Setup

Current status: No NIS Software is installed.
No NIS Server is configured.

Select what you want to do
● Install and set up an NIS Master Server
○ Install and set up an NIS Slave Server
○ Do nothing and leave set up

3. Determine basic NIS setup options:

3a Enter the NIS domain name.

3b Define whether the host should also be a NIS client, enabling users to log in and access data from the NIS server, by selecting *This host is also a NIS client*.

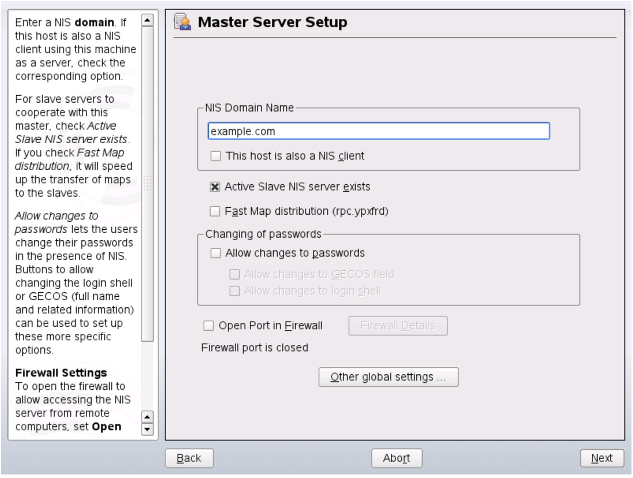
Select *Changing of passwords* to allow users in your network (both local users and those managed through the NIS server) to change their passwords on the NIS server (with the command `yppasswd`).

This makes the options *Allow Changes to GECOS Field* and *Allow Changes to Login Shell* available. "GECOS" means that the users can also change their names and address settings with the command `ypchfn`. "SHELL" allows users to change their default shell with the command `ypchsh`, for example, to switch from bash to sh. The new shell must be one of the predefined entries in `/etc/shells`.

3c If your NIS server should act as a master server to NIS slave servers in other subnets, select *Active Slave NIS Server exists*.

3d Select *Open Ports in Firewall* to have YaST adapt the firewall settings for the NIS server.

Figure 25.2 *Master Server Setup*



Enter a NIS **domain**. If this host is also a NIS client using this machine as a server, check the corresponding option.

For slave servers to cooperate with this master, check *Active Slave NIS server exists*. If you check *Fast Map distribution*, it will speed up the transfer of maps to the slaves.

Allow changes to passwords lets the users change their passwords in the presence of NIS. Buttons to allow changing the login shell or GECOS (full name and related information) can be used to set up these more specific options.

Firewall Settings
To open the firewall to allow accessing the NIS server from remote computers, set **Open**

Master Server Setup

NIS Domain Name: example.com

☐ This host is also a NIS client

☒ Active Slave NIS server exists

☐ Fast Map distribution (rpc.ypxfrd)

Changing of passwords

☐ Allow changes to passwords

☐ Allow changes to GECOS field

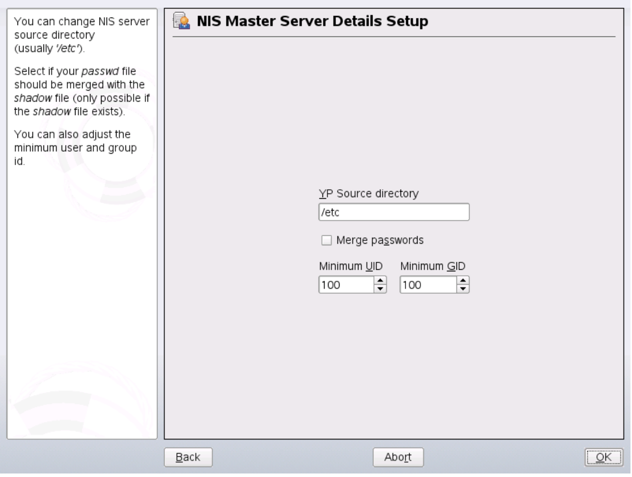
☐ Allow changes to login shell

☐ Open Port in Firewall

Firewall port is closed

3e Leave this dialog with *Next* or click *Other global settings* to make additional settings. *Other global settings* include changing the source directory of the NIS server (`/etc` by default). In addition, passwords can be merged here. The setting should be *Yes* so the files (`/etc/passwd`, `/etc/shadow`, and `/etc/group`) are used to build the user database. Also determine the smallest user and group ID that should be offered by NIS. Click *OK* to confirm your settings and return to the previous screen.

Figure 25.3 *Changing the Directory and Synchronizing Files for a NIS Server*



You can change NIS server source directory (usually '*/etc*').

Select if your *passwd* file should be merged with the *shadow* file (only possible if the *shadow* file exists).

You can also adjust the minimum user and group id.

NIS Master Server Details Setup

YP Source directory: /etc

□ Merge passwords

Minimum UID: 100

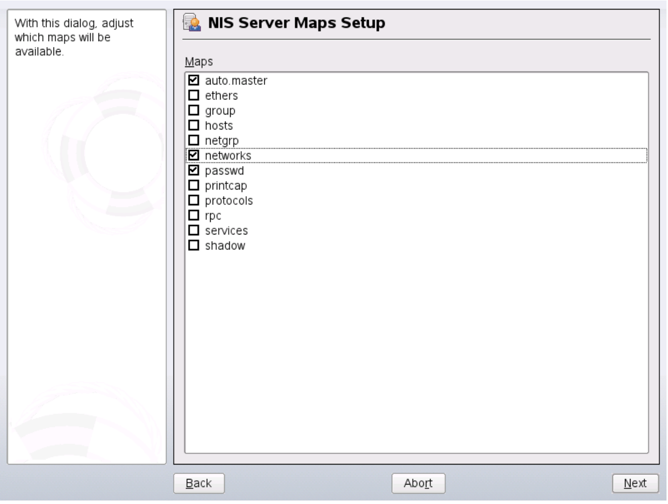
Minimum GID: 100

4 If you previously enabled *Active Slave NIS Server Exists*, enter the hostnames used as slaves and click *Next*.

5 If you do not use slave servers, the slave configuration is skipped and you continue directly to the dialog for the database configuration. Here, specify the *maps*, the partial databases to transfer from the NIS server to the client. The default settings are usually adequate. Leave this dialog with *Next*.

6 Check which maps should be available and click *Next* to continue.

Figure 25.4 *NIS Server Maps Setup*



With this dialog, adjust which maps will be available.

NIS Server Maps Setup

Maps

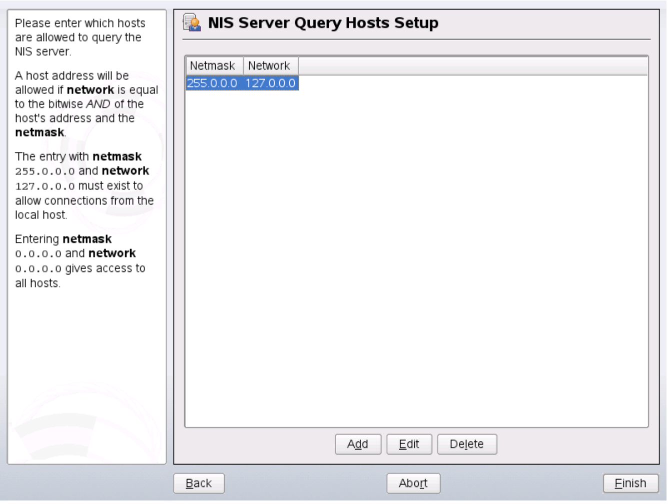
☑ auto.master
□ ethers
□ group
□ hosts
□ netgrp
☑ networks
☑ passwd
□ printcap
□ protocols
□ rpc
□ services
□ shadow

7 Enter the hosts that are allowed to query the NIS server. You can add, edit, or delete hosts by clicking the appropriate button. Specify from which networks requests can be sent to the NIS server. Normally, this is your internal network. In this case, there should be the following two entries:

```
255.0.0.0      127.0.0.0
0.0.0.0        0.0.0.0
```

The first entry enables connections from your own host, which is the NIS server. The second one allows all hosts to send requests to the server.

Figure 25.5 Setting Request Permissions for a NIS Server



Please enter which hosts are allowed to query the NIS server.

A host address will be allowed if **network** is equal to the bitwise *AND* of the host's address and the **netmask**.

The entry with **netmask** `255.0.0.0` and **network** `127.0.0.0` must exist to allow connections from the local host.

Entering **netmask** `0.0.0.0` and **network** `0.0.0.0` gives access to all hosts.

NIS Server Query Hosts Setup

Netmask	Network
255.0.0.0	127.0.0.0

8 Click *Finish* to save changes and exit the setup.

25.1.2 Configuring a NIS Slave Server

To configure additional NIS *slave servers* in your network, proceed as follows:

1 Start *YaST* > *Network Services* > *NIS Server*.

2 Select *Install and set up NIS Slave Server* and click *Next*.

> **TIP**
>
> If NIS server software is already installed on your machine, initiate the creation of a NIS slave server by clicking *Create NIS Slave Server*.

3 Complete the basic setup of your NIS slave server:

3a Enter the NIS domain.

3b Enter hostname or IP address of the master server.

3c Set *This host is also a NIS client* if you want to enable user logins on this server.

3d Adapt the firewall settings with *Open Ports in Firewall*.

3e Click *Next*.

4 Enter the hosts that are allowed to query the NIS server. You can add, edit, or delete hosts by clicking the appropriate button. Specify from which networks requests can be sent to the NIS server. Normally, this is all hosts. In this case, there should be the following two entries:

```
255.0.0.0      127.0.0.0
0.0.0.0        0.0.0.0
```

The first entry enables connections from your own host, which is the NIS server. The second one allows all hosts with access to the same network to send requests to the server.

5 Click *Finish* to save changes and exit the setup.

25.2 Configuring NIS Clients

Use the YaST module *NIS Client* to configure a workstation to use NIS. Select whether the host has a static IP address or receives one issued by DHCP. DHCP can also provide the NIS domain and the NIS server. For information about DHCP, see Chapter 23, *DHCP* (page 375). If a static IP address is used, specify the NIS domain and the NIS server manually. See Figure 25.6, "Setting Domain and Address of a NIS Server" (page 408). *Find* makes YaST search for an active NIS server in your whole network. Depending on the size of your local network, this may be a time-consuming process. *Broadcast* asks for a NIS server in the local network after the specified servers fail to respond.

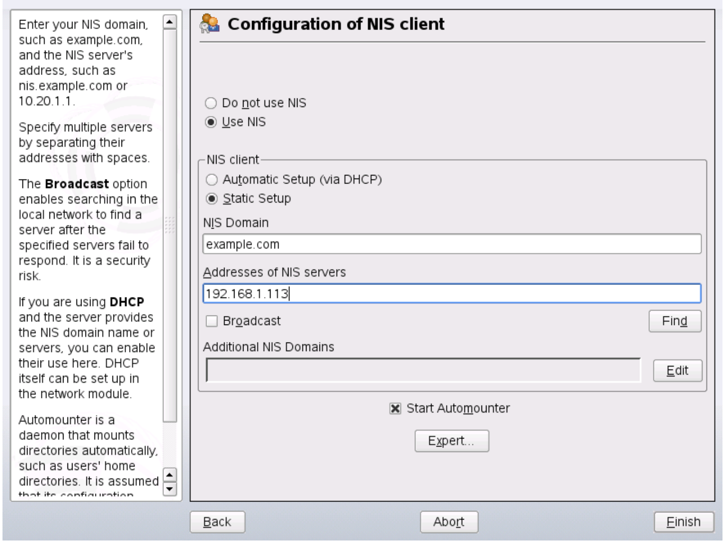
You can also specify multiple servers by entering their addresses in *Addresses of NIS servers* and separating them by spaces.

Depending on your local installation, you may also want to activate the automounter. This option also installs additional software if required.

In the expert settings, disable *Answer Remote Hosts* if you do not want other hosts to be able to query which server your client is using. By checking *Broken Server*, the client is enabled to receive replies from a server communicating through an unprivileged port. For further information, see `man ypbind`.

After you have made your settings, click *Finish* to save them and return to the YaST control center.

Figure 25.6 *Setting Domain and Address of a NIS Server*



Enter your NIS domain, such as example.com, and the NIS server's address, such as nis.example.com or 10.20.1.1.

Specify multiple servers by separating their addresses with spaces.

The **Broadcast** option enables searching in the local network to find a server after the specified servers fail to respond. It is a security risk.

If you are using **DHCP** and the server provides the NIS domain name or servers, you can enable their use here. DHCP itself can be set up in the network module.

Automounter is a daemon that mounts directories automatically, such as users' home directories. It is assumed that its configuration

Configuration of NIS client

○ Do not use NIS
● Use NIS

NIS client
○ Automatic Setup (via DHCP)
● Static Setup

NIS Domain: example.com

Addresses of NIS servers: 192.168.1.113

□ Broadcast

Additional NIS Domains:

☒ Start Automounter

LDAP—A Directory Service 26

The Lightweight Directory Access Protocol (LDAP) is a set of protocols designed to access and maintain information directories. LDAP can be used for numerous purposes, such as user and group management, system configuration management, or address management. This chapter provides a basic understanding of how OpenLDAP works and how to manage LDAP data with YaST. While there are several implementations of the LDAP protocol, this chapter focuses entirely on the OpenLDAP implementation.

It is crucial within a networked environment to keep important information structured and quickly available. This can be done with a directory service that, like the common yellow pages, keeps information available in a well-structured, quickly searchable form.

In the ideal case, a central server keeps the data in a directory and distributes it to all clients using a certain protocol. The data is structured in a way that allows a wide range of applications to access it. That way, it is not necessary for every single calendar tool and e-mail client to keep its own database—a central repository can be accessed instead. This notably reduces the administration effort for the information. The use of an open and standardized protocol like LDAP ensures that as many different client applications as possible can access such information.

A directory in this context is a type of database optimized for quick and effective reading and searching:

- To make numerous concurrent reading accesses possible, write access is limited to a small number of updates by the administrator. Conventional databases are optimized for accepting the largest possible data volume in a short time.

- Because write accesses can only be executed in a restricted fashion, a directory service is used to administer mostly unchanging, static information. Data in a conventional database typically changes very often (*dynamic* data). Phone numbers in a company directory do not change nearly as often as, for example, the figures administered in accounting.

- When static data is administered, updates of the existing data sets are very rare. When working with dynamic data, especially when data sets like bank accounts or accounting are concerned, the consistency of the data is of primary importance. If an amount should be subtracted from one place to be added to another, both operations must happen concurrently, within a *transaction*, to ensure balance over the data stock. Databases support such transactions. Directories do not. Short-term inconsistencies of the data are quite acceptable in directories.

The design of a directory service like LDAP is not laid out to support complex update or query mechanisms. All applications accessing this service should gain access quickly and easily.

26.1 LDAP versus NIS

The Unix system administrator traditionally uses the NIS service for name resolution and data distribution in a network. The configuration data contained in the files in `/etc` and the directories `group`, `hosts`, `mail`, `netgroup`, `networks`, `passwd`, `printcap`, `protocols`, `rpc`, and `services` are distributed by clients all over the network. These files can be maintained without major effort because they are simple text files. The handling of larger amounts of data, however, becomes increasingly difficult due to nonexistent structuring. NIS is only designed for Unix platforms. This means it is not suitable as a centralized data administration tool in heterogeneous networks.

Unlike NIS, the LDAP service is not restricted to pure Unix networks. Windows servers (from 2000) support LDAP as a directory service. Application tasks mentioned above are additionally supported in non-Unix systems.

The LDAP principle can be applied to any data structure that should be centrally administered. A few application examples are:

- Employment as a replacement for the NIS service

- Mail routing (postfix, sendmail)
- Address books for mail clients, like Mozilla, Evolution, and Outlook
- Administration of zone descriptions for a BIND9 name server
- User authentication with Samba in heterogeneous networks

This list can be extended because LDAP is extensible, unlike NIS. The clearly-defined hierarchical structure of the data eases the administration of large amounts of data, because it can be searched more easily.

26.2 Structure of an LDAP Directory Tree

To get a deep background knowledge on how a LDAP server works and how the data are stored, it is vital to understand the way the data are organized on the server and how this structure enables LDAP to provide fast access to the data you need. To successfully operate an LDAP setup, you also need to be familiar with some basic LDAP terminology. This section introduces the basic layout of an LDAP directory tree and provides the basic terminology used in an LDAP context. Skip this introductory section, if you already have some LDAP background knowledge and just want to learn how to set up an LDAP environment in openSUSE. Read on at Section 26.3, “Configuring an LDAP Server with YaST” (page 414) or Section 26.7, “Manually Configuring an LDAP Server” (page 429), respectively.

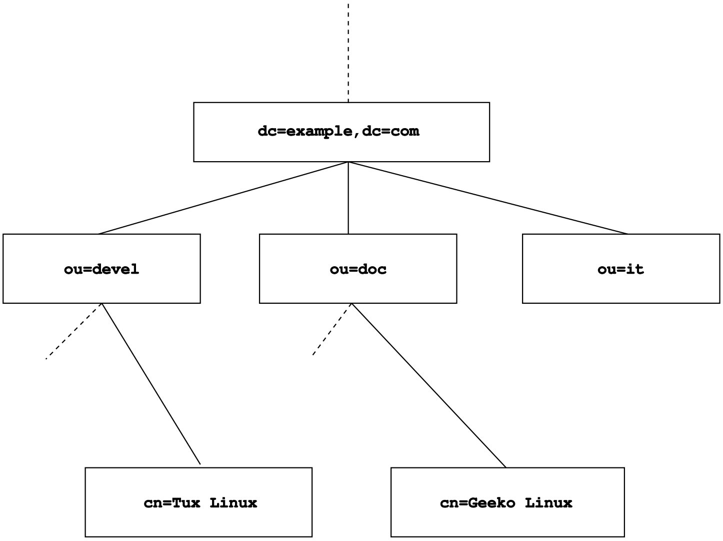
An LDAP directory has a tree structure. All entries (called objects) of the directory have a defined position within this hierarchy. This hierarchy is called the *directory information tree* (DIT). The complete path to the desired entry, which unambiguously identifies it, is called *distinguished name* or DN. A single node along the path to this entry is called *relative distinguished name* or RDN. Objects can generally be assigned to one of two possible types:

container
: These objects can themselves contain other objects. Such object classes are `root` (the root element of the directory tree, which does not really exist), `c` (country), `ou` (organizational unit), and `dc` (domain component). This model is comparable to the directories (folders) in a file system.

leaf
: These objects sit at the end of a branch and have no subordinate objects. Examples are person, InetOrgPerson, or groupofNames.

The top of the directory hierarchy has a root element root. This can contain c (country), dc (domain component), or o (organization) as subordinate elements. The relations within an LDAP directory tree become more evident in the following example, shown in Figure 26.1, "Structure of an LDAP Directory" (page 412).

Figure 26.1 *Structure of an LDAP Directory*



The complete diagram is a fictional directory information tree. The entries on three levels are depicted. Each entry corresponds to one box in the picture. The complete, valid *distinguished name* for the fictional employee Geeko Linux, in this case, is cn=Geeko Linux,ou=doc,dc=example,dc=com. It is composed by adding the RDN cn=Geeko Linux to the DN of the preceding entry ou=doc,dc=example,dc=com.

The types of objects that should be stored in the DIT are globally determined following a *scheme*. The type of an object is determined by the *object class*. The object class determines what attributes the concerned object must or can be assigned. A scheme, therefore, must contain definitions of all object classes and attributes used in the desired application scenario. There are a few common schemes (see RFC 2252 and 2256). It is, however, possible to create custom schemes or to use multiple schemes complementing each other if this is required by the environment in which the LDAP server should operate.

Table 26.1, "Commonly Used Object Classes and Attributes" (page 413) offers a small overview of the object classes from `core.schema` and `inetorgperson.schema` used in the example, including required attributes and valid attribute values.

***Table 26.1** Commonly Used Object Classes and Attributes*

Object Class	Meaning	Example Entry	Required Attributes
dcObject	*domainComponent* (name components of the domain)	example	dc
organizationalUnit	*organizationalUnit* (organizational unit)	doc	ou
inetOrgPerson	*inetOrgPerson* (person-related data for the intranet or Internet)	Geeko Linux	sn and cn

Example 26.1, "Excerpt from schema.core " (page 413) shows an excerpt from a scheme directive with explanations (line numbering for explanatory reasons).

***Example 26.1** Excerpt from schema.core*

```
#1 attributetype (2.5.4.11 NAME ( 'ou' 'organizationalUnitName')
#2        DESC 'RFC2256: organizational unit this object belongs to'
#3        SUP name )

...
#4 objectclass ( 2.5.6.5 NAME 'organizationalUnit'
#5        DESC 'RFC2256: an organizational unit'
#6        SUP top STRUCTURAL
#7        MUST ou
#8 MAY (userPassword $ searchGuide $ seeAlso $ businessCategory
    $ x121Address $ registeredAddress $ destinationIndicator
    $ preferredDeliveryMethod $ telexNumber
    $ teletexTerminalIdentifier $ telephoneNumber
    $ internationaliSDNNumber $ facsimileTelephoneNumber
    $ street $ postOfficeBox $ postalCode $ postalAddress
    $ physicalDeliveryOfficeName
    $ st $ l $ description) )
    ...
```

The attribute type organizationalUnitName and the corresponding object class organizationalUnit serve as an example here. Line 1 features the name of the attribute, its unique OID (*object identifier*) (numerical), and the abbreviation of the attribute.

Line 2 gives a brief description of the attribute with DESC. The corresponding RFC on which the definition is based is also mentioned here. SUP in line 3 indicates a superordinate attribute type to which this attribute belongs.

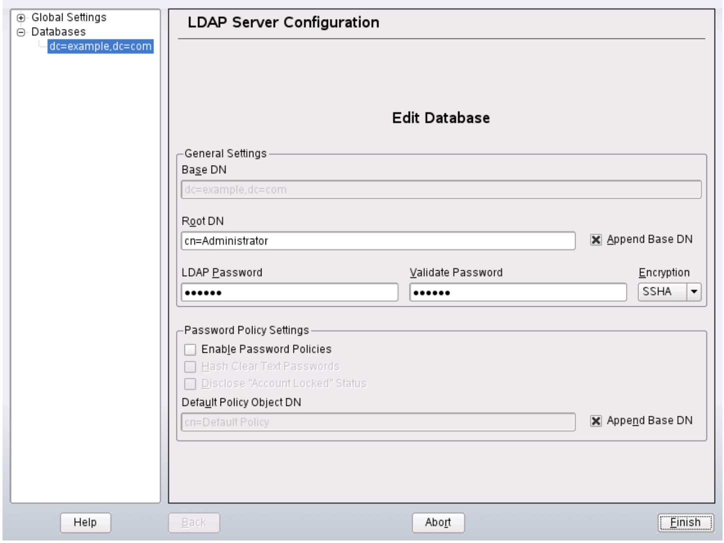
The definition of the object class organizationalUnit begins in line 4, like in the definition of the attribute, with an OID and the name of the object class. Line 5 features a brief description of the object class. Line 6, with its entry SUP top, indicates that this object class is not subordinate to another object class. Line 7, starting with MUST, lists all attribute types that must be used in conjunction with an object of the type organizationalUnit. Line 8, starting with MAY, lists all attribute types that are permitted in conjunction with this object class.

A very good introduction to the use of schemes can be found in the documentation of OpenLDAP. When installed, find it in /usr/share/doc/packages/openldap2/admin-guide/index.html.

26.3 Configuring an LDAP Server with YaST

Use YaST to set up an LDAP server. Typical use cases for LDAP servers include the management of user account data and the configuration of mail, DNS, and DHCP servers.

Figure 26.2 *YaST LDAP Server Configuration*



⊕ Global Settings
⊖ Databases
dc=example,dc=com

LDAP Server Configuration

Edit Database

General Settings

Base DN: dc=example,dc=com

Root DN: cn=Administrator ☒ Append Base DN

LDAP Password: •••••• Validate Password: •••••• Encryption: SSHA

Password Policy Settings

☐ Enable Password Policies
☐ Hash Clear Text Passwords
☐ Disclose "Account Locked" Status

Default Policy Object DN: cn=Default Policy ☒ Append Base DN

To set up an LDAP server for user account data, proceed as follows:

1. Log in as `root`.

2. Start YaST and select *Network Services* > *LDAP Server*.

3. Set LDAP to be started at system boot.

4. If the LDAP server should announce its services via SLP, check *Register at an SLP Daemon*.

5. Select *Configure* to configure *General Settings* and *Databases*.

To configure the *Global Settings* of your LDAP server, proceed as follows:

1. Accept or modify the schema files included in the server's configuration by selecting *Schema Files* in the left part of the dialog. The default selection of schema files applies to the server providing a source of YaST user account data.

2 With *Log Level Settings*, configure the degree of logging activity (verbosity) of the LDAP server. From the predefined list, select or deselect the logging options according to your needs. The more options are enabled, the larger your log files grow.

3 Determine the connection types the LDAP server should allow. Choose from:

bind_v2
: This option enables connection requests (bind requests) from clients using the previous version of the protocol (LDAPv2).

bind_anon_cred
: Normally the LDAP server denies any authentication attempts with empty credentials (DN or password). Enabling this option, however, makes it possible to connect with a password and no DN to establish an anonymous connection.

bind_anon_dn
: Enabling this option makes it possible to connect without authentication (anonymously) using a DN but no password.

update_anon
: Enabling this option allows non-authenticated (anonymous) update operations. Access is restricted according to ACLs and other rules (see Section 26.7.1, "Global Directives in slapd.conf" (page 430)).

4 To configure secure communication between client and server, proceed with *TLS Settings*:

4a Set *TLS Active* to *Yes* to enable TLS and SSL encryption of the client/server communication.

4b Click *Select Certificate* and determine how to obtain a valid certificate. Choose *Import Certificate* (import certificate from external source) or *Use Common Server Certificate* (use the certificate created during installation).

- If you opted for importing a certificate, YaST prompts you to specify the exact path to its location.

- If you opted for using the common server certificate and it has not been created during installation, it is subsequently created.

To configure the databases managed by your LDAP server, proceed as follows:

1. Select the *Databases* item in the left part of the dialog.

2. Click *Add Database* to add the new database.

3. Enter the requested data:

   Base DN
   Enter the base DN of your LDAP server.

   Root DN
   Enter the DN of the administrator in charge of the server. If you check *Append Base DN*, only provide the `cn` of the administrator and the system fills in the rest automatically.

   LDAP Password
   Enter the password for the database administrator.

   Encryption
   Determine the encryption algorithm to use to secure the password of Root DN. Choose *crypt*, *smd5*, *ssha*, or *sha*. The dialog also includes a *plain* option to enable the use of plain text passwords, but enabling this is not recommended for security reasons. To confirm your settings and return to the previous dialog, select *OK*.

4. Enable enforcement of password policies to provide extra security to your LDAP server:

   4a Select *Password Policy Settings* to be able to specify a password policy.

   4b Activate *Hash Clear Text Passwords* to have clear text passwords be hashed before they are written to the database whenever they are added or modified.

   4c *Disclose Account Locked Status* provides a meaningful error message to bind requests to locked accounts.

WARNING: Locked Accounts in Security Sensitive Environments

Do not use the *Disclose Account Locked Status* option if your environment is sensitive to security issues, because the “Locked Account” error message provides security sensitive information that can be exploited by a potential attacker.

4d Enter the DN of the default policy object. To use a DN other than the one suggested by YaST, enter your choice. Otherwise accept the default setting.

5 Complete the database configuration by clicking *Finish*.

If you have not opted for password policies, your server is ready to run at this point. If you chose to enable password policies, proceed with the configuration of the password policy in detail. If you chose a password policy object that does not yet exist, YaST creates one:

1 Enter the LDAP server password.

2 Configure the password change policies:

2a Determine the number of passwords stored in the password history. Saved passwords may not be reused by the user.

2b Determine whether users can change their password and whether they need to change their password after a reset by the administrator. Optionally require the old password for password changes.

2c Determine whether and to what extent passwords should be subject to quality checking. Set a minimum password length that must be met before a password is valid. If you select *Accept Uncheckable Passwords*, users are allowed to use encrypted passwords although the quality checks cannot be performed. If you opt for *Only Accept Checked Passwords* only those passwords that pass the quality tests are accepted as valid.

3 Configure the password aging policies:

3a Determine the minimum password age (the time that needs to pass between two valid password changes) and the maximum password age.

3b Determine the time between a password expiration warning and the actual password expiration.

3c Set the number of postponement uses of an expired password before the password expires entirely.

4 Configure the lockout policies:

4a Enable password locking.

4b Determine the number of bind failures that trigger a password lock.

4c Determine the duration of the password lock.

4d Determine for how long password failures are kept in the cache before they are purged.

5 Apply your password policy settings with *Accept*.

To edit a previously created database, select its base DN in the tree to the left. In the right part of the window, YaST displays a dialog similar to the one used for the creation of a new database—with the main difference that the base DN entry is grayed out and cannot be changed.

After leaving the LDAP server configuration by selecting *Finish*, you are ready to go with a basic working configuration for your LDAP server. To fine-tune this setup, edit the file `/etc/openldap/slapd.conf` accordingly then restart the server.

26.4 Configuring an LDAP Client with YaST

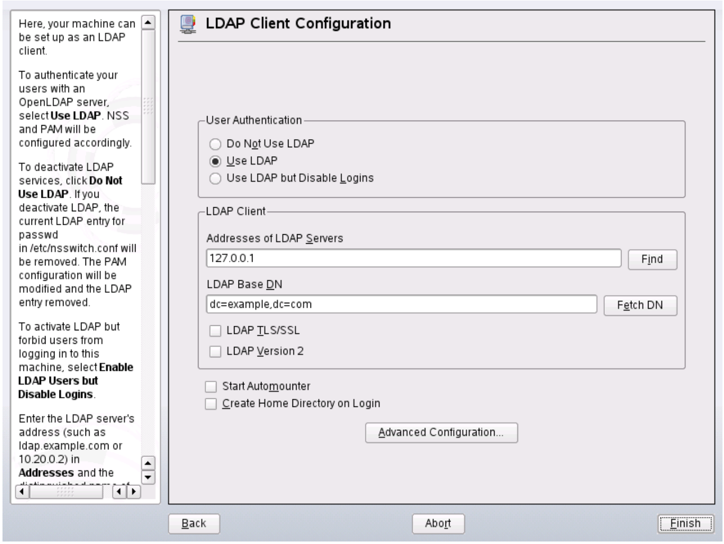
YaST includes a module to set up LDAP-based user management. If you did not enable this feature during the installation, start the module by selecting *Network Services* > *LDAP Client*. YaST automatically enables any PAM and NSS related changes as required by LDAP and installs the necessary files. Simply connect your client to the server and let YaST manage users over LDAP. This basic setup is described in Section 26.4.1, "Configuring Basic Settings" (page 420).

Use the YaST LDAP client to further configure the YaST group and user configuration modules. This includes manipulating the default settings for new users and groups and the number and nature of the attributes assigned to a user or a group. LDAP user management allows you to assign far more and different attributes to users and groups than traditional user or group management solutions. This is described in Section 26.4.2, "Configuring the YaST Group and User Administration Modules" (page 423).

26.4.1 Configuring Basic Settings

The basic LDAP client configuration dialog (Figure 26.3, "YaST: Configuration of the LDAP Client" (page 420)) opens during installation if you choose LDAP user management or when you select *Network Services* > *LDAP Client* in the YaST Control Center in the installed system.

Figure 26.3 *YaST: Configuration of the LDAP Client*



Here, your machine can be set up as an LDAP client.

To authenticate your users with an OpenLDAP server, select **Use LDAP**. NSS and PAM will be configured accordingly.

To deactivate LDAP services, click **Do Not Use LDAP**. If you deactivate LDAP, the current LDAP entry for passwd in /etc/nsswitch.conf will be removed. The PAM configuration will be modified and the LDAP entry removed.

To activate LDAP but forbid users from logging in to this machine, select **Enable LDAP Users but Disable Logins**.

Enter the LDAP server's address (such as ldap.example.com or 10.20.0.2) in **Addresses** and the

LDAP Client Configuration

User Authentication

○ Do Not Use LDAP
● Use LDAP
○ Use LDAP but Disable Logins

LDAP Client

Addresses of LDAP Servers: 127.0.0.1

LDAP Base DN: dc=example,dc=com

□ LDAP TLS/SSL
□ LDAP Version 2

□ Start Automounter
□ Create Home Directory on Login

To authenticate users of your machine against an OpenLDAP server and enable user management via OpenLDAP, proceed as follows:

1. Click *Use LDAP* to enable the use of LDAP. Select *Use LDAP but Disable Logins* instead if you want to use LDAP for authentication, but do not want other users to log in to this client.

2. Enter the IP address of the LDAP server to use.

3. Enter the *LDAP base DN* to select the search base on the LDAP server. To retrieve the base DN automatically, click *Fetch DN*. YaST then checks for any LDAP database on the server address specified above. Choose the appropriate base DN from the search results given by YaST.

4. If TLS or SSL protected communication with the server is required, select *LDAP TLS/SSL*.

5. If the LDAP server still uses LDAPv2, explicitly enable the use of this protocol version by selecting *LDAP Version 2*.

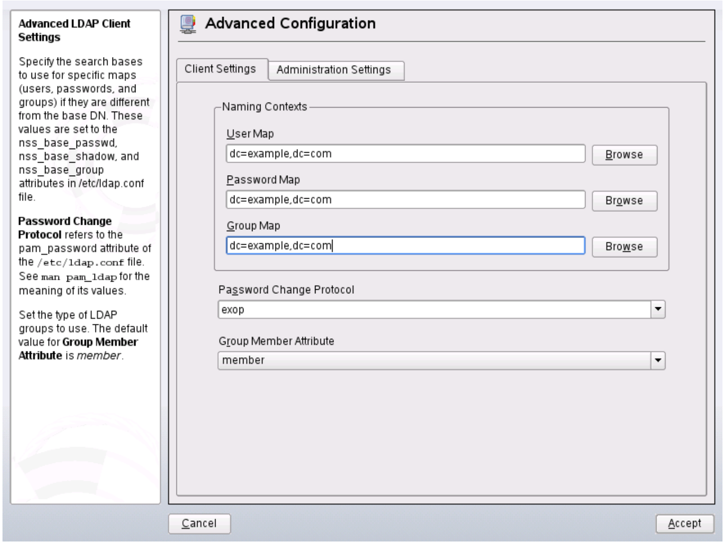
6. Select *Start Automounter* to mount remote directories on your client, such as a remotely managed `/home`.

7. Select *Create Home Directory on Login* to have a user's home automatically created on the first user login.

8. Click *Finish* to apply your settings.

To modify data on the server as administrator, click *Advanced Configuration*. The following dialog is split in two tabs. See Figure 26.4, “YaST: Advanced Configuration” (page 422).

Figure 26.4 YaST: Advanced Configuration



Advanced LDAP Client Settings

Specify the search bases to use for specific maps (users, passwords, and groups) if they are different from the base DN. These values are set to the nss_base_passwd, nss_base_shadow, and nss_base_group attributes in /etc/ldap.conf file.

Password Change Protocol refers to the pam_password attribute of the `/etc/ldap.conf` file. See `man pam_ldap` for the meaning of its values.

Set the type of LDAP groups to use. The default value for **Group Member Attribute** is *member*.

Advanced Configuration

Naming Contexts

User Map: dc=example,dc=com

Password Map: dc=example,dc=com

Group Map: dc=example,dc=com

Password Change Protocol: exop

Group Member Attribute: member

1 In the *Client Settings* tab, adjust the following settings to your needs:

1a If the search base for users, passwords, and groups differs from the global search base specified the *LDAP base DN*, enter these different naming contexts in *User Map*, *Password Map*, and *Group Map*.

1b Specify the password change protocol. The standard method to use whenever a password is changed is `crypt`, meaning that password hashes generated by `crypt` are used. For details on this and other options, refer to the `pam_ldap` man page.

1c Specify the LDAP group to use with *Group Member Attribute*. The default value for this is `member`.

2 In *Administration Settings*, adjust the following settings:

2a Set the base for storing your user management data via *Configuration Base DN*.

2b Enter the appropriate value for *Administrator DN*. This DN must be identical with the `rootdn` value specified in `/etc/openldap/slapd.conf` to enable this particular user to manipulate data stored on the LDAP server. Enter the full DN (such as `cn=Administrator,dc=example,dc=com`) or activate *Append Base DN* to have the base DN added automatically when you enter `cn=Administrator`.

2c Check *Create Default Configuration Objects* to create the basic configuration objects on the server to enable user management via LDAP.

2d If your client machine should act as a file server for home directories across your network, check *Home Directories on This Machine*.

2e Use the *Password Policy* section to select, add, delete, or modify the password policy settings to use. The configuration of password policies with YaST is part of the LDAP server setup.

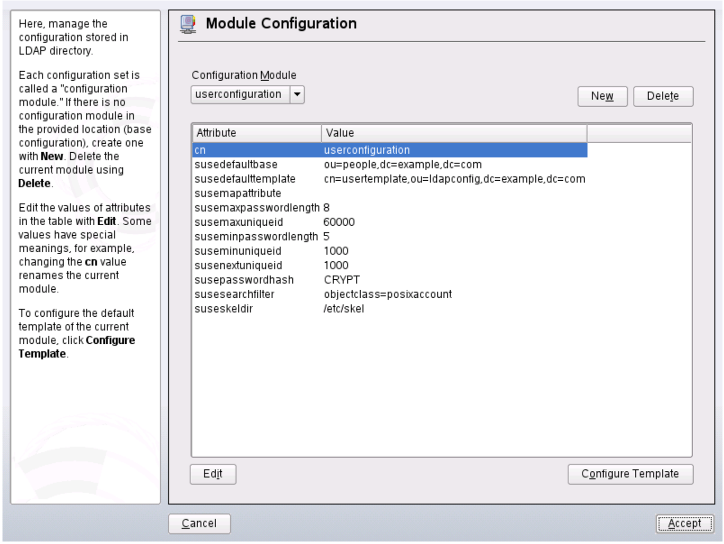
2f Click *Accept* to leave the *Advanced Configuration* then *Finish* to apply your settings.

Use *Configure User Management Settings* to edit entries on the LDAP server. Access to the configuration modules on the server is then granted according to the ACLs and ACIs stored on the server. Follow the procedures outlined in Section 26.4.2, “Configuring the YaST Group and User Administration Modules” (page 423).

26.4.2 Configuring the YaST Group and User Administration Modules

Use the YaST LDAP client to adapt the YaST modules for user and group administration and to extend them as needed. Define templates with default values for the individual attributes to simplify the data registration. The presets created here are stored as LDAP objects in the LDAP directory. The registration of user data is still done with the regular YaST modules for user and group management. The registered data is stored as LDAP objects on the server.

***Figure 26.5** YaST: Module Configuration*



Here, manage the configuration stored in LDAP directory.

Each configuration set is called a "configuration module." If there is no configuration module in the provided location (base configuration), create one with **New**. Delete the current module using **Delete**.

Edit the values of attributes in the table with **Edit**. Some values have special meanings, for example, changing the **cn** value renames the current module.

To configure the default template of the current module, click **Configure Template**.

Module Configuration

Configuration Module: userconfiguration

Attribute	Value
cn	userconfiguration
susedefaultbase	ou=people,dc=example,dc=com
susedefaulttemplate	cn=usertemplate,ou=ldapconfig,dc=example,dc=com
susemapattribute	
susemaxpasswordlength	8
susemaxuniqueid	60000
suseminpasswordlength	5
suseminuniqueid	1000
susenextuniqueid	1000
susepasswordhash	CRYPT
susesearchfilter	objectclass=posixaccount
suseskeldir	/etc/skel

The dialog for module configuration (Figure 26.5, "YaST: Module Configuration" (page 424)) allows the creation of new modules, selection and modification of existing configuration modules, and design and modification of templates for such modules.

To create a new configuration module, proceed as follows:

1 Click *New* and select the type of module to create. For a user configuration module, select `suseuserconfiguration` and for a group configuration choose `susegroupconfiguration`.

2 Choose a name for the new template. The content view then features a table listing all attributes allowed in this module with their assigned values. Apart from all set attributes, the list also contains all other attributes allowed by the current schema but currently not used.

3 Accept the preset values or adjust the defaults to use in group and user configuration by selecting the respective attribute, pressing *Edit*, and entering the new value. Rename a module by simply changing the `cn` attribute of the module. Clicking *Delete* deletes the currently selected module.

4 After you click *Accept*, the new module is added to the selection menu.

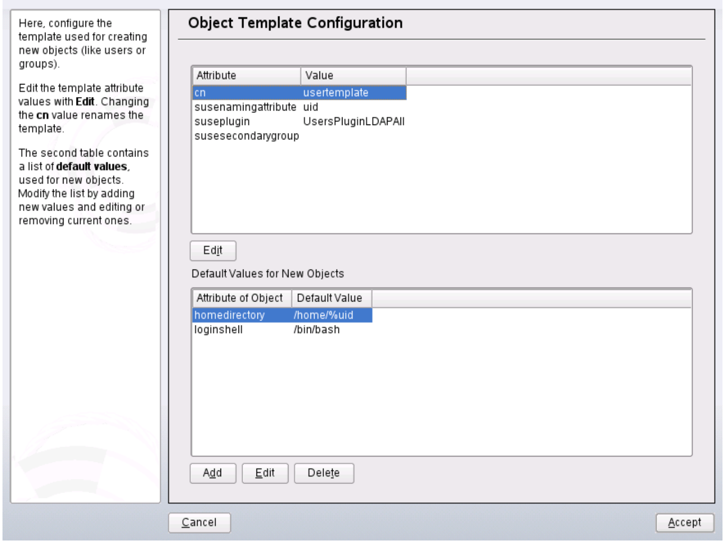
The YaST modules for group and user administration embed templates with sensible standard values. To edit a template associated with a configuration module, proceed as follows:

1. In the *Module Configuration* dialog, click *Configure Template*.

2. Determine the values of the general attributes assigned to this template according to your needs or leave some of them empty. Empty attributes are deleted on the LDAP server.

3. Modify, delete, or add new default values for new objects (user or group configuration objects in the LDAP tree).

Figure 26.6 *YaST: Configuration of an Object Template*



Here, configure the template used for creating new objects (like users or groups).

Edit the template attribute values with **Edit**. Changing the **cn** value renames the template.

The second table contains a list of **default values**, used for new objects. Modify the list by adding new values and editing or removing current ones.

Object Template Configuration

Attribute	Value
cn	usertemplate
susenamingattribute	uid
suseplugin	UsersPluginLDAPAll
susesecondarygroup	

Default Values for New Objects

Attribute of Object	Default Value
homedirectory	/home/%uid
loginshell	/bin/bash

Connect the template to its module by setting the `susedefaulttemplate` attribute value of the module to the DN of the adapted template.

TIP

The default values for an attribute can be created from other attributes by using a variable instead of an absolute value. For example, when creating a new user, `cn=%sn %givenName` is created automatically from the attribute values for `sn` and `givenName`.

Once all modules and templates are configured correctly and ready to run, new groups and users can be registered in the usual way with YaST.

26.5 Configuring LDAP Users and Groups in YaST

The actual registration of user and group data differs only slightly from the procedure when not using LDAP. The following brief instructions relate to the administration of users. The procedure for administering groups is analogous.

1 Access the YaST user administration with *Security & Users* > *User Administration*.

2 Use *Set Filter* to limit the view of users to the LDAP users and enter the password for Root DN.

3 Click *Add* and enter the configuration of a new user. A dialog with four tabs opens:

3a Specify username, login, and password in the *User Data* tab.

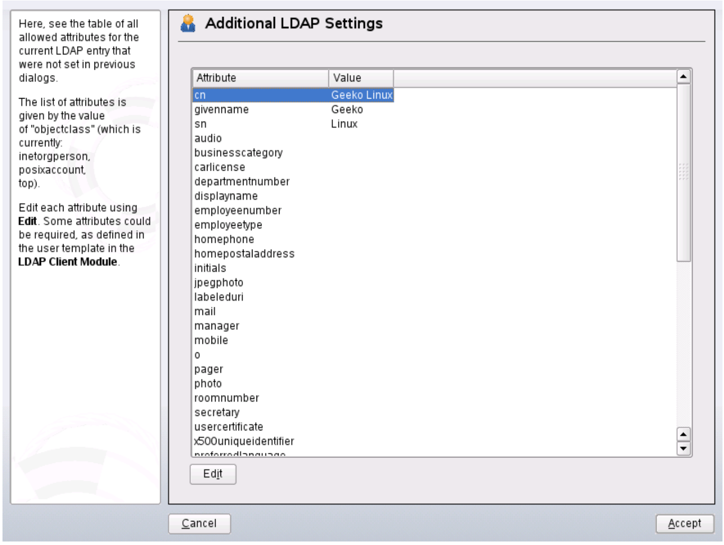
3b Check the *Details* tab for the group membership, login shell, and home directory of the new user. If necessary, change the default to values that better suit your needs. The default values as well as those of the password settings can be defined with the procedure described in Section 26.4.2, “Configuring the YaST Group and User Administration Modules” (page 423).

3c Modify or accept the default *Password Settings*.

3d Enter the *Plug-Ins* tab, select the LDAP plug-in, and click *Launch* to configure additional LDAP attributes assigned to the new user (see Figure 26.7, “YaST: Additional LDAP Settings” (page 427)).

4 Click *Accept* to apply your settings and leave the user configuration.

Figure 26.7 *YaST: Additional LDAP Settings*



Here, see the table of all allowed attributes for the current LDAP entry that were not set in previous dialogs.

The list of attributes is given by the value of "objectclass" (which is currently: inetorgperson, posixaccount, top).

Edit each attribute using **Edit**. Some attributes could be required, as defined in the user template in the **LDAP Client Module**.

Additional LDAP Settings

Attribute	Value
cn	Geeko Linux
givenname	Geeko
sn	Linux
audio	
businesscategory	
carlicense	
departmentnumber	
displayname	
employeenumber	
employeetype	
homephone	
homepostaladdress	
initials	
jpegphoto	
labeleduri	
mail	
manager	
mobile	
o	
pager	
photo	
roomnumber	
secretary	
usercertificate	
x500uniqueidentifier	
preferredlanguage	

The initial input form of user administration offers *LDAP Options*. This gives the possibility to apply LDAP search filters to the set of available users or go to the module for the configuration of LDAP users and groups by selecting *LDAP User and Group Configuration*.

26.6 Browsing the LDAP Directory Tree

To browse the LDAP directory tree and all its entries conveniently, use the YaST LDAP Browser:

1. Log in as `root`.

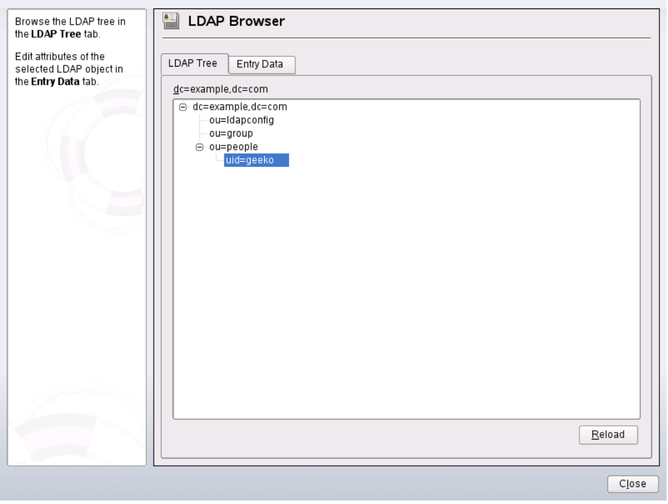
2. Start *YaST* > *Network Services* > *LDAP Browser*.

3. Enter the address of the LDAP server, the AdministratorDN, and the password for the RootDN of this server if you need both to read and write the data stored on the server.

   Alternatively, choose *Anonymous Access* and do not provide the password to gain read access to the directory.

   The *LDAP Tree* tab displays the content of the LDAP directory to which your machine connected. Click items to unfold their subitems.

Figure 26.8 *Browsing the LDAP Directory Tree*



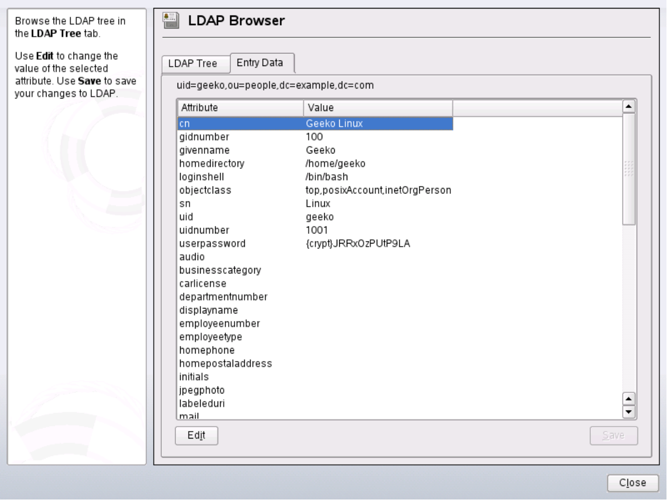
Browse the LDAP tree in the **LDAP Tree** tab.

Edit attributes of the selected LDAP object in the **Entry Data** tab.

4 To view any of the entries in detail, select it in the *LDAP Tree* view and open the *Entry Data* tab.

All attributes and values associated with this entry are displayed.

Figure 26.9 *Browsing the Entry Data*



Browse the LDAP tree in the **LDAP Tree** tab.

Use **Edit** to change the value of the selected attribute. Use **Save** to save your changes to LDAP.

LDAP Browser

uid=geeko,ou=people,dc=example,dc=com

Attribute	Value
cn	Geeko Linux
gidnumber	100
givenname	Geeko
homedirectory	/home/geeko
loginshell	/bin/bash
objectclass	top,posixAccount,inetOrgPerson
sn	Linux
uid	geeko
uidnumber	1001
userpassword	{crypt}JRRxOzPUtP9LA
audio	
businesscategory	
carlicense	
departmentnumber	
displayname	
employeenumber	
employeetype	
homephone	
homepostaladdress	
initials	
jpegphoto	
labeleduri	
mail	

5 To change the value of any of these attributes, select the attribute, click *Edit*, enter the new value, click *Save*, and provide the RootDN password when prompted.

6 Leave the LDAP browser with *Close*.

26.7 Manually Configuring an LDAP Server

Your installed system contains a complete configuration file for your LDAP server at `/etc/openldap/slapd.conf`. The single entries are briefly described here and necessary adjustments are explained. Entries prefixed with a hash (#) are inactive. This comment character must be removed to activate them.

26.7.1 Global Directives in slapd.conf

Example 26.2 *slapd.conf: Include Directive for Schemes*

```
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/rfc2307bis.schema
include         /etc/openldap/schema/yast.schema
```

This first directive in `slapd.conf`, shown in Example 26.2, "slapd.conf: Include Directive for Schemes" (page 430), specifies the scheme by which the LDAP directory is organized. The entry `core.schema` is required. Additionally required schemes are appended to this directive. Find information in the included OpenLDAP documentation.

Example 26.3 *slapd.conf: pidfile and argsfile*

```
pidfile /var/run/slapd/slapd.pid
argsfile /var/run/slapd/slapd.args
```

These two files contain the PID (process ID) and some of the arguments the `slapd` process is started with. There is no need for modifications here.

Example 26.4 *slapd.conf: Access Control*

```
# Sample Access Control
#       Allow read access of root DSE
# Allow self write access
#       Allow authenticated users read access
#       Allow anonymous users to authenticate
# access to dn="" by * read
  access to * by self write
              by users read
              by anonymous auth
#
# if no access controls are present, the default is:
#       Allow read by all
#
# rootdn can always write!
```

Example 26.4, "slapd.conf: Access Control" (page 430) is the excerpt from `slapd.conf` that regulates the access permissions for the LDAP directory on the server. The settings made here in the global section of `slapd.conf` are valid as long as no custom access rules are declared in the database-specific section. These would overwrite the global declarations. As presented here, all users have read access to the directory, but only the administrator (`rootdn`) can write to this directory. Access control regulation in LDAP is a highly complex process. The following tips can help:

- Every access rule has the following structure:

  ```
  access to <what> by <who> <access>
  ```

- *`what`* is a placeholder for the object or attribute to which access is granted. Individual directory branches can be protected explicitly with separate rules. It is also possible to process regions of the directory tree with one rule by using regular expressions. `slapd` evaluates all rules in the order in which they are listed in the configuration file. More general rules should be listed after more specific ones—the first rule `slapd` regards as valid is evaluated and all following entries are ignored.

- *`who`* determines who should be granted access to the areas determined with *`what`*. Regular expressions may be used. `slapd` again aborts the evaluation of `who` after the first match, so more specific rules should be listed before the more general ones. The entries shown in Table 26.2, "User Groups and Their Access Grants" (page 431) are possible.

Table 26.2 *User Groups and Their Access Grants*

Tag	Scope
`*`	All users without exception
`anonymous`	Not authenticated ("anonymous") users
`users`	Authenticated users
`self`	Users connected with the target object
`dn.regex=<regex>`	All users matching the regular expression

- *`access`* specifies the type of access. Use the options listed in Table 26.3, "Types of Access" (page 432).

Table 26.3 *Types of Access*

Tag	Scope of Access
`none`	No access
`auth`	For contacting the server
`compare`	To objects for comparison access
`search`	For the employment of search filters
`read`	Read access
`write`	Write access

`slapd` compares the access right requested by the client with those granted in `slapd.conf`. The client is granted access if the rules allow a higher or equal right than the requested one. If the client requests higher rights than those declared in the rules, it is denied access.

Example 26.5, “slapd.conf: Example for Access Control” (page 432) shows an example of a simple access control that can be arbitrarily developed using regular expressions.

Example 26.5 *slapd.conf: Example for Access Control*

```
access to  dn.regex="ou=([^,]+),dc=example,dc=com"
by dn.regex="cn=Administrator,ou=$1,dc=example,dc=com" write
by user read
by * none
```

This rule declares that only its respective administrator has write access to an individual `ou` entry. All other authenticated users have read access and the rest of the world has no access.

TIP: Establishing Access Rules

If there is no `access to` rule or no matching `by` directive, access is denied. Only explicitly declared access rights are granted. If no rules are declared at all, the default principle is write access for the administrator and read access for the rest of the world.

Find detailed information and an example configuration for LDAP access rights in the online documentation of the installed `openldap2` package.

Apart from the possibility to administer access permissions with the central server configuration file (`slapd.conf`), there is access control information (ACI). ACI allows storage of the access information for individual objects within the LDAP tree. This type of access control is not yet common and is still considered experimental by the developers. Refer to `http://www.openldap.org/faq/data/cache/758.html` for information.

26.7.2 Database-Specific Directives in slapd.conf

Example 26.6 *slapd.conf: Database-Specific Directives*

```
database bdb❶
suffix "dc=example,dc=com"❷
checkpoint      1024    5❸
cachesize       10000❹
rootdn "cn=Administrator,dc=example,dc=com"❺
# Cleartext passwords, especially for the rootdn, should
# be avoided.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw secret❻
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd/tools. Mode 700 recommended.
directory /var/lib/ldap❼
# Indices to maintain
index   objectClass     eq❽
overlay ppolicy❾
ppolicy_default "cn=Default Password Policy,dc=example,dc=com"
ppolicy_hash_cleartext
ppolicy_use_lockout
```

❶ The type of database, a Berkeley database in this case, is set in the first line of this section (see Example 26.6, “slapd.conf: Database-Specific Directives” (page 433)).

❷ `suffix` determines for which portion of the LDAP tree this server should be responsible.

❸ `checkpoint` determines the amount of data (in KB) that is kept in the transaction log before it is written to the actual database and the time (in minutes) between two write actions.

❹ `cachesize` sets the number of objects kept in the database's cache.

❺ `rootdn` determines who owns administrator rights to this server. The user declared here does not need to have an LDAP entry or exist as regular user.

❻ `rootpw` sets the administrator password. Instead of using `secret` here, it is possible to enter the hash of the administrator password created by `slappasswd`.

❼ The `directory` directive indicates the directory in the file system where the database directories are stored on the server.

❽ The last directive, `index objectClass eq`, results in the maintenance of an index of all object classes. Attributes for which users search most often can be added here according to experience.

❾ `overlay ppolicy` adds a layer of password control mechanisms. `ppolicy_default` specifies the DN of the pwdPolicy object to use when no specific policy is set on a given user's entry. If there is no specific policy for an entry and no default is given, no policies are enforced. `ppolicy_hash_cleartext` specifies that clear text passwords present in add and modify requests are hashed before being stored in the database. When this option is used, it is recommended to deny compare, search, and read access to the `userPassword` attribute for all directory users, because `ppolicy_hash_cleartext` violates the X.500/LDAP information model. `ppolicy_use_lockout` sends a specific error code when a client tries to connect to a locked account. When your site is sensitive to security issues, disable this option as the error code provides useful information to attackers.

Custom `Access` rules defined here for the database are used instead of the global `Access` rules.

26.7.3 Starting and Stopping the Servers

Once the LDAP server is fully configured and all desired entries have been made according to the pattern described in Section 26.8, “Manually Administering LDAP Data” (page 435), start the LDAP server as `root` by entering `rcldap start`. To stop the server manually, enter the command `rcldap stop`. Request the status of the running LDAP server with `rcldap status`.

The YaST runlevel editor, described in Section 12.2.3, “Configuring System Services (Runlevel) with YaST” (page 190), can be used to have the server started and stopped automatically on boot and halt of the system. It is also possible to create the corresponding links to the start and stop scripts with the `insserv` command from a command prompt as described in Section 12.2.2, "Init Scripts" (page 186).

26.8 Manually Administering LDAP Data

OpenLDAP offers a series of tools for the administration of data in the LDAP directory. The four most important tools for adding to, deleting from, searching through and modifying the data stock are explained below.

26.8.1 Inserting Data into an LDAP Directory

Once the configuration of your LDAP server in `/etc/openldap/slapd.conf` is correct and ready to go (it features appropriate entries for `suffix`, `directory`, `rootdn`, `rootpw` and `index`), proceed to entering records. OpenLDAP offers the `ldapadd` command for this task. If possible, add the objects to the database in bundles for practical reasons. LDAP is able to process the LDIF format (LDAP data interchange format) for this. An LDIF file is a simple text file that can contain an arbitrary number of attribute and value pairs. Refer to the schema files declared in `slapd.conf` for the available object classes and attributes. The LDIF file for creating a rough framework for the example in Figure 26.1, "Structure of an LDAP Directory" (page 412) would look like that in Example 26.7, "Example for an LDIF File" (page 436).

IMPORTANT: Encoding of LDIF Files

LDAP works with UTF-8 (Unicode). Umlauts must be encoded correctly. Use an editor that supports UTF-8, such as Kate or recent versions of Emacs. Otherwise, avoid umlauts and other special characters or use `recode` to recode the input to UTF-8.

Example 26.7 *Example for an LDIF File*

```
# The Organization
dn: dc=example,dc=com
objectClass: dcObject
objectClass: organization
o: Example dc: example

# The organizational unit development (devel)
dn: ou=devel,dc=example,dc=com
objectClass: organizationalUnit
ou: devel

# The organizational unit documentation (doc)
dn: ou=doc,dc=example,dc=com
objectClass: organizationalUnit
ou: doc

# The organizational unit internal IT (it)
dn: ou=it,dc=example,dc=com
objectClass: organizationalUnit
ou: it
```

Save the file with the `.ldif` suffix then pass it to the server with the following command:

```
ldapadd -x -D <dn of the administrator> -W -f <file>.ldif
```

`-x` switches off the authentication with SASL in this case. `-D` declares the user that calls the operation. The valid DN of the administrator is entered here just like it has been configured in `slapd.conf`. In the current example, this is `cn=Administrator,dc=example,dc=com`. `-W` circumvents entering the password on the command line (in clear text) and activates a separate password prompt. This password was previously determined in `slapd.conf` with `rootpw`. The `-f` option passes the filename. See the details of running `ldapadd` in Example 26.8, “ldapadd with example.ldif” (page 436).

Example 26.8 *ldapadd with example.ldif*

```
ldapadd -x -D cn=Administrator,dc=example,dc=com -W -f example.ldif

Enter LDAP password:
adding new entry "dc=example,dc=com"
adding new entry "ou=devel,dc=example,dc=com"
adding new entry "ou=doc,dc=example,dc=com"
adding new entry "ou=it,dc=example,dc=com"
```

The user data of individuals can be prepared in separate LDIF files. Example 26.9, “LDIF Data for Tux” (page 437) adds Tux to the new LDAP directory.

Example 26.9 *LDIF Data for Tux*

```
# coworker Tux
dn: cn=Tux Linux,ou=devel,dc=example,dc=com
objectClass: inetOrgPerson
cn: Tux Linux
givenName: Tux
sn: Linux
mail: tux@example.com
uid: tux
telephoneNumber: +49 1234 567-8
```

An LDIF file can contain an arbitrary number of objects. It is possible to pass entire directory branches to the server at once or only parts of it as shown in the example of individual objects. If it is necessary to modify some data relatively often, a fine subdivision of single objects is recommended.

26.8.2 Modifying Data in the LDAP Directory

The tool `ldapmodify` is provided for modifying the data stock. The easiest way to do this is to modify the corresponding LDIF file then pass this modified file to the LDAP server. To change the telephone number of colleague Tux from `+49 1234 567-8` to `+49 1234 567-10`, edit the LDIF file like in Example 26.10, “Modified LDIF File tux.ldif” (page 437).

Example 26.10 *Modified LDIF File tux.ldif*

```
# coworker Tux
dn: cn=Tux Linux,ou=devel,dc=example,dc=com
changetype: modify
replace: telephoneNumber
telephoneNumber: +49 1234 567-10
```

Import the modified file into the LDAP directory with the following command:

```
ldapmodify -x -D cn=Administrator,dc=example,dc=com -W -f tux.ldif
```

Alternatively, pass the attributes to change directly to `ldapmodify`. The procedure for this is described below:

1. Start `ldapmodify` and enter your password:

```
ldapmodify -x -D cn=Administrator,dc=example,dc=com -W
Enter LDAP password:
```

2 Enter the changes while carefully complying with the syntax in the order presented below:

```
dn: cn=Tux Linux,ou=devel,dc=example,dc=com
changetype: modify
replace: telephoneNumber
telephoneNumber: +49 1234 567-10
```

Find detailed information about `ldapmodify` and its syntax in the `ldapmodify` man page.

26.8.3 Searching or Reading Data from an LDAP Directory

OpenLDAP provides, with `ldapsearch`, a command line tool for searching data within an LDAP directory and reading data from it. A simple query would have the following syntax:

```
ldapsearch -x -b dc=example,dc=com "(objectClass=*)"
```

The `-b` option determines the search base—the section of the tree within which the search should be performed. In the current case, this is `dc=example,dc=com`. To perform a more finely-grained search in specific subsections of the LDAP directory (for example, only within the `devel` department), pass this section to `ldapsearch` with `-b`. `-x` requests activation of simple authentication. `(objectClass=*)` declares that all objects contained in the directory should be read. This command option can be used after the creation of a new directory tree to verify that all entries have been recorded correctly and the server responds as desired. Find more information about the use of `ldapsearch` in the corresponding man page (ldapsearch(1)).

26.8.4 Deleting Data from an LDAP Directory

Delete unwanted entries with `ldapdelete`. The syntax is similar to that of the other commands. To delete, for example, the complete entry for `Tux Linux`, issue the following command:

```
ldapdelete -x -D cn=Administrator,dc=example,dc=com -W cn=Tux \
Linux,ou=devel,dc=example,dc=com
```

26.9 For More Information

More complex subjects, like SASL configuration or establishment of a replicating LDAP server that distributes the workload among multiple slaves, were intentionally not included in this chapter. Detailed information about both subjects can be found in the *OpenLDAP 2.2 Administrator's Guide*.

The Web site of the OpenLDAP project offers exhaustive documentation for beginning and advanced LDAP users:

OpenLDAP Faq-O-Matic
: A very rich question and answer collection concerning installation, configuration, and use of OpenLDAP. Find it at `http://www.openldap.org/faq/data/cache/1.html`.

Quick Start Guide
: Brief step-by-step instructions for installing your first LDAP server. Find it at `http://www.openldap.org/doc/admin22/quickstart.html` or on an installed system in `/usr/share/doc/packages/openldap2/admin-guide/quickstart.html`.

OpenLDAP 2.2 Administrator's Guide
: A detailed introduction to all important aspects of LDAP configuration, including access controls and encryption. See `http://www.openldap.org/doc/admin22/` or, on an installed system, `/usr/share/doc/packages/openldap2/admin-guide/index.html`.

Understanding LDAP

A detailed general introduction to the basic principles of LDAP: http://www.redbooks.ibm.com/redbooks/pdfs/sg244986.pdf.

Printed literature about LDAP:

- *LDAP System Administration* by Gerald Carter (ISBN 1-56592-491-6)
- *Understanding and Deploying LDAP Directory Services* by Howes, Smith, and Good (ISBN 0-672-32316-8)

The ultimate reference material for the subject of LDAP is the corresponding RFCs (request for comments), 2251 to 2256.

Active Directory Support 27

Active Directory* (AD) is a directory service based on LDAP, Kerberos, and other services that is used by Microsoft Windows to manage resources, services, and people. In an MS Windows network, AD provides information about these objects, restricts access to any of them, and enforces policies. openSUSE® lets you join existing AD domains and integrate your Linux machine into a Windows environment.

27.1 Integrating Linux and AD Environments

With a Linux client configured as an Active Directory client that is joined to an existing Active Directory domain, benefit from various features not available on a pure openSUSE Linux client:

Browsing Shared Files and Folders with SMB
: Both Nautilus, the GNOME file manager, and Konqueror, its KDE counterpart, support browsing shared resources through SMB.

Sharing Files and Folders with SMB
: Both Nautilus, the GNOME file manager, and Konqueror, its KDE counterpart, support sharing folders and files as in Windows.

Accessing and Manipulating User Data on the Windows Server
: Through Nautilus and Konqueror, users are able to access their Windows user data and can edit, create, and delete files and folders on the Windows server. Users can access their data without having to enter their password again and again.

Offline Authentication
: Users are able to log in and access their local data on the Linux machine even if they are offline (for example, using a laptop) or the AD server is unavailable for other reasons.

Windows Password Change
: This port of AD support in Linux enforces corporate password policies stored in Active Directory. The display managers and console support password change messages and accept your input. You can even use the Linux `passwd` command to set Windows passwords.

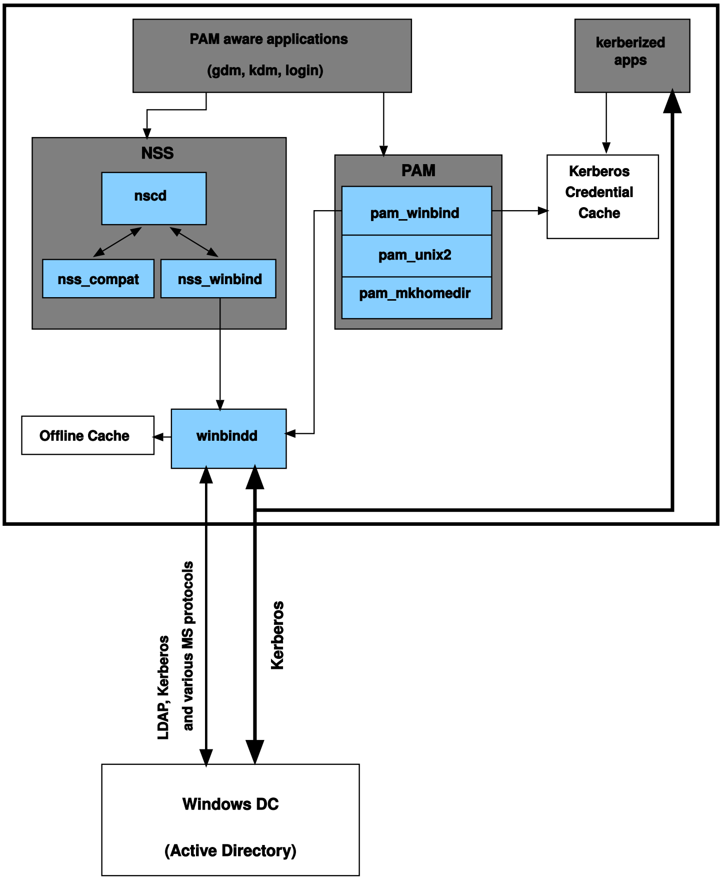
Single-Sign-On through Kerberized Applications
: Many applications of both desktops are Kerberos-enabled (*kerberized*), which means they can transparently handle authentication for the user without the need for password reentry at Web servers, proxies, groupware applications, or other locations.

A brief technical background for most of these features is given in the following section. For directions for file and printer sharing, refer to *GNOME User Guide* and *KDE User Guide*, where you can learn more about AD enablement in the GNOME and KDE application worlds.

27.2 Background Information for Linux AD Support

Many system components need to interact flawlessly to integrate a Linux client into an existing Windows Active Directory domain. Figure 27.1, “Active Directory Authentication Schema” (page 443) highlights the most prominent ones. The following sections focus on the underlying processes of the key events in AD server and client interaction.

Figure 27.1 Active Directory Authentication Schema



To communicate with the directory service, the client needs to share at least two protocols with the server:

LDAP
: LDAP is a protocol optimized for managing directory information. A Windows domain controller with AD can use the LDAP protocol to exchange directory information with the clients. To learn more about LDAP in general and about the open source port of it, OpenLDAP, refer to Chapter 26, *LDAP—A Directory Service* (page 409).

Kerberos
: Kerberos is a third-party trusted authentication service. All its clients trust Kerberos's judgment of another client's identity, enabling kerberized single-sign-on (SSO) solutions. Windows supports a Kerberos implementation, making Kerberos SSO possible even with Linux clients.

The following client components process account and authentication data:

Winbind
: The most central part of this solution is the winbind daemon that is a part of the Samba project and handles all communication with the AD server.

NSS (*Name Service Switch*)
: NSS routines provide name service information. Naming service for both users and groups is provided by `nss_winbind`. This module directly interacts with the winbind daemon.

PAM (*Pluggable Authentication Modules*)
: User authentication for AD users is done by the `pam_winbind` module. The creation of user homes for the AD users on the Linux client is handled by `pam_mkhomedir`. The `pam_winbind` module directly interacts with winbindd. To learn more about PAM in general, refer to Chapter 18, *Authentication with PAM* (page 263).

Applications that are PAM-aware, like the login routines and the GNOME and KDE display managers, interact with the PAM and NSS layer to authenticate against the Windows server. Applications supporting Kerberos authentication, such as file managers, Web browsers, or e-mail clients, use the Kerberos credential cache to access user's Kerberos tickets, making them part of the SSO framework.

27.2.1 Domain Join

During domain join, the server and the client establish a secure relation. On the client, the following tasks need to be performed to join the existing LDAP and Kerberos SSO environment provided by the Window domain controller. The entire join process is handled by the YaST Domain Membership module that can be run during installation or in the installed system:

1. The Windows domain controller providing both LDAP and KDC (Key Distribution Center) services is located.

2. A machine account for the joining client is created in the directory service.

3. An initial ticket granting ticket (TGT) is obtained for the client and stored in its local Kerberos credential cache. The client needs this TGT to get further tickets allowing it to contact other services, like contacting the directory server for LDAP queries.

4. NSS and PAM configurations are adjusted to enable the client to authenticate against the domain controller.

During client boot, the winbind daemon is started and retrieves the initial Kerberos ticket for the machine account. winbindd automatically refreshes the machine's ticket to keep it valid. To keep track of the current account policies, winbindd periodically queries the domain controller.

27.2.2 Domain Login and User Homes

The login managers of GNOME and KDE (GDM and KDM) have been extended to allow the handling of AD domain login. Users can choose to log in to the primary domain the machine has joined or to one of the trusted domains with which the domain controller of the primary domain has established a trust relationship.

User authentication is mediated by a number of PAM modules as described in Section 27.2, "Background Information for Linux AD Support" (page 442). The `pam_winbind` module used to authenticate clients against Active Directory or NT4 domains is fully aware of Windows error conditions that might prohibit a user's login. The Windows error codes are translated into appropriate user-readable error messages that PAM gives at login through any of the supported methods (GDM, KDM, console, and SSH):

`Password has expired`

The user sees a message stating that the password has expired and needs to be changed. The system prompts directly for a new password and informs the user if the new password does not comply with corporate password policies, for example, the password is too short, too simple, or already in the history. If a user's password change fails, the reason is shown and a new password prompt is given.

`Account disabled`
: The user sees an error message stating that his account has been disabled and that he should contact the system administrator.

`Account locked out`
: The user sees an error message stating that his account has been locked and that he should contact the system administrator.

`Password has to be changed`
: The user can log in but receives a warning that the password needs to be changed soon. This warning is sent three days before that password expires. After expiration, the user cannot login again.

`Invalid workstation`
: When a user is just allowed to log in from specific workstations and the current openSUSE machine is not in that list, a message appears that this user cannot log in from this workstation.

`Invalid logon hours`
: When a user is only allowed to log in during working hours and tries to log in outside working hours, a message shows that login is not possible at this point in time.

`Account expired`
: An administrator can set an expiration time for a specific user account. If that user tries to log in after that time has passed, the user gets a message that the account has expired and cannot be used to log in.

During a successful authentication, `pam_winbind` acquires a ticket granting ticket (TGT) from the Kerberos server of Active Directory and stores it in the user's credential cache. It also takes care of renewing the TGT in the background, not requiring any user interaction.

openSUSE supports local home directories for AD users. If configured through YaST as described in Section 27.3, "Configuring a Linux Client for Active Directory" (page 447), user homes are created at the first login of a Windows (AD) user into the Linux client. These home directories look and feel entirely the same as standard Linux user home directories and work independently of the AD domain controller. Using a local user home, it is possible to access a user's data on this machine, even when the

AD server is disconnected, if the Linux client has been configured to perform offline authentication.

27.2.3 Offline Service and Policy Support

Users in a corporate environment must have the ability to become roaming users, for example, to switch networks or even work disconnected for some time. To enable users to log in to a disconnected machine, extensive caching was integrated into the winbind daemon. The winbind daemon enforces password policies even in the offline state. It tracks the number of failed login attempts and reacts according to the policies configured in Active Directory. Offline support is disabled by default and must be explicitly enabled in the YaST Domain Membership module.

As in Windows, when the domain controller has become unavailable, the user can still access network resources (other than the AD server itself) with valid Kerberos tickets that have been acquired before losing the connection. Password changes cannot be processed unless the domain controller is online. While disconnected from the AD server, a user cannot access any data stored on this server. When a workstation has become disconnected from the network entirely and attaches to the corporate network again later, openSUSE acquires a new Kerberos ticket as soon as the user has locked and unlocked the desktop (for example, using a desktop screen saver).

27.3 Configuring a Linux Client for Active Directory

Before your client can join an AD domain, some adjustments must be made to your network setup to ensure a flawless interaction of client and server.

DNS

Configure your client machine to use a DNS server that can forward DNS requests to the AD DNS server. Alternatively, configure your machine to use the AD DNS server as the name service data source.

NTP

To succeed with Kerberos authentication, the client must have have its time set accurately. It is highly encouraged to use a central NTP time server for this purpose (this can be also the NTP server running on your Active Directory domain controller). If the clockskew between your Linux host and the domain controller exceeds a certain limit, Kerberos authentication fails and the client is logged in only using the weaker NTLM (NT LAN Manager) authentication.

DHCP

If your client uses dynamic network configuration with DHCP, configure DHCP to provide the same IP and hostname to the client. If possible, use static IP addresses to be on the safe side.

Firewall

To browse your network neighborhood, either disable the firewall entirely or mark the interface used for browsing as part of the internal zone.

To change the firewall settings on your client, log in as `root` and start the YaST firewall module. Select *Interfaces*. Select your network interface from the list of interfaces and click *Change*. Select *Internal Zone* and apply your settings with *OK*. Leave the firewall settings with *Next* > *Accept*. To disable the firewall, just set *Service Start* to *Manually* and leave the firewall module with *Next* > *Accept*.

AD Account

You cannot log in to an AD domain unless the AD administrator has provided you with a valid user account for this domain. Use the AD username and password to log in to the AD domain from your Linux client.

Join an existing AD domain during installation or by later activating SMB user authentication with YaST in the installed system.

NOTE

Currently only a domain administrator account, such as `Administrator`, can join openSUSE into Active Directory.

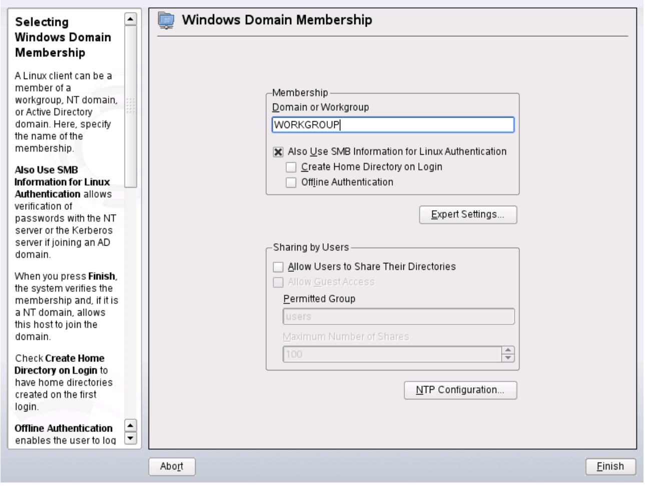
To join an AD domain in a running system, proceed as follows:

1 Log in as `root` and start YaST.

2 Start *Network Services* > *Windows Domain Membership*.

3 Enter the domain to join at *Domain or Workgroup* in the *Windows Domain Membership* screen (see Figure 27.2, "Determining Windows Domain Membership” (page 449)). If the DNS settings on your host are properly integrated with the Windows DNS server, enter the AD domain name in its DNS format (`mydomain.mycompany.com`). If you enter the short name of your domain (also known as the pre–Windows 2000 domain name), YaST must rely on NetBIOS name resolution instead of DNS to find the correct domain controller. To select from a list of available domains instead, use *Browse* to list the NetBIOS domains then select the desired domain.

Figure 27.2 *Determining Windows Domain Membership*



Selecting Windows Domain Membership

A Linux client can be a member of a workgroup, NT domain, or Active Directory domain. Here, specify the name of the membership.

Also Use SMB Information for Linux Authentication allows verification of passwords with the NT server or the Kerberos server if joining an AD domain.

When you press **Finish**, the system verifies the membership and, if it is a NT domain, allows this host to join the domain.

Check **Create Home Directory on Login** to have home directories created on the first login.

Offline Authentication enables the user to log

Windows Domain Membership

Membership

Domain or Workgroup: WORKGROUP

☒ Also Use SMB Information for Linux Authentication

□ Create Home Directory on Login

□ Offline Authentication

Sharing by Users

□ Allow Users to Share Their Directories

□ Allow Guest Access

Permitted Group: users

Maximum Number of Shares: 100

4 Check *Also Use SMB Information for Linux Authentication* to use the SMB source for Linux authentication.

5 Check *Create Home Directory on Login* to automatically create a local home directory for your AD user on the Linux machine.

6 Check *Offline Authentication* to allow your domain users to log in even if the AD server is temporarily unavailable or you do not have a network connection.

7 Select *Expert Settings*, if you want to change the UID and GID ranges for the Samba users and groups. Let DHCP retrieve the WINS server (default setting).

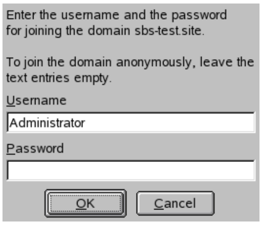
8 Configure NTP time synchronization for your AD environment by selecting *NTP Configuration* and entering an appropriate server name or IP address.

This step is obsolete if you have already entered the appropriate settings in the standalone YaST NTP configuration module.

9 Click *Finish* and confirm the domain join when prompted for it.

10 Provide the password for the Windows administrator on the AD server and click *OK* (see Figure 27.3, “Providing Administrator Credentials” (page 450)).

***Figure 27.3** Providing Administrator Credentials*



Enter the username and the password for joining the domain sbs-test.site.

To join the domain anonymously, leave the text entries empty.

Username: Administrator

Password:

After you have joined the AD domain, you can log in to it from your workstation using the display manager of your desktop or the console.

27.4 Logging In to an AD Domain

Provided your machine has been configured to authenticate against Active Directory and you have a valid Windows user identity, you can log in to your machine using the AD credentials. Login is supported for both desktop environments (GNOME and KDE), the console, SSH, and any other PAM-aware application.

IMPORTANT: Offline Authentication

openSUSE supports offline authentication, allowing you to remain logged in to your client machine even if the client machine is disconnected from the network. This enables you to maintain a mobile style of working, for example, it allows you to continue to work even if you are on an airplane and do not have a network connection.

27.4.1 GDM and KDM

To authenticate a GNOME client machine against an AD server, proceed as follows:

1. Select the domain.

2. Enter your Windows username and press Enter.

3. Enter your Windows password and press Enter.

To authenticate a KDE client machine against an AD server, proceed as follows:

1. Select the domain.

2. Enter your Windows username.

3. Enter your Windows password and press Enter.

If configured to do so, openSUSE creates a user home directory on the local machine on the first login of each AD authenticated user. This allows you to benefit from the AD support of openSUSE while still having a completely capable Linux machine at your disposal.

27.4.2 Console Login

As well as logging in to the AD client machine using a graphical front-end, you can log in using the text-based console login or even remotely using SSH.

To log in to your AD client from a console, enter `*DOMAIN*\*user*` at the `login:` prompt and provide the password.

To remotely log in to your AD client machine using SSH, proceed as follows:

1. At the login prompt, enter:

   ```
   ssh DOMAIN\\user@hostname
   ```

   The \ domain and login delimiter is escaped with another \ sign.

2. Provide the user's password.

27.5 Changing Passwords

openSUSE has the ability to help a user choose a suitable new password that meets the corporate security policy. The underlying PAM module retrieves the current password policy settings from the domain controller. It informs about the specific password quality requirements a user account typically has by means of a message at login time. Like the Windows counterpart, openSUSE presents a message describing:

- Password history settings
- Minimum password length requirements
- Minimum password age
- Password complexity

The password change process cannot succeed unless all possible requirements have been successfully satisfied. Feedback about the password status is given both through the display managers and the console.

GDM and KDM provide feedback about password expiration and prompt for new passwords in an interactive mode. To change passwords in the display managers, just provide the password information when prompted to do so.

To change your Windows password, you can use the standard Linux utility, `passwd`, instead of having to manipulate this data on the server. To change your Windows password, proceed as follows:

1. Log in at the console.
2. Enter `passwd`.
3. Enter your current password when prompted to do so.
4. Enter the new password.
5. Reenter the new password for confirmation. If your new password does not comply with the policies on the Windows server, this feedback is given to you and you are prompted for another password.

To change your Windows password from the GNOME desktop, proceed as follows:

1. Click the *Computer* icon on the left edge of the panel.

2. Select *Control Center*.

3. From the *Personal* section, select *Change Password*.

4. Enter your old password.

5. Enter and confirm the new password.

6. Leave the dialog with *Close* to apply your settings.

To change your Windows password from the KDE desktop, proceed as follows:

1. Select *Personal Settings* from the main menu.

2. Select *Security & Privacy*.

3. Click *Password & User Account*.

4. Click *Change Password*.

5. Enter your current password.

6. Enter and confirm the new password and apply your settings with *OK*.

7. Leave the *Personal Settings* with *File* > *Quit*.

Sharing File Systems with NFS 28

One of the possibilities to distribute files over the network is NFS (Network File System). NFS works together with network information services like NIS (see also Chapter 25, *Using NIS* (page 401) or a directory service like LDAP (see also Chapter 26, *LDAP—A Directory Service* (page 409) to handle the information about how to use the available services. To prevent unauthorized access, NFSv4 also makes it possible to use authentication with Kerberos (see also Chapter 39, *Installing and Administering Kerberos* (page 621). When configured correctly, it does not matter at which terminal users are logged in, they always find themselves in the same environment.

Like NIS, NFS is a client/server system. A machine can be both—it can supply file systems over the network (export) and mount file systems from other hosts (import).

In principle, all exports can be made using IP addresses only. To avoid time-outs, however, you should have a working DNS system. This is necessary at least for logging purposes, because the mounted daemon does reverse lookups.

All networked services heavily rely on a correct system time. If you intend to setup such services, one of the first things that should be configured is the time synchronization as described in Chapter 24, *Time Synchronization with NTP* (page 393).

28.1 NFS Client Configuration

To configure your host as an NFS client, you do not need to install additional software. All packages needed to configure an NFS client are installed by default.

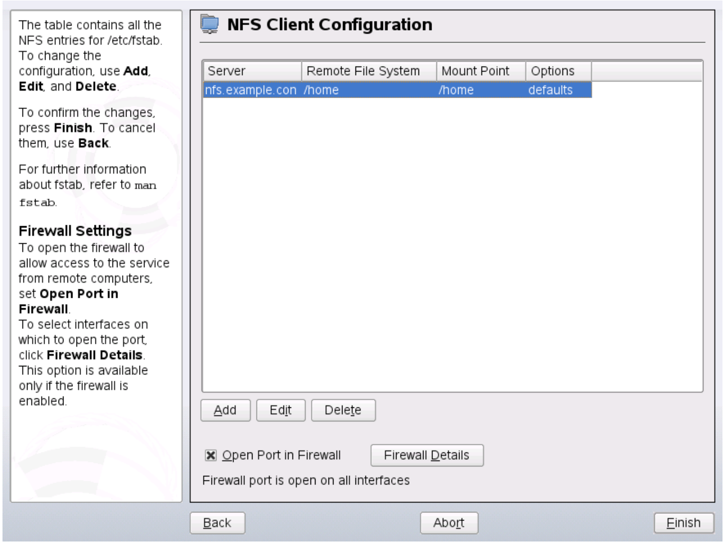
28.1.1 Importing File Systems with YaST

Users authorized to do so can mount NFS directories from an NFS server into their own file trees. This can be achieved most easily using the YaST module *NFS Client*. Just enter the hostname of the NFS server, the directory to import, and the mount point at which to mount this directory locally. All this is done after *Add* is clicked in the first dialog. Activate *Open Port in Firewall* to open the firewall to allow access to the service from remote computers. The firewall status is displayed next to the check box. Press *Finish* to saves your changes. See Figure 28.1, "NFS Client Configuration with YaST" (page 456).

This saves the changes to `/etc/fstab` and imports the specified file systems. When you start the client at a later point in time, it reads the information from this file.

Currently, it is not possible to import NFSv4 file systems with YaST. NFSv4 file systems can be imported manually or with the automounter. This is explained in Section 28.1.3, "Importing NFSv4 File Systems Manually" (page 457) and Section 28.1.4, "Using the Automount Service" (page 458).

Figure 28.1 *NFS Client Configuration with YaST*



The table contains all the NFS entries for /etc/fstab. To change the configuration, use **Add**, **Edit**, and **Delete**.

To confirm the changes, press **Finish**. To cancel them, use **Back**.

For further information about fstab, refer to `man fstab`.

Firewall Settings
To open the firewall to allow access to the service from remote computers, set **Open Port in Firewall**.
To select interfaces on which to open the port, click **Firewall Details**. This option is available only if the firewall is enabled.

Server	Remote File System	Mount Point	Options
nfs.example.con	/home	/home	defaults

☒ Open Port in Firewall

Firewall port is open on all interfaces

28.1.2 Importing NFSv3 File Systems Manually

The only prerequisite to mount NFSv3 file systems is a running RPC port mapper. Start this service by entering `rcportmap start` as `root`. Once this prerequisite is met, remote exported file systems can be mounted in the file system just like local hard disks using the `mount` command in the following manner:

```
mount host:remote-path local-path
```

If the `/home` directory from the machine nfs.example.com, should be imported, first create a local directory `/home` and then use the following command:

```
mount nfs.example.com:/home /home
```

28.1.3 Importing NFSv4 File Systems Manually

The idmapd service must be up and running on the client to do an NFSv4 import. Start the idmapd service from the command prompt with `rpc.idmapd`. Use the command `ps x | grep rpc.idmapd` to check the status of idmapd.

The idmapd services get their configuration parameters from the file `/etc/idmapd.conf`. The value of the `Domain` parameter must be the same as on all NFSv4 servers and clients of this domain. For example, this may be `localdomain` . Ensure that the value specified is the same for both the NFS client and NFS server.

Make NFSv4 imports by giving a command from the shell prompt. To import NFSv4 remote file systems, use the following command:

```
mount -t nfs4 host:/ local-path
```

Replace `host` with the NFS server that hosts one or more NFSv4 exports and `local-path` with the directory location in the client machine where this should be mounted. For example, to import `/home` exported with NFSv4 on nfs.example.com to `/local/home`, use the following command:

```
mount -t nfs4 nfs.example.com:/ /local
```

Note, that the remote file system path that follows the server name and a colon is always a slash "/". This is unlike the way it is specified for v3 imports, where the exact path of the remote file system is given. This is a concept called *pseudo file system*, which is explained in Section 28.2, "Exporting File Systems over NFS" (page 459).

28.1.4 Using the Automount Service

As well as the regular local device mounts, the autofs daemon can be used to mount remote file systems automatically too. To do this, add the following entry in the your `/etc/auto.master` file:

```
/nfsmounts /etc/auto.nfs
```

Now the `/nfsmounts` directory acts as a root for all the NFS mounts on the client if the `auto.nfs` file is completed appropriately. The name `auto.nfs` is chosen for sake of convenience—you can choose any name. In the selected file (create it if it does not exist), add entries for all the NFS mounts as in the following example:

```
localdata -fstype=nfs server1:/data
nfs4mount -fstype=nfs4 server2:/
```

Make sure that `auto.nfs` is executable with the command `chmod 755 auto.nfs`. Then activate the settings with `rcautofs start`. For this example, `/nfsmounts/localdata`, the `/data` directory of server1, is then mounted with NFS and `/nfsmounts/nfs4mount` from server2 is mounted with NFSv4.

If the `/etc/auto.master` file is edited while the service autofs is running, the automounter must be restarted for the changes to take effect. Do this with `rcautofs restart`.

28.1.5 Manually Editing `/etc/fstab`

A typical NFSv3 mount entry in `/etc/fstab` looks like this:

```
nfs.example.com:/data /local/path nfs rw,noauto 0 0
```

NFSv4 mounts may also be added to the `/etc/fstab` file manually. For these mounts, use `nfs4` instead of `nfs` in the third column and make sure that the remote file system is given as / after the *`nfs.example.com:`* in the first column. A sample line for an NFSv4 mount in `/etc/fstab` looks like this:

```
nfs.example.com:/ /local/pathv4 nfs4 rw,noauto 0 0
```

The `noauto` option prevents the file system from being mounted automatically at start up. If you want to mount the respective file system manually, it is possible to shorten the command for mounting and it is only needed to provide the mount point as in:

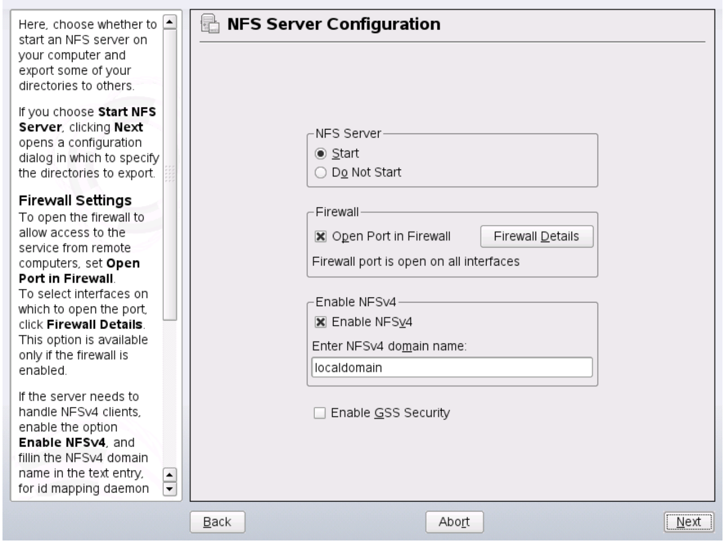
```
mount /local/path
```

Note, that if you do not enter the noauto option, the initialization scripts of the system will handle the mount of those file systems at start up.

28.2 Exporting File Systems over NFS

With YaST, turn a host in your network into an NFS server—a server that exports directories and files to all hosts granted access to it. This could be done to provide applications to all members of a group without installing them locally on each and every host. To install such a server, start YaST and select *Network Services* > *NFS Server*. A dialog like that in Figure 28.2, "NFS Server Configuration Tool" (page 459) opens.

Figure 28.2 *NFS Server Configuration Tool*



Here, choose whether to start an NFS server on your computer and export some of your directories to others.

If you choose **Start NFS Server**, clicking **Next** opens a configuration dialog in which to specify the directories to export.

Firewall Settings
To open the firewall to allow access to the service from remote computers, set **Open Port in Firewall**. To select interfaces on which to open the port, click **Firewall Details**. This option is available only if the firewall is enabled.

If the server needs to handle NFSv4 clients, enable the option **Enable NFSv4**, and fillin the NFSv4 domain name in the text entry, for id mapping daemon

NFS Server Configuration

NFS Server
● Start
○ Do Not Start

Firewall
☒ Open Port in Firewall
Firewall port is open on all interfaces

Enable NFSv4
☒ Enable NFSv4
Enter NFSv4 domain name: localdomain

□ Enable GSS Security

Next, activate *NFS Server: Start*. If you intend to use NFSv4, activate *Enable NFSv4* and enter the *NFSv4 domain name*. The NFSv4 domain name must be identical with the one used on all clients that should connect to the server.

Click *Enable GSS Security* if you need secure access to the server. A prerequisite for this is to have Kerberos installed in your domain and both the server and the clients are kerberized. Click *Next*.

In the upper text field, enter the directories to export. Below, enter the hosts that should have access to the respective directory. This dialog is shown in Figure 28.3, "Configuring an NFS Server with YaST" (page 461). The figure shows the scenario where NFSv4 is enabled in the previous dialog.

For a fixed set of clients, there are two types of directories that can be exported—directories that act as pseudo root file systems and those that are bound to some subdirectory of the pseudo file system. This pseudo file system acts as a base point under which all file systems exported for the same client set take their place. For a client or set of clients, only one directory on the server can be configured as pseudo root for export. For this same client, export multiple directories by binding them to some existing subdirectory in the pseudo root.

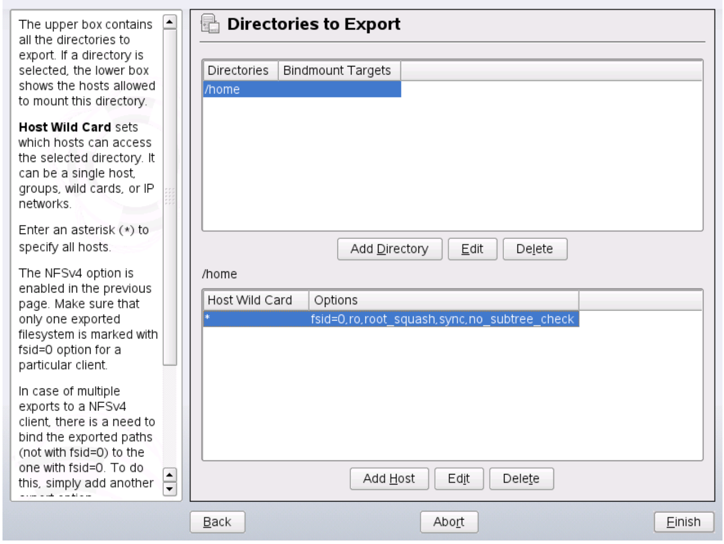
After adding a directory in the upper half, another dialog for entering the client and option information pops up automatically. Later on, to add a new client (client set), click *Add Host*.

In the small dialog that opens, enter the host wild card. There are four possible types of host wild cards that can be set for each host: a single host (name or IP address), netgroups, wild cards (such as `*` indicating all machines can access the server), and IP networks. Then, in *Options*, include `fsid=0` in the comma-separated list of options to configure the directory as pseudo root. If this directory should be bound to another directory under an already configured pseudo root, make sure that a target bind path is given in the option list with `bind=/target/path`.

For example, suppose that the directory `/exports` is chosen as the pseudo root directory for all the clients that can access the server. Then add this in the upper half and make sure that the options entered for this directory include `fsid=0`. If there is another directory, `/data`, that also needs to be NFSv4 exported, add this directory to the upper half. While entering options for this, make sure that `bind=/exports/data` is in the list and that `/exports/data` is an already existing subdirectory of `/exports`. Any change in the option `bind=/target/path`, whether addition, deletion, or change in value, is reflected in *Bindmount Targets*. This column is not directly editable, instead it contains summarizing directories and their nature. After the information is complete, click *Finish* to complete the configuration or *Start* to restart the service.

For more information about options available regarding the directory export, refer to the manual page of `Export`. Click *Finish* to complete the configuration.

***Figure 28.3** Configuring an NFS Server with YaST*



The upper box contains all the directories to export. If a directory is selected, the lower box shows the hosts allowed to mount this directory.

Host Wild Card sets which hosts can access the selected directory. It can be a single host, groups, wild cards, or IP networks.

Enter an asterisk (*) to specify all hosts.

The NFSv4 option is enabled in the previous page. Make sure that only one exported filesystem is marked with fsid=0 option for a particular client.

In case of multiple exports to a NFSv4 client, there is a need to bind the exported paths (not with fsid=0) to the one with fsid=0. To do this, simply add another

Directories to Export

Directories	Bindmount Targets
/home	

/home

Host Wild Card	Options
*	fsid=0,ro,root_squash,sync,no_subtree_check

IMPORTANT: Automatic Firewall Configuration

If a firewall is active on your system (SuSEfirewall2), YaST adapts its configuration for the NFS server by enabling the `nfs` service when *Open Ports in Firewall* is selected.

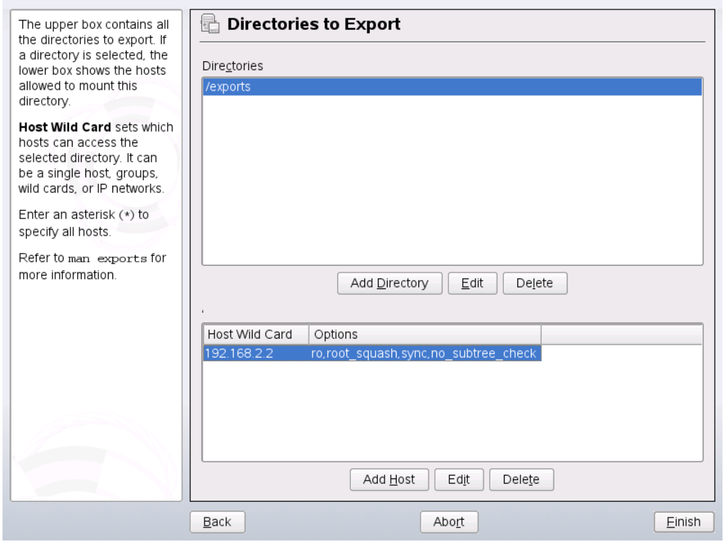
28.2.1 NFSv3 and NFSv2 Exports

Make sure that *Enable NFSv4* is not checked in the initial dialog before clicking *Next*.

The next dialog has two parts. In the upper text field, enter the directories to export. Below, enter the hosts that should have access to them. There are four types of host wild cards that can be set for each host: a single host (name or IP address), netgroups, wild cards (such as * indicating all machines can access the server), and IP networks.

This dialog is shown in Figure 28.4, "Exporting Directories with NFSv2 and v3" (page 462). Find a more thorough explanation of these options in `man exports`. Click *Finish* to complete the configuration.

***Figure 28.4** Exporting Directories with NFSv2 and v3*



The upper box contains all the directories to export. If a directory is selected, the lower box shows the hosts allowed to mount this directory.

Host Wild Card sets which hosts can access the selected directory. It can be a single host, groups, wild cards, or IP networks.

Enter an asterisk (*) to specify all hosts.

Refer to `man exports` for more information.

Directories to Export

Directories

/exports

Host Wild Card	Options
192.168.2.2	ro,root_squash,sync,no_subtree_check

28.2.2 Coexisting v3 and v4 Exports

Both NFSv3 and NFSv4 exports can coexist on a server. After enabling the support for NFSv4 in the initial configuration dialog, those exports for which `fsid=0` and `bind=/target/path` are not included in the option list are considered v3 exports. Consider the example in Figure 28.3, "Configuring an NFS Server with YaST" (page 461). If you add another directory, such as `/data2`, using *Add Directory* then in the corresponding options list do not mention either `fsid=0` or `bind=/target/path`, this export acts as a v3 export.

IMPORTANT

Automatic Firewall Configuration

If SuSEfirewall2 is active on your system, YaST adapts its configuration for the NFS server by enabling service when *Open Ports in Firewall* is selected.

28.3 Exporting File Systems Manually

The configuration files for the NFS export service are `/etc/exports` and `/etc/sysconfig/nfs`. In addition to these files, `/etc/idmapd.conf` is needed for the NFSv4 server configuration. To start or restart the services, run the commands `rcnfsserver restart`. This also starts the `rpc.idmapd` if NFSv4 configured in `/etc/sysconfig/nfs`. The NFS server depends on a running RPC portmapper. Therefore, also start or restart the portmapper service with `rcportmap restart`.

28.3.1 Exporting File Systems with NFSv4

There are some special configurations to accomplish when exporting file systems with NFSv4. The following sections handle these in more detail.

The `/etc/exports` File

This file contains a list of entries. Each entry indicates a directory that is shared and how it is shared. A typical entry in `/etc/exports` consists of:

```
/shared/directory host(option_list)
```

For example:

```
/export 192.168.1.2(rw,fsid=0,sync)
/data 192.168.1.2(rw,bind=/export/data,sync)
```

Those directories for which `fsid=0` is specified in the option list are called pseudo root file systems. Here, the IP address `192.168.1.2` is used. You can use the name of the host, a wild card indicating a set of hosts (`*.abc.com`, `*`, etc.), or netgroups.

For a fixed set of clients, there are only two types of directories that can be NFSv4 exported:

- A single directory that is chosen as the pseudo root file system. In this example, `/export` is the pseudo root directory because `fsid=0` is specified in the option list for this entry.
- Directories that are chosen to be bound to some an existing subdirectory of the pseudo file system. In the example entries above, `/data` is such a directory that binds to an existing subdirectory (`/export/data`) of the pseudo file system `/export`.

The pseudo file system is the top level directory under which all file systems that need to be NFSv4 exported take their places. For a client or set of clients, there can only be one directory on the server configured as the pseudo root for export. For this same client or client set, multiple other directories can be exported by binding them to some existing subdirectory in the pseudo root.

/etc/sysconfig/nfs

This file contains a few parameters that determine NFSv4 server daemon behavior. Importantly, the parameter NFSv4_SUPPORT must be set to yes. This parameter determines whether the NFS server supports NFSv4 exports and clients.

/etc/idmapd.conf

Every user on a Linux machine has a name and ID. idmapd does the name-to-ID mapping for NFSv4 requests to the server and replies to the client. This must be running on both server and client for NFSv4, because NFSv4 uses only names in its communication.

Make sure that there is a uniform way in which usernames and IDs (uid) are assigned to users across machines that might probably be sharing file systems using NFS. This can be achieved by using NIS, LDAP, or any uniform domain authentication mechanism in your domain.

For proper function, the parameter `Domain` must be set the same for both client and server in this file. If you are not sure, leave the domain as `localdomain` in both server and client files. A sample configuration file looks like the following:

```
[General]

Verbosity = 0
Pipefs-Directory = /var/lib/nfs/rpc_pipefs
Domain = localdomain

[Mapping]

Nobody-User = nobody
Nobody-Group = nobody
```

Do not change these parameters unless you are sure of what you are doing. For further reference, read the man page of `idmapd` and `idmapd.conf`; `man idmapd`, `man idmapd.conf` .

Starting and Stopping Services

After changing `/etc/exports` or `/etc/sysconfig/nfs`, start or restart the nfs server service with `rcnfsserver restart`. After changing `/etc/idmapd.conf`, start or restart the idmapd service with `rcidmapd restart`. Make sure that both services are running.

If this service should start at boot time, run the command `chkconfig nfsserver on`.

28.3.2 Exporting File Systems with NFSv2 and NFSv3

This is specific to NFSv3 and NFSv2 exports. Refer to Section 28.3.1, "Exporting File Systems with NFSv4" (page 463) for exporting with NFSv4.

Exporting file systems with NFS involves two configuration files: `/etc/exports` and `/etc/sysconfig/nfs`. A typical `/etc/exports` file entry is in the format:

```
/shared/directory host(list_of_options)
```

For example:

```
/export 192.168.1.2(rw,sync)
```

Here, the directory /export is shared with the host 192.168.1.2 with the option list rw,sync. This IP address can be replaced with a client name or set of clients using a wild card (such as *.abc.com) or even netgroups.

For a detailed explanation of all options and their meanings, refer to the man page of exports (man exports).

After changing /etc/exports or /etc/sysconfig/nfs, start or restart the NFS server using the command rcnfsserver restart.

28.4 NFS with Kerberos

To use Kerberos authentication for NFS, GSS security must be enabled. To do so, select *Enable GSS Security* in the initial YaST dialog. Note, that you must have a working Kerberos server to install this feature. YaST does not set up the server but only uses the provided functionality. If you want to use Kerberos authentication, in addition to the YaST configuration, complete at least the following steps before running the NFS configuration:

- Make sure that both the server and the client are in the same Kerberos domain. This means that they access the same KDC (Key Distribution Center) server and share their krb5.keytab file (the default location on any machine is /etc/krb5.keytab).

- Start the gssd service on the client with rcgssd start.

- Start the svcgssd service on the server with rcsvcgssd start.

For further information about configuring kerberized NFS, refer to the links in Section 28.5, “For More Information” (page 466).

28.5 For More Information

As well as the man pages of exports, nfs, and mount, information about configuring an NFS server and client is available in /usr/share/doc/packages/nfs-tils/README. Online documentation can be found at the following Web documents:

- Find the detailed technical documentation online at SourceForge [http://nfs.sourceforge.net/].
- For instructions for setting up kerberized NFS, refer to NFS Version 4 Open Source Reference Implementation [http://www.citi.umich.edu/projects/nfsv4/linux/krb5-setup.html].
- If you have any questions on NFSv4, refer to the Linux NFSv4 Frequently Asked Questions [http://www.citi.umich.edu/projects/nfsv4/linux/faq/] FAQ.

29 Samba

Using Samba, a Unix machine can be configured as a file and print server for DOS, Windows, and OS/2 machines. Samba has developed into a fully-fledged and rather complex product. Configure Samba with YaST, SWAT (a Web interface), or the configuration file.

29.1 Terminology

The following are some terms used in Samba documentation and in the YaST module.

SMB protocol
: Samba uses the SMB (server message block) protocol that is based on the NetBIOS services. Due to pressure from IBM, Microsoft released the protocol so other software manufacturers could establish connections to a Microsoft domain network. With Samba, the SMB protocol works on top of the TCP/IP protocol, so the TCP/IP protocol must be installed on all clients.

CIFS protocol
: CIFS (common Internet file system) protocol is another protocol supported by Samba. CIFS defines a standard remote file system access protocol for use over the network, enabling groups of users to work together and share documents across the network.

NetBIOS
: NetBIOS is a software interface (API) designed for communication between machines. Here, a name service is provided. It enables machines connected to the

network to reserve names for themselves. After reservation, these machines can be addressed by name. There is no central process that checks names. Any machine on the network can reserve as many names as it wants as long as the names are not already in use. The NetBIOS interface can now be implemented for different network architectures. An implementation that works relatively closely with network hardware is called NetBEUI, but this is often referred to as NetBIOS. Network protocols implemented with NetBIOS are IPX from Novell (NetBIOS via TCP/IP) and TCP/IP.

The NetBIOS names sent via TCP/IP have nothing in common with the names used in `/etc/hosts` or those defined by DNS. NetBIOS uses its own, completely independent naming convention. However, it is recommended to use names that correspond to DNS hostnames to make administration easier. This is the default used by Samba.

Samba server

Samba server is a server that provides SMB/CIFS services and NetBIOS over IP naming services to clients. For Linux, there are two daemons for Samba server: smnd for SMB/CIFS services and nmbd for naming services.

Samba client

Samba client is a system that uses Samba services from a Samba server over the SMB protocol. All common operating systems, such as Mac OS X, Windows, and OS/2, support the SMB protocol. The TCP/IP protocol must be installed on all computers. Samba provides a client for the different UNIX flavors. For Linux, there is a kernel module for SMB that allows the integration of SMB resources on the Linux system level. You do not need run any daemon for Samba client.

Shares

SMB servers provide hardware space to their clients by means of shares. Shares are printers and directories with their subdirectories on the server. It is exported by means of a name and can be accessed by its name. The share name can be set to any name—it does not have to be the name of the export directory. A printer is also assigned a name. Clients can access the printer by its name.

29.2 Installing a Samba Server

To install a Samba server, start YaST and select *Software* > *Software Management*. Choose *Filter* > *Patterns* and select *File Server*. Confirm the installation of the required packages to finish the installation process.

29.3 Starting and Stopping Samba

You can start or stop the Samba server automatically during boot or manually. Starting and stopping policy is a part of the YaST Samba server configuration described in Section 29.4.1, "Configuring a Samba Server with YaST" (page 471).

To stop or start running Samba services with YaST, use *System* > *System Services (Runlevel)*. From a command line, stop services required for Samba with `rcsmb stop && rcnmb stop` and start them with `rcnmb start && rcsmb start`.

29.4 Configuring a Samba Server

A Samba server in openSUSE® can be configured in two different ways: with YaST or manually. Manual configuration offers a higher level of detail, but lacks the convenience of the YaST GUI.

29.4.1 Configuring a Samba Server with YaST

To configure a Samba server, start YaST and select *Network Services* > *Samba Server*. When starting the module for the first time, the *Samba Server Installation* dialog starts, prompting you to make just a few basic decisions concerning administration of the server then at the end of the configuration prompts for the password of Samba root. For later starts, the *Samba Server Configuration* dialog appears.

The *Samba Server Installation* dialog consists of two steps:

Workgroup or Domain Name
: Select an existing name from *Workgroup or Domain Name* or enter a new one and click *Next*.

Samba Server Type
: In the next step, specify whether your server should act as PDC and click *Next*.

You can change all settings from *Samba Server Installation* later in the *Samba Server Configuration* dialog with the *Identity* tab.

Advanced Samba Configuration with YaST

During first start of Samba server module the *Samba Server Configuration* dialog appears directly after *Samba Server Installation* dialog. Use it to adjust your Samba server configuration.

After editing your configuration, click *Finish* to close the configuration.

Starting the Server

In the *Start Up* tab, configure the start of the Samba server. To start the service every time your system boots, select *During Boot*. To activate manual start, choose *Manually*. More information about starting a Samba server is provided in Section 29.3, "Starting and Stopping Samba" (page 471).

In this tab, you can also open ports in your firewall. To do so, select *Open Port in Firewall*. If you have multiple network interfaces, select the network interface for Samba services by clicking *Firewall Details*, selecting the interfaces, and clicking *OK*.

Shares

In the *Shares* tab, determine the Samba shares to activate. There are some predefined shares, like homes and printers. Use *Toggle Status* to switch between *Active* and *Inactive*. Click *Add* to add new shares and *Delete* to delete the selected share.

Identity

In the *Identity* tab, you can determine the domain with which the host is associated (*Base Settings*) and whether to use an alternative hostname in the network (*NetBIOS Host Name*). To set expert global settings or set user authentication, click *Advanced Settings*.

29.4.2 Web Administration with SWAT

An alternative tool for Samba server administration is SWAT (Samba Web Administration Tool). It provides a simple Web interface with which to configure the Samba server. To use SWAT, open `http://localhost:901` in a Web browser and log in as user `root`. If you do not have a special Samba root account, use the system `root` account.

NOTE: Activating SWAT

After Samba server installation, SWAT is not activated. To activate it, open *Network Services* > *Network Services (xinetd)* in YaST, enable the network services configuration, select *swat* from the table, and click *Toggle Status (On or Off)*.

29.4.3 Configuring the Server Manually

If you intend to use Samba as a server, install `samba`. The main configuration file of Samba is `/etc/samba/smb.conf`. This file can be divided into two logical parts. The `[global]` section contains the central and global settings. The `[share]` sections contain the individual file and printer shares. By means of this approach, details regarding the shares can be set differently or globally in the `[global]` section, which enhances the structural transparency of the configuration file.

The global Section

The following parameters of the `[global]` section need some adjustment to match the requirements of your network setup so other machines can access your Samba server via SMB in a Windows environment.

workgroup = TUX-NET

This line assigns the Samba server to a workgroup. Replace `TUX-NET` with an appropriate workgroup of your networking environment. Your Samba server appears under its DNS name unless this name has been assigned to any other machine in the network. If the DNS name is not available, set the server name using `netbiosname=MYNAME`. See `mansmb.conf` for more details about this parameter.

os level = 2

This parameter triggers whether your Samba server tries to become LMB (local master browser) for its workgroup. Choose a very low value to spare the existing Windows network from any disturbances caused by a misconfigured Samba server. More information about this important topic can be found in the files `BROWSING.txt` and `BROWSING-Config.txt` under the `textdocs` subdirectory of the package documentation.

If no other SMB server is present in your network (such as a Windows NT or 2000 server) and you want the Samba server to keep a list of all systems present in the local environment, set the `os level` to a higher value (for example, `65`). Your Samba server is then chosen as LMB for your local network.

When changing this setting, consider carefully how this could affect an existing Windows network environment. First test the changes in an isolated network or at a noncritical time of day.

wins support and wins server

To integrate your Samba server into an existing Windows network with an active WINS server, enable the `wins server` option and set its value to the IP address of that WINS server.

If your Windows machines are connected to separate subnets and should still be aware of each other, you need to set up a WINS server. To turn a Samba server into such a WINS server, set the option `wins support = Yes`. Make sure that only one Samba server of the network has this setting enabled. The options `wins server` and `wins support` must never be enabled at the same time in your `smb.conf` file.

Shares

The following examples illustrate how a CD-ROM drive and the user directories (homes) are made available to the SMB clients.

[cdrom]

To avoid having the CD-ROM drive accidentally made available, these lines are deactivated with comment marks (semicolons in this case). Remove the semicolons in the first column to share the CD-ROM drive with Samba.

Example 29.1 *A CD-ROM Share (deactivated)*

```
;[cdrom]
;       comment = Linux CD-ROM
;       path = /media/cdrom
;       locking = No
```

[cdrom] and comment

The entry [cdrom] is the name of the share that can be seen by all SMB clients on the network. An additional comment can be added to further describe the share.

path = /media/cdrom

path exports the directory /media/cdrom.

By means of a very restrictive default configuration, this kind of share is only made available to the users present on this system. If this share should be made available to everybody, add a line guest ok = yes to the configuration. This setting gives read permissions to anyone on the network. It is recommended to handle this parameter with great care. This applies even more to the use of this parameter in the [global] section.

[homes]

The [home] share is of special importance here. If the user has a valid account and password for the Linux file server and his own home directory, he can be connected to it.

***Example 29.2** homes Share*

```
[homes]
 comment = Home Directories
 valid users = %S
 browseable = No
 read only = No
 create mask = 0640
 directory mask = 0750
```

[homes]
: As long as there is no other share using the share name of the user connecting to the SMB server, a share is dynamically generated using the `[homes]` share directives. The resulting name of the share is the username.

`valid users = %S`
: `%S` is replaced with the concrete name of the share as soon as a connection has been successfully established. For a `[homes]` share, this is always the username. As a consequence, access rights to a user's share are restricted exclusively to the user.

`browseable = No`
: This setting makes the share invisible in the network environment.

`read only = No`
: By default, Samba prohibits write access to any exported share by means of the `read only = Yes` parameter. To make a share writable, set the value `read only = No`, which is synonymous with `writable = Yes`.

`create mask = 0640`
: Systems that are not based on MS Windows NT do not understand the concept of UNIX permissions, so they cannot assign permissions when creating a file. The parameter `create mask` defines the access permissions assigned to newly created files. This only applies to writable shares. In effect, this setting means the owner has read and write permissions and the members of the owner's primary group have read permissions. `valid users = %S` prevents read access even if the group has read permissions. For the group to have read or write access, deactivate the line `valid users = %S`.

Security Levels

To improve security, each share access can be protected with a password. SMB has three possible ways of checking the permissions:

Share Level Security (security = share)
: A password is firmly assigned to a share. Everyone who knows this password has access to that share.

User Level Security (security = user)
: This variation introduces the concept of the user to SMB. Each user must register with the server with his own password. After registration, the server can grant access to individual exported shares dependent on usernames.

Server Level Security (security = server):
: To its clients, Samba pretends to be working in user level mode. However, it passes all password queries to another user level mode server, which takes care of authentication. This setting expects an additional parameter (`password server`).

The selection of share, user, or server level security applies to the entire server. It is not possible to offer individual shares of a server configuration with share level security and others with user level security. However, you can run a separate Samba server for each configured IP address on a system.

More information about this subject can be found in the Samba HOWTO Collection. For multiple servers on one system, pay attention to the options `interfaces` and `bind interfaces only`.

29.5 Configuring Clients

Clients can only access the Samba server via TCP/IP. NetBEUI and NetBIOS via IPX cannot be used with Samba.

29.5.1 Configuring a Samba Client with YaST

Configure a Samba client to access resources (files or printers) on the Samba server. Enter the domain or workgroup in the dialog *Network Services* > *Windows Domain Membership*. Click *Browse* to display all available groups and domains, which can be selected with the mouse. If you activate *Also Use SMB Information for Linux Authentication*, the user authentication runs over the Samba server. After completing all settings, click *Finish* to finish the configuration.

29.5.2 Windows 9x and ME

Windows 9x and ME already have built-in support for TCP/IP. However, this is not installed as the default. To add TCP/IP, go to *Control Panel* > *System* and choose *Add* > *Protocols* > *TCP/IP from Microsoft*. After rebooting your Windows machine, find the Samba server by double-clicking the desktop icon for the network environment.

TIP

To use a printer on the Samba server, install the standard or Apple-PostScript printer driver from the corresponding Windows version. It is best to link this to the Linux printer queue, which accepts Postscript as an input format.

29.6 Samba as Login Server

In networks where predominantly Windows clients are found, it is often preferable that users may only register with a valid account and password. In a Windows-based network, this task is handled by a primary domain controller (PDC). You can use a Windows NT server configured as PDC, but this task can also be done with the help of a Samba server. The entries that must be made in the `[global]` section of `smb.conf` are shown in Example 29.3, "Global Section in smb.conf" (page 478).

Example 29.3 *Global Section in smb.conf*

```
[global]
workgroup = TUX-NET
domain logons = Yes
domain master = Yes
```

If encrypted passwords are used for verification purposes—this is the default setting with well-maintained MS Windows 9x installations, MS Windows NT 4.0 from service pack 3, and all later products—the Samba server must be able to handle these. The entry `encrypt passwords = yes` in the `[global]` section enables this (with Samba version 3, this is now the default). In addition, it is necessary to prepare user accounts and passwords in an encryption format that conforms with Windows. Do this with the command `smbpasswd -a name`. Create the domain account for the computers, required by the Windows NT domain concept, with the following commands:

***Example 29.4** Setting Up a Machine Account*

```
useradd hostname\$
smbpasswd -a -m hostname
```

With the `useradd` command, a dollar sign is added. The command `smbpasswd` inserts this automatically when the parameter `-m` is used. The commented configuration example (`/usr/share/doc/packages/Samba/examples/smb.conf.SuSE`) contains settings that automate this task.

***Example 29.5** Automated Setup of a Machine Account*

```
add machine script = /usr/sbin/useradd -g nogroup -c "NT Machine Account" \
-s /bin/false %m\$
```

To make sure that Samba can execute this script correctly, choose a Samba user with the required administrator permissions. To do so, select one user and add it to the `ntadmin` group. After that, all users belonging to this Linux group can be assigned `Domain Admin` status with the command:

```
net groupmap add ntgroup="Domain Admins" unixgroup=ntadmin
```

More information about this topic is provided in Chapter 12 of the Samba HOWTO Collection, found in `/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf`.

29.7 For More Information

Detailed Samba information is available in the digital documentation. Enter `apropos samba` at the command line to display some manual pages or just browse the `/usr/share/doc/packages/samba` directory if Samba documentation is installed for more online documentation and examples. Find a commented example configuration (`smb.conf.SuSE`) in the `examples` subdirectory.

The Samba HOWTO Collection provided by the Samba team includes a section about troubleshooting. In addition to that, Part V of the document provides a step-by-step guide to checking your configuration. You can find Samba HOWTO Collection in `/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf` after installing the package `samba-doc`.

Find detailed information about LDAP and migration from Windows NT or 2000 in `/usr/share/doc/packages/samba/examples/LDAP/smbldap-tools-*/doc`, where * is your smbldap-tools version.

The Apache HTTP Server 30

With a share of more than 70%, the Apache HTTP Server (Apache) is the world's most widely-used Web server according to the Survey from http://www.netcraft.com/. Apache, developed by the Apache Software Foundation (http://www.apache.org/), is available for most operating systems. openSUSE® includes Apache version 2.2. In this chapter, learn how to install, configure and set up a Web server; how to use SSL, CGI, and additional modules; and how to troubleshoot Apache.

30.1 Quick Start

With the help of this section, quickly set up and start Apache. time. You must be root to install and configure Apache.

30.1.1 Requirements

Make sure that the following requirements are met before trying to set up the Apache Web server:

1. The machine's network is configured properly. For more information about this topic, refer to Chapter 20, *Basic Networking* (page 299).

2. The machine's exact system time is maintained by synchronizing with a time server. This is necessary because parts of the HTTP protocol depend on the correct time. See Chapter 24, *Time Synchronization with NTP* (page 393) to learn more about this topic.

3. The latest security updates are installed. If in doubt, run a YaST Online Update.

4. The default Web server port (port 80) is opened in the firewall. For this, configure the SUSEFirewall2 to allow the service *HTTP Server* in the external zone. This can be done using YaST. Section 35.4.1, “Configuring the Firewall with YaST” (page 583) gives details.

30.1.2 Installation

Apache on openSUSE is not installed by default. To install it, start YaST and select *Software* > *Software Management*. Now choose *Filter* > *Patterns* and select *Web and LAMP Server* under *Server Functions*. Confirm the installation of the dependent packages to finish the installation process.

Apache is installed with a standard, predefined configuration that runs “out of the box”. The installation includes the multiprocessing module `apache2-prefork` as well the PHP5 module. Refer to Section 30.4, “Installing, Activating, and Configuring Modules” (page 499) for more information about modules.

30.1.3 Start

To start Apache and make sure that it is automatically started during boot, start YaST and select *System* > *System Services (Runlevel)*. Search for *apache2* and *Enable* the service. The Web server starts immediately. By saving your changes with *Finish*, the system is configured to automatically start Apache in runlevels 3 and 5 during boot. For more information about the runlevels in openSUSE and a description of the YaST runlevel editor, refer to Section 12.2.3, “Configuring System Services (Runlevel) with YaST” (page 190).

To start Apache using the shell, run `rcapache2 start`. To make sure that Apache is automatically started during boot in runlevels 3 and 5, use `chkconfig -a apache2`.

If you have not received error messages when starting Apache, the Web server should be running now. Start a browser and open `http://localhost/`. You should see an Apache test page starting with “If you can see this, it means that the installation of the Apache Web server software on this system was successful.” If you do not see this page, refer to Section 30.8, “Troubleshooting” (page 517).

Now that the Web server is running, you can add your own documents, adjust the configuration according to your needs, or add functionality by installing modules.

30.2 Configuring Apache

Apache in openSUSE can be configured in two different ways: with YaST or manually. Manual configuration offers a higher level of detail, but lacks the convenience of the YaST GUI.

IMPORTANT: Configuration Changes

Changes to most configuration values for Apache only take effect after Apache is restarted or reloaded. This happens automatically when using YaST and finishing the configuration with *Enabled* checked for the *HTTP Service*. Manual restart is described in Section 30.3, “Starting and Stopping Apache” (page 497). Most configuration changes only require a reload with `rcapache2 reload`.

30.2.1 Configuring Apache Manually

Configuring Apache manually involves editing the plain text configuration files as the user `root`.

Configuration Files

Apache configuration files can be found in two different locations:

- `/etc/sysconfig/apache2`
- `/etc/apache2/`

`/etc/sysconfig/apache2`

`/etc/sysconfig/apache2` controls some global settings of Apache, like modules to load, additional configuration files to include, flags with which the server should be started, and flags that should be added to the command line. Every configuration option in this file is extensively documented and therefore not mentioned here. For a general-purpose Web server, the settings in /etc/sysconfig/apache2 should be sufficient for any configuration needs.

/etc/apache2/

/etc/apache2/ hosts all configuration files for Apache. In the following, the purpose of each file is explained. Each file includes several configuration options (also referred to as *directives*). Every configuration option in these files is extensively documented and therefore not mentioned here.

The Apache configuration files are organized as follows:

```
/etc/apache2/
 |
 |- charset.conv
 |- conf.d/
 |   |
 |   |- *.conf
 |
 |- default-server.conf
 |- errors.conf
 |- httpd.conf
 |- listen.conf
 |- magic
 |- mime.types
 |- mod_*.conf
 |- server-tuning.conf
 |- ssl.*
 |- ssl-global.conf
 |- sysconfig.d
 |   |
 |   |- global.conf
 |   |- include.conf
 |   |- loadmodule.conf . .
 |
 |- uid.conf
 |- vhosts.d
 |   |- *.conf
```

Apache Configuration Files in /etc/apache2/

charset.conv
: Specifies which character sets to use for different languages. Do not edit.

conf.d/*.conf
: Configuration files added by other modules. These configuration files can be included into your virtual host configuration where needed. See vhosts.d/vhost

.template for examples. By doing so, you can provide different module sets for different virtual hosts.

default-server.conf
: Global configuration for all virtual hosts with reasonable defaults. Instead of changing the values, overwrite them with a virtual host configuration.

errors.conf
: Defines how Apache responds to errors. To customize these messages for all virtual hosts, edit this file. Otherwise overwrite these directives in your virtual host configurations.

httpd.conf
: The main Apache server configuration file. Avoid changing this file. It mainly contains include statements and global settings. Overwrite global settings in the respective configuration files listed here. Change host-specific settings (such as document root) in your virtual host configuration.

listen.conf
: Binds Apache to specific IP addresses and ports. Name-based virtual hosting (see Section "Name-Based Virtual Hosts" (page 487) is also configured here.

magic
: Data for the mime_magic module that helps Apache automatically determine the MIME type of an unknown file. Do not change.

mime.types
: MIME types known by the system (this actually is a link to /etc/mime.types). Do not edit. If you need to add MIME types not listed here, add them to mod_mime-defaults.conf.

mod_*.conf
: Configuration files for the modules that are installed by default. Refer to Section 30.4, "Installing, Activating, and Configuring Modules" (page 499) for details. Note that configuration files for optional modules reside in the directory conf.d.

server-tuning.conf
: Contains configuration directives for the different MPMs (see Section 30.4.4, "Multiprocessing Modules" (page 503)) as well as general configuration options that control Apache's performance. Properly test your Web server when making changes here.

`ssl-global.conf` and `ssl.*`
: Global SSL configuration and SSL certificate data. Refer to Section 30.6, “Setting Up a Secure Web Server with SSL” (page 509) for details.

`sysconfig.d/*.conf`
: Configuration files automatically generated from `/etc/sysconfig/apache2`. Do not change any of these files—edit `/etc/sysconfig/apache2` instead. Put no other configuration files in this directory.

`uid.conf`
: Specifies under which user and group ID Apache runs. Do not change.

`vhosts.d/*.conf`
: Your virtual host configuration should go here.The directory contains template files for virtual hosts with and without SSL. Every file in this directory ending in `.conf` is automatically included in the Apache configuration. Refer to Section “Virtual Host Configuration” (page 486) for details.

Virtual Host Configuration

The term *virtual host* refers to Apache's ability to serve multiple URIs (universal resource identifiers) from the same physical machine. This means that several domains, such as www.example.com and www.example.net, are run by a single Web server on one physical machine.

It is common practice to use virtual hosts to save administrative effort (only a single Web server needs to be maintained) and hardware expenses (each domain does not require a dedicated server). Virtual hosts can be name based, IP based, or port based.

Virtual hosts can be configured via YaST (see Section “Virtual Hosts” (page 493)) or by manually editing a configuration file. By default, Apache in openSUSE is prepared for one configuration file per virtual host in `/etc/apache2/vhosts.d/`. All files in this directory with the extension `.conf` are automatically included to the configuration. A basic template for a virtual host is provided in this directory (`vhost.template` or `vhost-ssl.template` for a virtual host with SSL support).

TIP: Always Create a Virtual Host Configuration

It is recommended to always create a virtual host configuration file, even if your Web server only hosts one domain. In doing so, you not only have the domain-specific configuration in one file, but you can always fall back to a working basic configuration by simply moving, deleting, or renaming the configuration file for the virtual host. For the same reason, you should also create separate configuration files for each virtual host.

The `<VirtualHost></VirtualHost>` block holds the information that applies to a particular domain. When Apache receives a client request for a defined virtual host, it uses the directives enclosed in this section. Almost all directives can be used in a virtual host context. See `http://httpd.apache.org/docs/2.2/mod/quickreference.html` for further information about Apache's configuration directives.

Name-Based Virtual Hosts

With name-based virtual hosts, more than one Web site is served per IP address. Apache uses the host field in the HTTP header sent by the client to connect the request to a matching `ServerName` entry of one of the virtual host declarations. If no matching `ServerName` is found, the first specified virtual host is used as a default.

The directive `NameVirtualHost` tells Apache on which IP address and, optionally, which port to listen for requests by clients containing the domain name in the HTTP header. This option is configured in the configuration file `/etc/apache2/listen.conf`.

The first argument can be a fully qualified domain name, but it is recommended to use the IP address. The second argument is the port and is optional. By default, port 80 is used and is configured via the `Listen` directive.

The wild card `*` can be used for both the IP address and the port number to receive requests on all interfaces. IPv6 addresses must be enclosed in square brackets.

Example 30.1 *Variations of Name-Based VirtualHost Entries*

```
# NameVirtualHost IP-address[:Port]
NameVirtualHost 192.168.3.100:80
NameVirtualHost 192.168.3.100
NameVirtualHost *:80
NameVirtualHost *
NameVirtualHost [2002:c0a8:364::]:80
```

The opening `VirtualHost` tag takes the IP address (or fully qualified domain name) previously declared with the `NameVirtualHost` as an argument in a name-based virtual host configuration. A port number previously declared with the `NameVirtualHost` directive is optional.

The wild card * is also allowed as a substitute for the IP address. This syntax is only valid in combination with the wild card usage in `NameVirtualHost *`. When using IPv6 addresses, the address must be included in square brackets.

Example 30.2 *Name-Based VirtualHost Directives*

```
<VirtualHost 192.168.3.100:80>
  ...
</VirtualHost>

<VirtualHost 192.168.3.100>
  ...
</VirtualHost>

<VirtualHost *:80>
  ...
</VirtualHost>

<VirtualHost *>
  ...
</VirtualHost>

<VirtualHost [2002:c0a8:364::]>
  ...
</VirtualHost>
```

IP-Based Virtual Hosts

This alternative virtual host configuration requires the setup of multiple IPs for a machine. One instance of Apache hosts several domains, each of which is assigned a different IP.

The physical server must have one IP address for each IP-based virtual host. If the machine does not have multiple network cards, virtual network interfaces (IP aliasing) can also be used.

The following example shows Apache running on a machine with the IP `192.168.3.100`, hosting two domains on the additional IPs `192.168.3.101` and `192.168.3.102`. A separate `VirtualHost` block is needed for every virtual server.

Example 30.3 *IP-Based VirtualHost Directives*

```
<VirtualHost 192.168.3.101>
  ...
</VirtualHost>

<VirtualHost 192.168.3.102>
  ...
</VirtualHost>
```

Here, `VirtualHost` directives are only specified for interfaces other than `192.168.3.100`. When a `Listen` directive is also configured for `192.168.3.100`, a separate IP-based virtual host must be created to answer HTTP requests to that interface—otherwise the directives found in the default server configuration (`/etc/apache2/default-server.conf`) are applied.

Basic Virtual Host Configuration

At least the following directives should be present in each virtual host configuration in order to set up a virtual host. See `/etc/apache2/vhosts.d/vhost.template` for more options.

`ServerName`
: The fully qualified domain name under which the host should be addressed.

`DocumentRoot`
: Path to the directory from which Apache should serve files for this host. For security reasons, access to the entire file system is forbidden by default, so you must explicitly unlock this directory within a `Directory` container.

`ServerAdmin`
: E-mail address of the server administrator. This address is, for example, shown on error pages Apache creates.

`ErrorLog`
: The error log file for this virtual host. Although it is not necessary to create separate error log files for each virtual host, it is common practice to do so, because it makes debugging of errors much easier. `/var/log/apache2/` is the default directory where Apache's log files should be kept.

`CustomLog`
: The access log file for this virtual host. Although it is not necessary to create separate access log files for each virtual host, it is common practice to do so, because it allows separate analysis of access statistics for each host. `/var/log/apache2/` is the default directory where Apache's log files should be kept.

As mentioned above, access to the whole file system is forbidden by default for security reasons. Therefore, explicitly unlock the directories in which you have placed the files Apache should serve—for example the `DocumentRoot`:

```
<Directory "/srv/www/www.example.com/htdocs">
  Order allow,deny
  Allow from all
</Directory>
```

The complete configuration file looks like this:

***Example 30.4** Basic VirtualHost Configuration*

```
<VirtualHost 192.168.3.100>
  ServerName www.example.com;
  DocumentRoot /srv/www/www.example.com/htdocs
  ServerAdmin webmaster@example.com
  ErrorLog /var/log/apache2/www.example.com_log
  CustomLog /var/log/apache2/www.example.com-access_log common
  <Directory "/srv/www/www.example.com/htdocs">
    Order allow,deny
    Allow from all
  </Directory>
</VirtualHost>
```

30.2.2 Configuring Apache with YaST

To configure your Web server with YaST, start YaST and select *Network Services* > *HTTP Server*. When starting the module for the first time, the `HTTP Server Wizard` starts, prompting you to make just a few basic decisions concerning administration of the server. After having finished the wizard, the dialog in Section “HTTP Server Configuration” (page 495) starts every time you call the *HTTP Server* module.

HTTP Server Wizard

The HTTP Server Wizard consists of five steps. In the last step of the dialog, you are given the opportunity to enter the expert configuration mode to make even more specific settings.

Network Device Selection

Here, specify the network interfaces and ports Apache uses to listen for incoming requests. You can select any combination of existing network interfaces and their respective IP addresses. Ports from all three ranges (well-known ports, registered ports, and dynamic or private ports) that are not reserved by other services can be used. The default setting is to listen on all network interfaces (IP addresses) on port 80.

Check *Open Firewall for Selected Ports* to open the ports in the firewall that the Web server listens on. This is necessary to make the Web server available on the network, which can be a LAN, WAN, or the public Internet. Keeping the port closed is only useful in test situations where no external access to the Web server is necessary.

Click *Next* to continue with configuration.

Modules

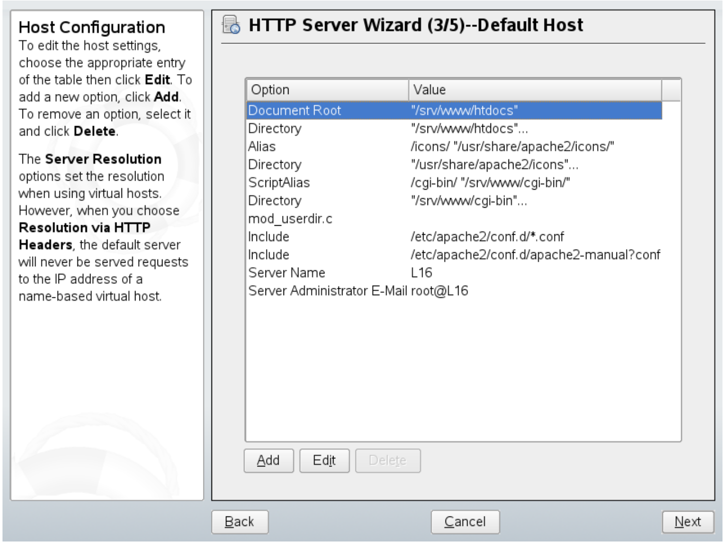
The *Modules* configuration option allows for the activation or deactivation of the script languages, the web server should support. For the activation or deactivation of other modules, refer to Section “Server Modules” (page 496). Click *Next* to advance to the next dialog.

Default Host

This option pertains to the default Web server. As explained in Section “Virtual Host Configuration” (page 486), Apache can serve multiple virtual hosts from a single physical machine. The first declared virtual host in the configuration file is commonly referred to as the *default host*. Each virtual host inherits the default host's configuration.

To edit the host settings (also called *directives*), choose the appropriate entry in the table then click *Edit*. To add new directives, click *Add*. To delete a directive, select it and click *Delete*.

Figure 30.1 *HTTP Server Wizard: Default Host*



Host Configuration

To edit the host settings, choose the appropriate entry of the table then click **Edit**. To add a new option, click **Add**. To remove an option, select it and click **Delete**.

The **Server Resolution** options set the resolution when using virtual hosts. However, when you choose **Resolution via HTTP Headers**, the default server will never be served requests to the IP address of a name-based virtual host.

HTTP Server Wizard (3/5)--Default Host

Option	Value
Document Root	"/srv/www/htdocs"
Directory	"/srv/www/htdocs"...
Alias	/icons/ "/usr/share/apache2/icons/"
Directory	"/usr/share/apache2/icons"...
ScriptAlias	/cgi-bin/ "/srv/www/cgi-bin/"
Directory	"/srv/www/cgi-bin"...
mod_userdir.c	
Include	/etc/apache2/conf.d/*.conf
Include	/etc/apache2/conf.d/apache2-manual?conf
Server Name	L16
Server Administrator E-Mail	root@L16

Here is list of the default settings of the server:

`Document Root`
: Path to the directory from which Apache serves files for this host. `/srv/www/htdocs` is the default location.

`Alias`
: With the help of `Alias` directives, URLs can be mapped to physical file system locations. This means that a certain path even outside the `Document Root` in the file system can be accessed via a URL aliasing that path.

  The default openSUSE `Alias /icons` points to `/usr/share/apache2/icons` for the Apache icons displayed in the directory index view.

`ScriptAlias`
: Similar to the `Alias` directive, the `ScriptAlias` directive maps a URL to a file system location. The difference is that `ScriptAlias` designates the target directory as a CGI location, meaning that CGI scripts should be executed in that location.

`Directory`
: With the `Directory` setting, you can enclose a group of configuration options that will only apply to the specified directory.

  Access and display options for the directories `/usr/share/apache2/icons` and `/srv/www/cgi-bin` are configured here. It should not be necessary to change the defaults.

`Include`
: With include, additional configuration files can be specified. Two `Include` directives are already preconfigured: `/etc/apache2/conf.d/` is the directory containing the configuration files that come with external modules. With this directive, all files in this directory ending in `.conf` are included. With the second directive, `/etc/apache2/conf.d/apache2-manual?conf`, the `apache2-manual` configuration file is included.

`Server Name`
: This specifies the default URL used by clients to contact the Web server. Use a fully qualified domain name (FQDN) to reach the Web server at `http://`*`FQDN`*`/` or its IP address. You cannot choose an arbitrary name here—the server must be "known" under this name.

`Server Administrator E-Mail`
: E-mail address of the server administrator. This address is, for example, shown on error pages Apache creates.

After finishing with the *Default Host* step, click *Next* to continue with the configuration.

Virtual Hosts

In this step, the wizard displays a list of already configured virtual hosts (see Section "Virtual Host Configuration" (page 486)). If you have not made manual changes prior to starting the YaST HTTP wizard, no virtual host is present.

To add a host, click *Add* to open a dialog in which to enter basic information about the host. *Server Identification* includes the server name, server contents root (`DocumentRoot`), and administrator e-mail. *Server Resolution* is used to determine how a host is identified (name based or IP based). Specify the name or IP address with *Change Virtual Host ID*

Clicking *Next* advances to the second part of the virtual host configuration dialog.

In part two of the virtual host configuration you can specify whether to enable CGI scripts and which directory to use for these scripts. It is also possible to enable SSL. If you do so, you must specify the path to the certificate as well. See Section 30.6.2, "Configuring Apache with SSL" (page 514) for details on SSL and certificates. With the *Directory Index* option, you can specify which file to display when the client requests a directory (by default, index.html). Add one or more filenames (space-separated) if you want to change this. With *Enable Public HTML*, the content of the users public directories (`~user/public_html/`) is made available on the server under `http://www.example.com/~user`.

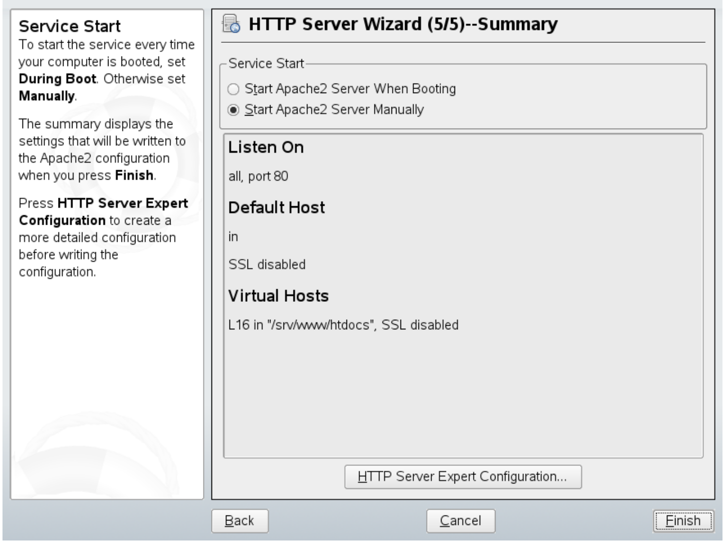
IMPORTANT: Creating Virtual Hosts

It is not possible to add virtual hosts at will. If using name-based virtual hosts, each hostname must be resolved on the network. If using IP-based virtual hosts, you can assign only one host to each IP address available.

Summary

This is the final step of the wizard. Here, determine how and when the Apache server is started: when booting or manually. Also see a short summary of the configuration made so far. If you are satisfied with your settings, click *Finish* to complete configuration. If you want to change something, click *Back* until you have reached the desired dialog. Clicking *HTTP Server Expert Configuration* opens the dialog described in Section "HTTP Server Configuration" (page 495).

Figure 30.2 *HTTP Server Wizard: Summary*



Service Start

To start the service every time your computer is booted, set **During Boot**. Otherwise set **Manually**.

The summary displays the settings that will be written to the Apache2 configuration when you press **Finish**.

Press **HTTP Server Expert Configuration** to create a more detailed configuration before writing the configuration.

HTTP Server Wizard (5/5)--Summary

Service Start

○ Start Apache2 Server When Booting
● Start Apache2 Server Manually

Listen On

all, port 80

Default Host

in

SSL disabled

Virtual Hosts

L16 in "/srv/www/htdocs", SSL disabled

HTTP Server Configuration

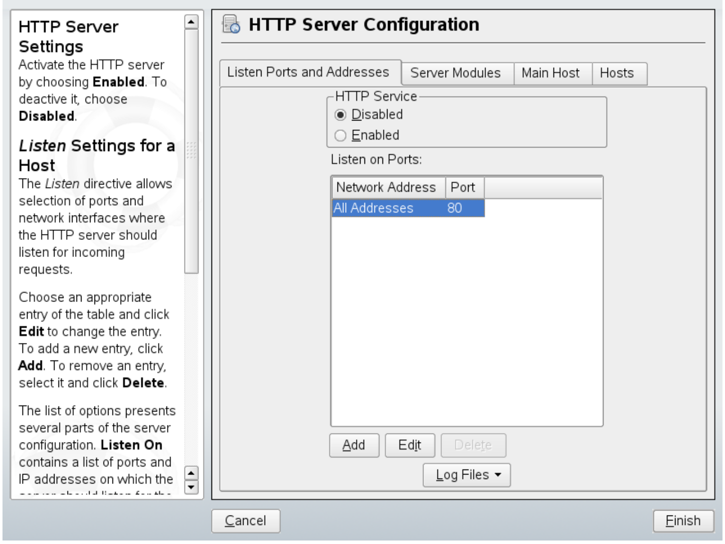
The *HTTP Server Configuration* dialog also lets you make even more adjustments to the configuration than the wizard (which only runs if you configure your Web server for the first time). It consists of four tabs described in the following. No configuration option you change here is effective immediately—you always must confirm your changes with *Finish* to make them effective. Clicking *Cancel* leaves the configuration module and discards your changes.

Listen Ports and Addresses

In *HTTP Service*, select whether Apache should be running (*Enabled*) or stopped (*Disabled*). In *Listen on Ports*, *Add*, *Edit*, or *Delete* addresses and ports on which the server should be available. The default is to listen on all interfaces on port 80. You should always check *Open Firewall on Selected Ports*, because otherwise the Web server is not reachable from the outside. Keeping the port closed is only useful in test situations where no external access to the Web server is necessary.

With *Log Files*, watch either the access log or the error log. This is useful if you want to test your configuration. The log file opens in a separate window from which you can also restart or reload the Web server (see Section 30.3, “Starting and Stopping Apache” (page 497) for details). These commands are effective immediately.

Figure 30.3 *HTTP Server Configuration: Listen Ports and Addresses*



HTTP Server Settings

Activate the HTTP server by choosing **Enabled**. To deactive it, choose **Disabled**.

Listen Settings for a Host

The *Listen* directive allows selection of ports and network interfaces where the HTTP server should listen for incoming requests.

Choose an appropriate entry of the table and click **Edit** to change the entry. To add a new entry, click **Add**. To remove an entry, select it and click **Delete**.

The list of options presents several parts of the server configuration. **Listen On** contains a list of ports and IP addresses on which the

HTTP Server Configuration

HTTP Service

● Disabled

○ Enabled

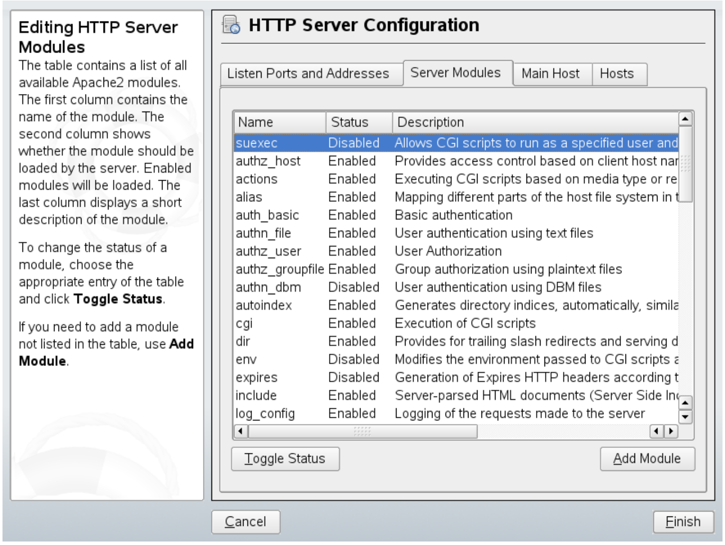
Listen on Ports:

Network Address	Port
All Addresses	80

Server Modules

You can change the status (enabled or disabled) of Apache2 modules by clicking *Toggle Status*. Click *Add Module* to add a new module that is already installed but not yet listed. Learn more about modules in Section 30.4, “Installing, Activating, and Configuring Modules” (page 499).

Figure 30.4 *HTTP Server Configuration: Server Modules*



Editing HTTP Server Modules

The table contains a list of all available Apache2 modules. The first column contains the name of the module. The second column shows whether the module should be loaded by the server. Enabled modules will be loaded. The last column displays a short description of the module.

To change the status of a module, choose the appropriate entry of the table and click **Toggle Status**.

If you need to add a module not listed in the table, use **Add Module**.

Name	Status	Description
suexec	Disabled	Allows CGI scripts to run as a specified user and
authz_host	Enabled	Provides access control based on client host nar
actions	Enabled	Executing CGI scripts based on media type or re
alias	Enabled	Mapping different parts of the host file system in t
auth_basic	Enabled	Basic authentication
authn_file	Enabled	User authentication using text files
authz_user	Enabled	User Authorization
authz_groupfile	Enabled	Group authorization using plaintext files
authn_dbm	Disabled	User authentication using DBM files
autoindex	Enabled	Generates directory indices, automatically, simila
cgi	Enabled	Execution of CGI scripts
dir	Enabled	Provides for trailing slash redirects and serving d
env	Disabled	Modifies the environment passed to CGI scripts a
expires	Disabled	Generation of Expires HTTP headers according t
include	Enabled	Server-parsed HTML documents (Server Side Inc
log_config	Enabled	Logging of the requests made to the server

Main Host or Hosts

These dialogs are identical to the ones already described. Refer to Section "Default Host" (page 491) and Section "Virtual Hosts" (page 493).

30.3 Starting and Stopping Apache

If configured with YaST (see Section 30.2.2, "Configuring Apache with YaST" (page 490)), Apache is started at boot time in runlevels 3 and 5 and stopped in runlevels 0, 1, 2, and 6. You can change this behavior using YaST's runlevel editor or the command line tool `chkconfig`.

To start, stop, or manipulate Apache on a running system, use the init script `/usr/sbin/rcapache2` (refer to Section 12.2.2, "Init Scripts" (page 186) for a general information about init scripts.). The `rcapache2` command takes the following parameters:

`start`
: Starts Apache if it is not already running.

`startssl`

Starts Apache with SSL support if it is not already running. For more information about SSL support, refer to Section 30.6, “Setting Up a Secure Web Server with SSL” (page 509).

`stop`

Stops Apache by terminating the parent process.

`restart`

Stops then restarts Apache. Starts the Web server if it was not running before.

`try-restart`

Stops then restarts Apache only if it has been running before.

`reload or graceful`

Stops the Web server by advising all forked Apache processes to first finish their requests before shutting down. As each process dies, it is replaced by a newly started one, resulting in complete “restart” of Apache.

TIP

`rcapache2 reload` is the preferred method of restarting Apache in production environments, for example, to activate a change in the configuration, because it allows all clients to be served without causing connection break-offs.

`configtest`

Checks the syntax of the configuration files without affecting a running Web server. Because this check is forced every time the server is started, reloaded, or restarted, it is usually not necessary to run the test explicitly (if a configuration error is found, the Web server is not started, reloaded, or restarted).

`probe`

Probes for the necessity of a reload (checks whether the configuration has changed) and suggests the required arguments for the `rcapache2` command.

`server-status and full-server-status`

Dumps a short or full status screen, respectively. Requires either lynx or w3m installed as well as the module mod_status enabled. In addition to that, `status` must be added to `APACHE_SERVER_FLAGS` in the file `/etc/sysconfig/apache2`.

TIP: Additional Flags

If you specify additional flags to the `rcapache2`, these are passed through to the Web server.

30.4 Installing, Activating, and Configuring Modules

The Apache software is built in a modular fashion: all functionality except some core tasks is handled by modules. This has progressed so far that even HTTP is processed by a module (http_core).

Apache modules can be compiled into the Apache binary at build time or dynamically loaded at runtime. Refer to Section 30.4.2, “Activation and Deactivation” (page 500) for details of how to load modules dynamically.

Apache modules can be divided into four different categories:

Base Modules
: Base modules are compiled into Apache by default. Apache in SUSE Linux has only mod_so (needed to load other modules) and http_core compiled in. All others are available as shared objects: rather than being included in the server binary itself, they can be included at runtime.

Extension Modules
: In general, modules labeled as extensions are included in the Apache software package, but are usually not compiled into the server statically. In openSUSE, they are available as shared objects that can be loaded into Apache at runtime.

External Modules
: Modules labeled external are not included in the official Apache distribution. openSUSE provides several of them readily available for use.

Multiprocessing Modules
: MPMs are responsible for accepting and handling requests to the Web server, representing the core of the Web server software.

30.4.1 Module Installation

If you have followed the default way of installing Apache (described in Section 30.1.2, "Installation" (page 482)), it is installed with all base and extension modules, the multiprocessing module Prefork MPM, and the external modules mod_php5 and mod_python.

You can install additional external modules by starting YaST and choosing *Software* > *Software Management*. Now choose *Filter* > *Search* and search for *apache*. Among other packages, the result list contains all available external Apache modules.

30.4.2 Activation and Deactivation

Using YaST, you can activate or deactivate the script language modules (PHP5, Perl, Python, and Ruby) with the module configuration described in Section "HTTP Server Wizard" (page 491). All other modules can be enabled or disabled as described in Section "Server Modules" (page 496).

If you prefer to activate or deactivate the modules manually, use the commands `a2enmod` *`mod_foo`* or `a2dismod` *`mod_foo`*, respectively. `a2enmod -l` outputs a list of all currently active modules.

IMPORTANT: Including Configuration Files for External Modules

If you have activated external modules manually, make sure to load their configuration files in all virtual host configurations. Configuration files for external modules are located under `/etc/apache2/conf.d/` and are not loaded by default. If you need the same modules on each virtual host, you can include `*.conf` from this directory. Otherwise include individual files. See `/etc/apache2/vhost.d/vhost.template` for examples.

30.4.3 Base and Extension Modules

All base and extension modules are described in detail in the Apache documentation. Only a brief description of the most important modules is available here. Refer to `http://httpd.apache.org/docs/2.2/mod/` to learn details about each module.

mod_actions

Provides methods to execute a script whenever a certain MIME type (such as `application/pdf`), a file with a specific extension (like `.rpm`), or a certain request method (such as `GET`) is requested. This module is enabled by default.

mod_alias

Provides `Alias` and `Redirect` directives with which you can map a URl to a specific directory (`Alias`) or redirect a requested URL to another location. This module is enabled by default.

mod_auth*

The authentication modules provide different authentication methods: basic authentication with mod_auth_basic or digest authentication with mod_auth_digest. Digest authentication in Apache 2.2 is considered experimental.

mod_auth_basic and mod_auth_digest must be combined with an authentication provider module, mod_authn_* (for example, mod_authn_file for text file–based authentication) and with an authorization module mod_authz_* (for example, mod_authz_user for user authorization).

More information about this topic is available in the “Authentication HOWTO” at `http://httpd.apache.org/docs/2.2/howto/auth.html`

mod_autoindex

Autoindex generates directory listings when no index file (for example, `index.html`) is present. The look and feel of these indexes is configurable. This module is enabled by default. However, directory listings are disabled by default via the `Options` directive—overwrite this setting in your virtual host configuration. The default configuration file for this module is located at `/etc/apache2/mod_autoindex-defaults.conf`.

mod_cgi

mod_cgi is needed to execute CGI scripts. This module is enabled by default.

mod_deflate

Using this module, Apache can be configured to compress given file types on the fly before delivering them.

mod_dir

mod_dir provides the `DirectoryIndex` directive with which you can configure which files are automatically delivered when a directory is requested (`index`

.html by default). It also provides an automatic redirect to the correct URl when a directory request does not contain a trailing slash. This module is enabled by default.

mod_env
: Controls the environment that is passed to CGI scripts or SSI pages. Environment variables can be set or unset or passed from the shell that invoked the httpd process. This module is enabled by default.

mod_expires
: With mod_expires, you can control how often proxy and browser caches refresh your documents by sending an Expires header. This module is enabled by default.

mod_include
: mod_include lets you use Server Side Includes (SSI), which provide a basic functionality to generate HTML pages dynamically. This module is enabled by default.

mod_info
: Provides a comprehensive overview of the server configuration under http://localhost/server-info/. For security reasons, you should always limit access to this URL. By default only localhost is allowed to access this URL. mod_info is configured at /etc/apache2/mod_info.conf

mod_log_config
: With this module, you can configure the looks of the Apache log files. This module is enabled by default.

mod_mime
: The mime module takes care that a file is delivered with the correct MIME header based on the filename's extension (for example text/html for HTML documents). This module is enabled by default.

mod_negotiation
: Necessary for content negotiation. See http://httpd.apache.org/docs/2.2/content-negotiation.html for more information. This module is enabled by default.

mod_rewrite
: Provides the functionality of mod_alias, but offers more features and flexibility. With mod_rewrite, you can redirect URLs based on multiple rules, request headers, and more.

mod_setenvif
: Sets environment variables based on details of the client's request, such as the browser string the client sends, or the client's IP address. This module is enabled by default.

mod_speling
: mod_speling attempts to automatically correct typographical errors in URLs, such as capitalization errors.

mod_ssl
: Enables encrypted connections between Web server and clients. See Section 30.6, “Setting Up a Secure Web Server with SSL” (page 509) for details. This module is enabled by default.

mod_status
: Provides information on server activity and performance under http://localhost/server-status/. For security reasons, you should always limit access to this URL. By default, only `localhost` is allowed to access this URl. mod_status is configured at `/etc/apache2/mod_status.conf`

mod_suexec
: mod_suexec lets you run CGI scripts under a different user and group. This module is enabled by default.

mod_userdir
: Enables user-specific directories available under `~`*`user`*`/`. The `UserDir` directive must be specified in the configuration. This module is enabled by default.

30.4.4 Multiprocessing Modules

openSUSE provides two different multiprocessing modules (MPMs) for use with Apache.

Prefork MPM

The prefork MPM implements a nonthreaded, preforking Web server. It makes the Web server behave similarly to Apache version 1.x in that it isolates each request and handles it by forking a separate child process. Thus problematic requests cannot affect others, avoiding a lockup of the Web server.

While providing stability with this process-based approach, the prefork MPM consumes more system resources than its counterpart, the worker MPM. The prefork MPM is considered the default MPM for Unix-based operating systems.

IMPORTANT: MPMs in This Document

This document assumes Apache is used with the prefork MPM.

Worker MPM

The worker MPM provides a multithreaded Web server. A thread is a "lighter" form of a process. The advantage of a thread over a process is its lower resource consumption. Instead of only forking child processes, the worker MPM serves requests by using threads with server processes. The preforked child processes are multithreaded. This approach makes Apache perform better by consuming fewer system resources than the prefork MPM.

One major disadvantage is the stability of the worker MPM: if a thread becomes corrupt, all threads of a process can be affected. In the worst case, this may result in a server crash. Especially when using the Common Gateway Interface (CGI) with Apache under heavy load, internal server errors might occur due to threads unable to communicate with system resources. Another argument against using the worker MPM with Apache is that not all available Apache modules are thread-safe and thus cannot be used in conjunction with the worker MPM.

WARNING: Using PHP Modules with MPMs

Not all available PHP modules are thread-safe. Using the worker MPM with mod_php is strongly discouraged.

30.4.5 External Modules

Find a list of all external modules shipped with openSUSE here. Find the module's documentation in the listed directory.

mod-apparmor
: Adds support to Apache to provide Novell AppArmor confinement to individual CGI scripts handled by modules like mod_php5 and mod_perl.

Package Name: `apache2-mod_apparmor`
More Information: *Novell AppArmor Administration Guide* (↑Novell AppArmor Administration Guide)

mod_fcgid
: mod_fcgid is a binary-compatible alternative to mod_fastcgi. It is a language-independent, scalable, and open extension to CGI that provides high performance without the limitations of server-specific APIs. mod_fcgid applications are very fast because they are persistent. There is no per-request start-up and initialization overhead.

Package Name: `apache2-mod_fcgid`
Configuration File: `/etc/apache2/conf.d/mod_fcgid.conf`
More Information: `/usr/share/doc/packages/apache2-mod_fastcgi`

mod_perl
: mod_perl enables you to run Perl scripts in an embedded interpreter. The persistent interpreter embedded in the server avoids the overhead of starting an external interpreter and the penalty of Perl start-up time.

Package Name: `apache2-mod_perl`
Configuration File: `/etc/apache2/conf.d/mod_perl.conf`
More Information: `/usr/share/doc/packages/apache2-mod_perl`

mod_php5
: PHP is a server-side, cross-platform HTML embedded scripting language.

Package Name: `apache2-mod_php5`
Configuration File: `/etc/apache2/conf.d/php5.conf`
More Information: `/usr/share/doc/packages/apache2-mod_php5`

mod_python
: mod_python allows embedding Python within the Apache HTTP server for a considerable boost in performance and added flexibility in designing Web-based applications.

Package Name: `apache2-mod_python`
More Information: `/usr/share/doc/packages/apache2-mod_python`

mod_jk-ap20
: This module provides connectors between Apache and a Tomcat Servlet Container.

Package Name: mod_jk-ap20
More Information: /usr/share/doc/packages/mod_jk-ap20

30.4.6 Compilation

Apache can be extended by advanced users by writing custom modules. To develop modules for Apache or compile third-party modules, the package apache2-devel is required along with the corresponding development tools. apache2-devel also contains the apxs2 tools, which are necessary for compiling additional modules for Apache.

apxs2 enables the compilation and installation of modules from source code (including the required changes to the configuration files), which creates *dynamic shared objects* (DSOs) that can be loaded into Apache at runtime.

The apxs2 binaries are located under /usr/sbin:

- /usr/sbin/apxs2—suitable for building an extension module that works with any MPM. The installation location is /usr/lib/apache2.

- /usr/sbin/apxs2-prefork—suitable for prefork MPM modules. The installation location is /usr/lib/apache2-prefork.

- /usr/sbin/apxs2-worker—suitable for worker MPM modules.

apxs2 installs modules so they can be used for all MPMs. The other two programs install modules so they can only be used for the respective MPMs. apxs2 installs modules in /usr/lib/apache2, apxs2-prefork and apxs2-worker installs modules in /usr/lib/apache2-prefork or /usr/lib/apache2-worker.

Install and activate a module from source code with the commands `cd /path/to/module/source; apxs2 -cia` *`mod_foo`*`.c` (-c compiles the module, -i installs it, and -a activates it). Other options of apxs2 are described in the apxs2(1) man page.

30.5 Getting CGI Scripts to Work

Apache's Common Gateway Interface (CGI) lets you create dynamic content with programs or scripts usually referred to as CGI scripts. CGI scripts can be written in any programming language. Usually, script languages such as Perl or PHP are used.

To enable Apache to deliver content created by CGI scripts, mod_cgi needs to be activated. mod_alias is also needed. Both modules are enabled by default. Refer to Section 30.4.2, "Activation and Deactivation" (page 500) for details on activating modules.

WARNING: CGI Security

Allowing the server to execute CGI scripts is a potential security hole. Refer to Section 30.7, "Avoiding Security Problems" (page 515) for additional information.

30.5.1 Apache Configuration

In openSUSE, the execution of CGI scripts is only allowed in the directory `/srv/www/cgi-bin/`. This location is already configured to execute CGI scripts. If you have created a virtual host configuration (see Section "Virtual Host Configuration" (page 486)) and want to place your scripts in a host-specific directory, you must unlock and configure this directory.

Example 30.5 *VirtualHost CGI Configuration*

```
ScriptAlias /cgi-bin/ "/srv/www/www.example.com/cgi-bin/"❶

<Directory "/srv/www/www.example.com/cgi-bin/">
 Options +ExecCGI❷
 AddHandler cgi-script .cgi .pl❸
 Order allow,deny❹
 Allow from all
</Directory>
```

❶ Tells Apache to handle all files within this directory as CGI scripts.

❷ Enables CGI script execution

❸ Tells the server to treat files with the extensions .pl and .cgi as CGI scripts. Adjust according to your needs.

❹ The `Order` and `Allow` directives control the default access state and the order in which Allow and Deny directives are evaluated. In this case “deny” statements are evaluated before “allow” statements and access from everywhere is enabled.

30.5.2 Running an Example Script

CGI programming differs from "regular" programming in that the CGI programs and scripts must be preceded by a MIME-Type header such as `Content-type: text/html`. This header is sent to the client, so it understands what kind of content it receives. Secondly, the script's output must be something the client, usually a Web browser, understands—HTML in most cases or plain text or images, for example.

A simple test script available under `/usr/share/doc/packages/apache2/test-cgi` is part of the Apache package. It outputs the content of some environment variables as plain text. Copy this script to either `/srv/www/cgi-bin/` or the script directory of your virtual host (`/srv/www/www.example.com/cgi-bin/`) and name it `test.cgi`.

Files accessible by the Web server should be owned by to the user `root` (see Section 30.7, “Avoiding Security Problems” (page 515) for additional information). Because the Web server runs with a different user, the CGI scripts must be world-executable and world-readable. Change into the CGI directory and use the command `chmod 755 test.cgi` to apply the proper permissions.

Now call `http://localhost/cgi-bin/test.cgi` or `http://www.example.com/cgi-bin/test.cgi`. You should see the “CGI/1.0 test script report”.

30.5.3 Troubleshooting

If you do not see the output of the test program but an error message instead, check the following:

CGI Troubleshooting

- Have you reloaded the server after having changed the configuration? Check with `rcapache2 probe`.

- If you have configured your custom CGI directory, is it configured properly? If in doubt, try the script within the default CGI directory `/srv/www/cgi-bin/` and call it with `http://localhost/cgi-bin/test.cgi`.

- Are the file permissions correct? Change into the CGI directory and execute the `ls -l test.cgi`. Its output should start with

  ```
  -rwxr-xr-x  1 root root
  ```

- Make sure that the script does not contain programming errors. If you have not changed test.cgi, this should not be the case, but if you are using your own programs, always make sure that they do not contain programming errors.

30.6 Setting Up a Secure Web Server with SSL

Whenever sensitive data, such as credit card information, is transferred between Web server and client, it would be desirable to have a secure, encrypted connection with authentication. mod_ssl provides strong encryption using the secure sockets layer (SSL) and transport layer security (TLS) protocols for HTTP communication between a client and the Web server. Using SSL/TSL, a private connection between Web server and client is established. Data integrity is ensured and client and server are able to authenticate each other.

For this purpose, the server sends an SSL certificate that holds information proving the server's valid identity before any request to a URL is answered. In turn, this guarantees that the server is the uniquely correct end point for the communication. Additionally, the certificate generates an encrypted connection between client and server that can transport information without the risk of exposing sensitive, plain-text content.

mod_ssl does not implement the SSL/TSL protocols itself, but acts as an interface between Apache and an SSL library. In openSUSE, the OpenSSL library is used. OpenSSL is automatically installed with Apache.

The most visible effect of using mod_ssl with Apache is that URLs are prefixed with `https://` instead of `http://`.

30.6.1 Creating an SSL Certificate

In order to use SSL/TSL with the Web server, you need to create an SSL certificate. This certificate is needed for the authorization between Web server and client, so that each party can clearly identify the other party. To ensure the integrity of the certificate, it must be signed by a party every user trusts.

There are three types of certificates you can create: a “dummy” certificate for testing purposes only, a self-signed certificate for a defined circle of users that trust you, and a certificate signed by an independent, publicly-known certificate authority (CA).

Creating a certificate is basically a two step process. First, a private key for the certificate authority is generated then the server certificate is signed with this key.

TIP: For More Information

To learn more about concepts and definitions of SSL/TSL, refer to `http://httpd.apache.org/docs/2.2/ssl/ssl_intro.html`.

Creating a “Dummy” Certificate

Generating a dummy certificate is simple. Just call the script `/usr/bin/gensslcert`. It creates or overwrites the following files:

- `/etc/apache2/ssl.crt/ca.crt`

- /etc/apache2/ssl.crt/server.crt

- /etc/apache2/ssl.key/server.key

- /etc/apache2/ssl.csr/server.csr

A copy of ca.crt is also placed at /srv/www/htdocs/CA.crt for download.

IMPORTANT

A dummy certificate should never be used on a production system. Only use it for testing purposes.

Creating a Self-Signed Certificate

If you are setting up a secure Web server for an Intranet or for a defined circle of users, it might be sufficient if you sign a certificate with your own certificate authority (CA).

Creating a self-signed certificate is an interactive nine-step process. Change into the directory /usr/share/doc/packages/apache2 and run the following command: ./mkcert.sh make --no-print-directory /usr/bin/openssl /usr/sbin/ custom. Do not attempt to run this command from outside this directory. The program provides a series of prompts, some of which require user input.

***Procedure 30.1** Creating a Self-Signed Certificate with mkcert.sh*

1 Decide the signature algorithm used for certificates

   Choose RSA (R, the default), because some older browsers have problems with DSA.

2 Generating RSA private key for CA (1024 bit)

   No interaction needed.

3 Generating X.509 certificate signing request for CA

   Create the CA's distinguished name here. This requires you to answer a few questions, such as country name or organization name. Enter valid data, because everything you enter here later shows up in the certificate. You do not need to answer every question. If one does not apply to you or you want to leave it blank, use “.”. Common name is the name of the CA itself—choose a significant name, such as `My company` CA.

4 `Generating X.509 certificate for CA signed by itself`

Choose certificate version 3 (the default).

5 `Generating RSA private key for SERVER (1024 bit)`

No interaction needed.

6 `Generating X.509 certificate signing request for SERVER`

Create the distinguished name for the server key here. Questions are almost identical to the ones already answered for the CA's distinguished name. The data entered here applies to the Web server and does not necessarily need to be identical to the CA's data (for example, if the server is located elsewhere).

IMPORTANT: Selecting a Common Name

The common name you enter here must be the fully qualified hostname of your secure server (for example, www.example.com). Otherwise the browser issues a warning that the certificate does not match the server when accessing the Web server.

7 `Generating X.509 certificate signed by own CA`

Choose certificate version 3 (the default).

8 `Encrypting RSA private key of CA with a pass phrase for security`

It is strongly recommended to encrypt the private key of the CA with a password, so choose Y and enter a password.

9 `Encrypting RSA private key of SERVER with a pass phrase for security`

Encrypting the server key with a password requires you to enter this password every time you start the Web server. This makes it difficult to automatically start the server on boot or to restart the Web server. Therefore, it is common sense to say N to this question. Keep in mind that your key is unprotected when not encrypted with a password and make sure that only authorized persons have access to the key.

IMPORTANT: Encrypting the Server Key

If you choose to encrypt the server key with a password, increase the value for `APACHE_TIMEOUT` in `/etc/sysconfig/apache2`. Otherwise you do not have enough time to enter the passphrase before the attempt to start the server is stopped unsuccessfully.

The script's result page presents a list of certificates and keys it has generated. Contrary to what the script outputs, the files have not been generated in the local directory `conf`, but to the correct locations under `/etc/apache2/`.

The last step is to copy the CA certificate file from `/etc/apache2/ssl.crt/ca.crt` to a location where your users can access it in order to incorporate it into the list of known and trusted CAs in their Web browsers. Otherwise a browser complains that the certificate was issued by an unknown authority. The certificate is valid for one year.

IMPORTANT: Self-Signed Certificates

Only use a self-signed certificate on a Web server that is accessed by people who know and trust you as a certificate authority. It is not recommended to use such a certificate on a public shop, for example.

Getting an Officially Signed Certificate

There are a number of official certificate authorities that sign your certificates. The certificate is signed by a trustworthy third party, so can be fully trusted. Publicly operating secure Web servers usually have got an officially signed certificate.

The best-known official CAs are Thawte (`http://www.thawte.com/`) or Verisign (`http://www.verisign.com`). These and other CAs are already compiled into all browsers, so certificates signed by these certificate authorities are automatically accepted by the browser.

When requesting an officially signed certificate, you do not send a certificate to the CA. Instead, issue a Certificate Signing Request (CSR). To create a CSR, call the script `/usr/share/ssl/misc/CA.sh -newreq`.

First the script asks for a password with which the CSR should be encrypted. Then you are asked to enter a distinguished name. This requires you to answer a few questions, such as country name or organization name. Enter valid data—everything you enter here later shows up in the certificate and is checked. You do not need to answer every question. If one does not apply to you or you want to leave it blank, use ".". Common name is the name of the CA itself—choose a significant name, such as *My company* CA. Last, a challenge password and an alternative company name must be entered.

Find the CSR in the directory from which you called the script. The file is named `newreq.pem`.

30.6.2 Configuring Apache with SSL

The default port for SSL and TLS requests on the Web server side is 443. There is no conflict between a "regular" Apache listening on port 80 and an SSL/TLS-enabled Apache listening on port 443. In fact, HTTP and HTTPS can be run with the same Apache instance. Usually separate virtual hosts are used to dispatch requests to port 80 and port 443 to separate virtual servers.

IMPORTANT: Firewall Configuration

Do not forget to open the firewall for SSL-enabled Apache on port 443. This can be done with YaST as described in Section 35.4.1, "Configuring the Firewall with YaST" (page 583).

To use SSL, it must be activated in the global server configuration. Open `/etc/sysconfig/apache2` in an editor and search for `APACHE_MODULES`. Add "ssl" to the list of modules if it is not already present (mod_ssl is activated by default). Next, search for `APACHE_SERVER_FLAGS` and add "SSL". If you have chosen to encrypt your server certificate with a password, you should also increase the value for `APACHE_TIMEOUT`, so you have enough time to enter the passphrase when Apache starts. Restart the server to make these changes active. A reload is not sufficient.

The virtual host configuration directory contains a template `/etc/apache2/vhosts.d/vhost-ssl.template` with SSL-specific directives that are extensively documented. Refer to Section “Virtual Host Configuration” (page 486) for the general virtual host configuration.

To get started, it should be sufficient to adjust the values for the following directives:

- `DocumentRoot`
- `ServerName`
- `ServerAdmin`
- `ErrorLog`
- `TransferLog`

IMPORTANT: Name-Based Virtual Hosts and SSL

It is not possible to run multiple SSL-enabled virtual hosts on a server with only one IP address. Users connecting to such a setup receive a warning message stating that the certificate does not match the server name every time they visit the URL. A separate IP address or port is necessary for every SSL-enabled domain to achieve communication based on a valid SSL certificate.

30.7 Avoiding Security Problems

A Web server exposed to the public Internet requires an ongoing administrative effort. It is inevitable that security issues appear, both related to the software and to accidental misconfiguration. Here are some tips for how to deal with them.

30.7.1 Up-to-Date Software

If there are vulnerabilities found in the Apache software, a security advisory will be issued by SUSE. It contains instructions for fixing the vulnerabilities, which in turn should be applied soon as possible. The SUSE security announcements are available from the following locations:

- **Web Page** `http://www.novell.com/linux/security/securitysupport.html`

- **Mailing List** http://en.opensuse.org/Communicate#Mailinglists

- **RSS Feed** http://www.novell.com/linux/security/suse_security.xml

30.7.2 DocumentRoot Permissions

By default in openSUSE, the DocumentRoot directory /srv/www/htdocs and the CGI directory /srv/www/cgi-bin belong to the user and group root. You should not change these permissions. If the directories were writable for all, any user could place files into them. These files might then be executed by Apache with the permissions of wwwrun, which may give the user unintended access to file system resources. Use subdirectories of /srv/www to place the DocumentRoot and CGI directories for your virtual hosts and make sure that directories and files belong to user and group root.

30.7.3 File System Access

By default, access to the whole file system is denied in /etc/apache2/httpd.conf. You should never overwrite these directives, but specifically enable access to all directories Apache should be able to read (see Section “Basic Virtual Host Configuration” (page 489) for details). In doing so, ensure that no critical files, such as password or system configuration files, can be read from the outside.

30.7.4 CGI Scripts

Interactive scripts in Perl, PHP, SSI, or any other programming language can essentially run arbitrary commands and therefore present a general security issue. Scripts that will be executed from the server should only be installed from sources the server administrator trusts—allowing users to run their own scripts is generally not a good idea. It is also recommended to do security audits for all scripts.

To make the administration of scripts as easy as possible, it is common practice to limit the execution of CGI scripts to specific directories instead of globally allowing them. The directives ScriptAlias and Option ExecCGI are used for configuration. The openSUSE default configuration does not allow execution of CGI scripts from everywhere.

All CGI scripts run as the same user, so different scripts can potentially conflict with each other. The module suEXEC lets you run CGI scripts under a different user and group.

30.7.5 User Directories

When enabling user directories (with mod_userdir or mod_rewrite) you should strongly consider not allowing `.htaccess` files, which would allow users to overwrite security settings. At least you should limit the user's engagement by using the directive `AllowOverRide`. In openSUSE, `.htaccess` files are enabled by default, but the user is not allowed to overwrite any `Option` directives when using mod_userdir (see the `/etc/apache2/mod_userdir.conf` configuration file).

30.8 Troubleshooting

If Apache does not start, the Web page is not accessible, or users cannot connect to the Web server, it is important to find the cause of the problem. Here are some typical places to look for error explanations and important things to check.

First, `rcapache2` (described in Section 30.3, “Starting and Stopping Apache” (page 497)) is verbose about errors, so can be quite helpful if it is actually used for operating Apache. Sometimes it is tempting to use the binary `/usr/sbin/httpd2` for starting or stopping the Web server. Avoid doing this and use the `rcapache2` script instead. `rcapache2` even provides tips and hints for solving configuration errors.

Second, the importance of log files cannot be overemphasized. In case of both fatal and nonfatal errors, the Apache log files, mainly the error log file, are the places to look for causes. Additionally, you can control the verbosity of the logged messages with the `LogLevel` directive if more detail is needed in the log files. By default, the error log file is located at `/var/log/apache2/error_log`.

TIP: A Simple Test

Watch the Apache log messages with the command `tail -F /var/log/apache2/my_error_log`. Then run `rcapache2 restart`. Now, try to connect with a browser and check the output.

A common mistake is not to open the ports for Apache in the firewall configuration of the server. If you configure Apache with YaST, there is a separate option available to take care of this specific issue (see Section 30.2.2, "Configuring Apache with YaST" (page 490)). If you are configuring Apache manually, open firewall ports for HTTP and HTTPS via YaST's firewall module.

If the error cannot be tracked down with the help of any these, check the online Apache bug database at http://httpd.apache.org/bug_report.html. Additionally, the Apache user community can be reached via a mailing list available at http://httpd.apache.org/userslist.html. A recommended newsgroup is comp.infosystems.www.servers.unix.

30.9 For More Information

The package `apache2-doc` contains the complete Apache manual in various localizations for local installation and reference. It is not installed by default—the quickest way to install it is to use the command `yast -i apache2-doc`. Once installed, the Apache manual is available at http://localhost/manual/. You may also access it on the Web at http://httpd.apache.org/docs-2.2/. SUSE-specific configuration hints are available in the directory `/usr/share/doc/packages/apache2/README.*`.

30.9.1 Apache 2.2

For a list of new features in Apache 2.2, refer to http://httpd.apache.org/docs/2.2/new_features_2_2.html. Information about upgrading from version 2.0 to 2.2 is available at http://httpd.apache.org/docs-2.2/upgrading.html.

30.9.2 Apache Modules

More information about external Apache modules from Section 30.4.5, "External Modules" (page 504) is available at the following locations:

mod-apparmor
  `http://en.opensuse.org/AppArmor`

mod_fcgid
  `http://fastcgi.coremail.cn/`

mod_perl
  `http://perl.apache.org/`

mod_php5
  `http://www.php.net/manual/en/install.unix.apache2.php`

mod_python
  `http://www.modpython.org/`

30.9.3 Development

More information about developing Apache modules or about getting involved in the Apache Web server project are available at the following locations:

Apache Developer Information
  `http://httpd.apache.org/dev/`

Apache Developer Documentation
  `http://httpd.apache.org/docs/2.2/developer/`

Writing Apache Modules with Perl and C
  `http://www.modperl.com/`

30.9.4 Miscellaneous Sources

If you experience difficulties specific to Apache in openSUSE, take a look at the openSUSE wiki at `http://http://en.opensuse.org/Apache`. The history of Apache is provided at `http://httpd.apache.org/ABOUT_APACHE.html`. This page also explains why the server is called Apache.

Setting up a FTP server with YaST 31

Using the YaST *FTP Server* module, you can configure your machine to function as a FTP server. Anonymous and/or authenticated users can connect to your machine and download and, depending on the configuration, upload files using the FTP protocol. YaST provides a unified configuration interface for various FTP server daemons installed on your system.

The YaST *FTP Server* configuration module can be used to configure two different FTP server daemons: vsftpd (Very Secure FTP Daemon) and pure-ftpd. Only installed servers can be configured. Standard openSUSE media does not contain the pure-ftpd package. However, if the pure-ftpd package is installed from another repository, it can be configured using the YaST module.

vsftpd and pure-ftpd have slightly different configuration options, especially in the *Experts Settings* dialog. This chapter describes the settings of the vsftpd for being the default server for openSUSE.

To configure the FTP server with run YaST and choose *Network Services* > *FTP Server*. If no FTP server is installed, you will be asked which server should be installed. Choose a server and confirm the dialog.

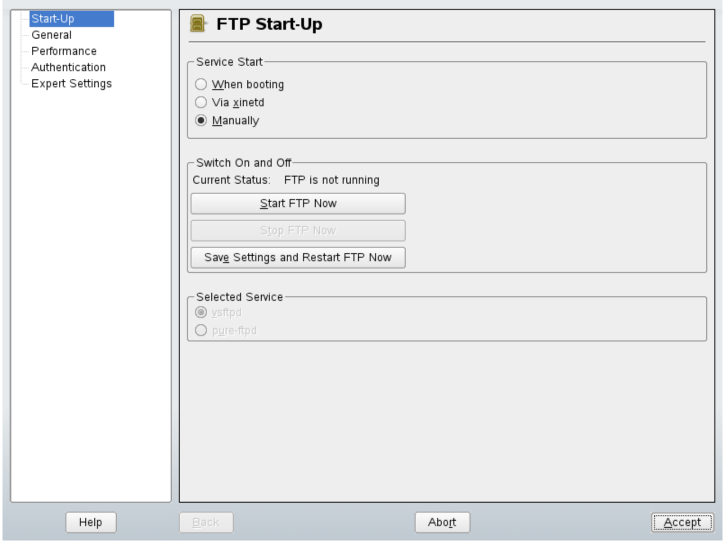
31.1 Starting the FTP server

In the *Service Start* frame of the *Start-Up* dialog you are able to set the way the FTP server is started up. You can choose between starting the server automatically during the system boot and starting it manually. If the FTP server should be started only after FTP connection request, choose *Via xinetd*.

The current status of the FTP server is shown in the *Switch On and Off* frame. Start the FTP server by pressing *Start FTP Now*. To stop the server, press *Stop FTP Now*. After having changed the settings of the server press *Save Settings and Restart FTP Now*. Your configurations will be saved by leaving the configuration module with *Accept* as well.

The *Select Service* frame of the *FTP Start-Up* dialog shows which FTP server is used. Either vsftpd (Very Secure FTP Daemon) or pure-ftpd can be used. If both servers are installed, you can choose between them. The pure-ftpd package is not included in the standard openSUSE media so you have to install it from a different installation source if you want to use it.

Figure 31.1 *FTP Server Configuration — Start-Up*



Service Start

○ When booting
○ Via xinetd
● Manually

Switch On and Off

Current Status: FTP is not running

Selected Service

● vsftpd
○ pure-ftpd

31.2 FTP General Settings

In the *General Settings* frame of the *FTP General Settings* dialog you can set the *Welcome message* which is shown after connecting to the FTP server.

If you check the *Chroot Everyone* option, all local users will be placed in a chroot jail in their home directory after login. This option has security implications, especially if the users have upload permission or shell access, so be careful enabling this option.

If you check the *Verbose Logging* option, all FTP requests and responses are logged.

In the *Umask for Anonymous* and *Umask for Authenticated Users* set the file creation mask for anonymous and authorized users respectively.

In the *FTP Directories* frame set the directories used for anonymous and authorized users. The default FTP directory for anonymous users is `/srv/ftp`. Note that vsftpd does not allow this directory to be writable for all users. The subdirectory `upload` with write permissions for anonymous users is created instead.

NOTE

pure-ftpd allows the FTP directory for anonymous users to be writable. Make sure you removed the write permissions in the directory that was used with pure-ftpd before switching back to the vsftpd server.

31.3 FTP Performance Settings

In the *FTP Performance Settings* set the parameters which affect the loading of the FTP server. *Max Idle Time* is the maximum time (in minutes) the remote client may spend between FTP commands. In case of longer inactivity, the remote client is disconnected. *Max Clients for One IP* determines the maximum number of clients which can be connected from a single IP address. *Max Clients* determines the maximum number of clients which may be connected. Any additional clients will get an error message.

The maximum data transfer rate (in KB/s) is set in *Local Max Rate* for local authenticated users, and in *Anonymous Max Rate* for anonymous clients respectively. The default value for the rate settings is 0, which means unlimited data transfer rate.

31.4 Authentication

In the *Enable/Disable Anonymous and Local Users* frame of the *Authentication* dialog, you are able to set which users are allowed to access your FTP server. You can grant access only for anonymous users, only for authenticated users with accounts on the system or for both types of users.

If you want to allow users to upload files to the FTP server, check *Enable Upload* in the *Uploading* frame of the *Authentication* dialog. Here you are able to allow uploading or creating directories even for anonymous users by checking the respective box.

NOTE

If a vsftpd server is used and you want anonymous users to be able to upload files or create directories, a subdirectory with writing permissions for all users has to be created in this directory.

31.5 Expert Settings

A FTP server can run in active or in passive mode. By default the server runs in passive mode. To switch into the active mode, just uncheck *Passive Mode* option in *Expert Settings* dialog. You can also change the range of ports on the server used for the data stream by tweaking the *Min Port for Pas. Mode* and *Max Port for Pas Mode* options.

If you want encrypted communication between clients and the server, you can use the FTPS protocol (FTP/SSH). However note that FTPS is different from the much more common SFTP (SSH File Transport Protocol) protocol. If you want to use the FTPS, you can set options in the *Expert Settings* dialog.

If your system is protected by a firewall, check *Open Port in Firewall* to enable a connection to the FTP server.

31.6 For more information

For more information about the vsftpd server read the manual pages of `vsftpd` and `vsftpd.conf`.

Part V. Mobility

Power Management 32

Power management is especially important on laptop computers, but is also useful on other systems. ACPI (advanced configuration and power interface) is available on all modern computers (laptops, desktops, and servers). Power management technologies require suitable hardware and BIOS routines. Most laptops and many modern desktops and servers meet these requirements. It is also possible to control CPU frequency scaling to save power or decrease noise.

32.1 Power Saving Functions

Power saving functions are not only significant for the mobile use of laptops, but also for desktop systems. The main functions and their use in ACPI are:

Standby

This operating mode turns off the display. On some computers, the processor performance is throttled. This function corresponds to the ACPI state S1 or S2.

Suspend (to memory)

This mode writes the entire system state to the RAM. Subsequently, the entire system except the RAM is put to sleep. In this state, the computer consumes very little power. The advantage of this state is the possibility of resuming work at the same point within a few seconds without having to boot and restart applications. This function corresponds to the ACPI state S3. The support of this state is still under development and therefore largely depends on the hardware.

Hibernation (suspend to disk)
: In this operating mode, the entire system state is written to the hard disk and the system is powered off. There must be a swap partition at least as big as the RAM to write all the active data. Reactivation from this state takes about 30 to 90 seconds. The state prior to the suspend is restored. Some manufacturers offer useful hybrid variants of this mode, such as RediSafe in IBM Thinkpads. The corresponding ACPI state is S4. In Linux, suspend to disk is performed by kernel routines that are independent from ACPI.

Battery Monitor
: ACPI checks the battery charge status and provides information about it. Additionally, it coordinates actions to perform when a critical charge status is reached.

Automatic Power-Off
: Following a shutdown, the computer is powered off. This is especially important when an automatic shutdown is performed shortly before the battery is empty.

Shutdown of System Components
: Switching off the hard disk is the greatest single aspect of the power saving potential of the overall system. Depending on the reliability of the overall system, the hard disk can be put to sleep for some time. However, the risk of losing data increases with the duration of the sleep periods. Other components, like PCI devices that can be put into a special power saving mode, can be deactivated with ACPI (at least theoretically) or permanently disabled in the BIOS setup.

Processor Speed Control
: In connection with the CPU, energy can be saved in three different ways: frequency and voltage scaling (also known as PowerNow! or Speedstep), throttling, and putting the processor to sleep (C states). Depending on the operating mode of the computer, these methods can also be combined.

32.2 ACPI

ACPI (advanced configuration and power interface) was designed to enable the operating system to set up and control the individual hardware components. ACPI supersedes both PnP and APM. It delivers information about the battery, AC adapter, temperature, fan, and system events, like “close lid” or “battery low.”

The BIOS provides tables containing information about the individual components and hardware access methods. The operating system uses this information for tasks like assigning interrupts or activating and deactivating components. Because the operating system executes commands stored in the BIOS, the functionality depends on the BIOS implementation. The tables ACPI can detect and load are reported in `/var/log/boot.msg`. See Section 32.2.4, "Troubleshooting" (page 534) for more information about troubleshooting ACPI problems.

32.2.1 ACPI in Action

If the kernel detects an ACPI BIOS when the system is booted, ACPI is activated automatically. The boot parameter `acpi=force` may be necessary for some older machines. The computer must support ACPI 2.0 or later. Check the kernel boot messages in `/var/log/boot.msg` to see if ACPI was activated.

Subsequently, a number of modules must be loaded. This is done by the start script of acpid. If any of these modules cause problems, the respective module can be excluded from loading or unloading in `/etc/sysconfig/powersave/common`. The system log (`/var/log/messages`) contains the messages of the modules, enabling you to see which components were detected.

`/proc/acpi` now contains a number of files that provide information about the system state or can be used to change some of the states. Some features do not work yet because they are still under development and the support of some functions largely depends on the implementation of the manufacturer.

All files (except `dsdt` and `fadt`) can be read with `cat`. In some files, settings can be modified with `echo`, for example, `echo X > file` to specify suitable values for X. One possibility for easy access to those values is the `powersave` command, which acts as a front-end for the Powersave daemon. The following describes the most important files:

`/proc/acpi/info`
: General information about ACPI.

`/proc/acpi/alarm`
: Here, specify when the system should wake from a sleep state. Currently, this feature is not fully supported.

`/proc/acpi/sleep`
: Provides information about possible sleep states.

`/proc/acpi/event`
: All events are reported here and processed by the Powersave daemon (`powersaved`). If no daemon accesses this file, events, such as a brief click on the power button or closing the lid, can be read with `cat /proc/acpi/event` (terminate with Ctrl + C).

`/proc/acpi/dsdt` and `/proc/acpi/fadt`
: These files contain the ACPI tables DSDT (differentiated system description table) and FADT (fixed ACPI description table). They can be read with `acpidmp`, `acpidisasm`, and `dmdecode`. These programs and their documentation are located in the package `pmtools`. For example, `acpidmp DSDT | acpidisasm`.

`/proc/acpi/ac_adapter/AC/state`
: Shows whether the AC adapter is connected.

`/proc/acpi/battery/BAT*/{alarm,info,state}`
: Detailed information about the battery state. The charge level is read by comparing the `last full capacity` from `info` with the `remaining capacity` from `state`. A more comfortable way to do this is to use one of the special programs introduced in Section 32.2.3, "ACPI Tools" (page 534). The charge level at which a battery event (such as warning, low and critical) is triggered can be specified in `alarm`.

`/proc/acpi/button`
: This directory contains information about various switches, like the laptop lid and buttons.

`/proc/acpi/fan/FAN/state`
: Shows if the fan is currently active. Activate or deactivate the fan manually by writing `0` (on) or `3` (off) into this file. However, both the ACPI code in the kernel and the hardware (or the BIOS) overwrite this setting when the system gets too warm.

`/proc/acpi/processor/*`
: A separate subdirectory is kept for each CPU included in your system.

`/proc/acpi/processor/*/info`
: Information about the energy saving options of the processor.

`/proc/acpi/processor/*/power`
: Information about the current processor state. An asterisk next to `C2` indicates that the processor is idle. This is the most frequent state, as can be seen from the `usage` value.

`/proc/acpi/processor/*/throttling`
: Can be used to set the throttling of the processor clock. Usually, throttling is possible in eight levels. This is independent of the frequency control of the CPU.

`/proc/acpi/processor/*/limit`
: If the performance (outdated) and the throttling are automatically controlled by a daemon, the maximum limits can be specified here. Some of the limits are determined by the system. Some can be adjusted by the user.

`/proc/acpi/thermal_zone/`
: A separate subdirectory exists for every thermal zone. A thermal zone is an area with similar thermal properties whose number and names are designated by the hardware manufacturer. However, many of the possibilities offered by ACPI are rarely implemented. Instead, the temperature control is handled conventionally by the BIOS. The operating system is not given much opportunity to intervene, because the life span of the hardware is at stake. Therefore, some of the files only have a theoretical value.

`/proc/acpi/thermal_zone/*/temperature`
: Current temperature of the thermal zone.

`/proc/acpi/thermal_zone/*/state`
: The state indicates if everything is `ok` or if ACPI applies `active` or `passive` cooling. In the case of ACPI-independent fan control, this state is always `ok`.

`/proc/acpi/thermal_zone/*/cooling_mode`
: Select the cooling method controlled by ACPI. Choose from passive (less performance, economical) or active cooling mode (full performance, fan noise).

`/proc/acpi/thermal_zone/*/trip_points`
: Enables the determination of temperature limits for triggering specific actions, like passive or active cooling, suspension (`hot`), or a shutdown (`critical`). The

possible actions are defined in the DSDT (device-dependent). The trip points determined in the ACPI specification are `critical`, `hot`, `passive`, `active1`, and `active2`. Even if not all of them are implemented, they must always be entered in this file in this order. For example, the entry `echo 90:0:70:0:0 > trip_points` sets the temperature for `critical` to `90` and the temperature for `passive` to `70` (all temperatures measured in degrees Celsius).

`/proc/acpi/thermal_zone/*/polling_frequency`
: If the value in `temperature` is not updated automatically when the temperature changes, toggle the polling mode here. The command `echo X > /proc/acpi/thermal_zone/*/polling_frequency` causes the temperature to be queried every `X` seconds. Set `X=0` to disable polling.

None of these settings, information, and events need to be edited manually. This can be done with the Powersave daemon (powersaved) and its various front-ends, like powersave, kpowersave, and wmpowersave. See Section 32.2.3, “ACPI Tools” (page 534).

32.2.2 Controlling the CPU Performance

The CPU can save energy in three ways. Depending on the operating mode of the computer, these methods can be combined. Saving energy also means that the system heats up less and the fans are activated less frequently.

Frequency and Voltage Scaling
: PowerNow! and Speedstep are the designations AMD and Intel use for this technology. However, this technology is also applied in processors of other manufacturers. The clock frequency of the CPU and its core voltage are reduced at the same time, resulting in more than linear energy savings. This means that when the frequency is halved (half performance), far less than half of the energy is consumed. This technology is independent from ACPI. There are two main approaches to performing CPU frequency scaling—by the kernel itself or by a userspace application. Therefore, there are different kernel governors that can be set below `/sys/devices/system/cpu/cpu*/cpufreq/`.

  userspace governor
  : If the userspace governor is set, the kernel gives the control of CPU frequency scaling to a userspace application, usually a daemon. In openSUSE distributions, this daemon is the `powersaved` package. When this implementation is used,

the CPU frequency is adjusted in regard to the current system load. By default, one of the kernel implementations is used. However, on some hardware or in regard to specific processors or drivers, the userspace implementation is still the only working solution.

ondemand governor

This is the kernel implementation of a dynamic CPU frequency policy and should work on most systems. As soon as there is a high system load, the CPU frequency is immediately increased. It is lowered on a low system load.

conservative governor

This governor is similar to the on demand implementation, except that a more conservative policy is used. The load of the system must be high for a specific amount of time before the CPU frequency is increased.

powersave governor

The cpu frequency is statically set to the lowest possible.

performance governor

The cpu frequency is statically set to the highest possible.

Throttling the Clock Frequency

This technology omits a certain percentage of the clock signal impulses for the CPU. At 25% throttling, every fourth impulse is omitted. At 87.5%, only every eighth impulse reaches the processor. However, the energy savings are a little less than linear. Normally, throttling is only used if frequency scaling is not available or to maximize power savings. This technology, too, must be controlled by a special process. The system interface is `/proc/acpi/processor/*/throttling`.

Putting the Processor to Sleep

The operating system puts the processor to sleep whenever there is nothing to do. In this case, the operating system sends the CPU a `halt` command. There are three states: C1, C2, and C3. In the most economic state, C3, even the synchronization of the processor cache with the main memory is halted. Therefore, this state can only be applied if no other device modifies the contents of the main memory via bus master activity. Some drivers prevent the use of C3. The current state is displayed in `/proc/acpi/processor/*/power`.

Frequency scaling and throttling are only relevant if the processor is busy, because the most economic C state is applied anyway when the processor is idle. If the CPU is busy, frequency scaling is the recommended power saving method. Often the processor only works with a partial load. In this case, it can be run with a lower frequency. Usually, dynamic frequency scaling controlled by the kernel on demand governor or a daemon, such as powersaved, is the best approach. A static setting to a low frequency is useful for battery operation or if you want the computer to be cool or quiet.

Throttling should be used as the last resort, for example, to extend the battery operation time despite a high system load. However, some systems do not run smoothly when they are throttled too much. Moreover, CPU throttling does not make sense if the CPU has little to do.

In openSUSE these technologies are controlled by the powersave daemon. The configuration is explained in Section 32.4, "The powersave Package" (page 537).

32.2.3 ACPI Tools

The range of more or less comprehensive ACPI utilities includes tools that merely display information, like the battery charge level and the temperature (acpi, klaptopdaemon, wmacpimon, etc.), tools that facilitate the access to the structures in `/proc/acpi` or that assist in monitoring changes (akpi, acpiw, gtkacpiw), and tools for editing the ACPI tables in the BIOS (package `pmtools`).

32.2.4 Troubleshooting

There are two different types of problems. On one hand, the ACPI code of the kernel may contain bugs that were not detected in time. In this case, a solution will be made available for download. More often, however, the problems are caused by the BIOS. Sometimes, deviations from the ACPI specification are purposely integrated in the BIOS to circumvent errors in the ACPI implementation in other widespread operating systems. Hardware components that have serious errors in the ACPI implementation are recorded in a blacklist that prevents the Linux kernel from using ACPI for these components.

The first thing to do when problems are encountered is to update the BIOS. If the computer does not boot at all, one of the following boot parameters may be helpful:

pci=noacpi
: Do not use ACPI for configuring the PCI devices.

acpi=ht
: Only perform a simple resource configuration. Do not use ACPI for other purposes.

acpi=off
: Disable ACPI.

WARNING: Problems Booting without ACPI

Some newer machines (especially SMP systems and AMD64 systems) need ACPI for configuring the hardware correctly. On these machines, disabling ACPI can cause problems.

Monitor the boot messages of the system with the command `dmesg | grep -2i acpi` (or all messages, because the problem may not be caused by ACPI) after booting. If an error occurs while parsing an ACPI table, the most important table—the DSDT—can be replaced with an improved version. In this case, the faulty DSDT of the BIOS is ignored. The procedure is described in Section 32.4.3, "Troubleshooting" (page 541).

In the kernel configuration, there is a switch for activating ACPI debug messages. If a kernel with ACPI debugging is compiled and installed, experts searching for an error can be supported with detailed information.

If you experience BIOS or hardware problems, it is always advisable to contact the manufacturers. Especially if they do not always provide assistance for Linux, they should be confronted with the problems. Manufacturers will only take the issue seriously if they realize that an adequate number of their customers use Linux.

For More Information

- `http://www.cpqlinux.com/acpi-howto.html` (detailed ACPI HOWTO, contains DSDT patches)
- `http://www.intel.com/technology/iapc/acpi/faq.htm` (ACPI FAQ @Intel)
- `http://acpi.sourceforge.net/` (the ACPI4Linux project at Sourceforge)
- `http://www.poupinou.org/acpi/` (DSDT patches by Bruno Ducrot)

32.3 Rest for the Hard Disk

In Linux, the hard disk can be put to sleep entirely if it is not needed or it can be run in a more economic or quieter mode. On modern laptops, you do not need to switch off the hard disks manually, because they automatically enter an economic operating mode whenever they are not needed. However, if you want to maximize power savings, test some of the following methods.

The hdparm application can be used to modify various hard disk settings. The option `-y` instantly switches the hard disk to the standby mode. `-Y` puts it to sleep. `hdparm -S x` causes the hard disk to be spun down after a certain period of inactivity. Replace *x* as follows: `0` disables this mechanism, causing the hard disk to run continuously. Values from `1` to `240` are multiplied by 5 seconds. Values from `241` to `251` correspond to 1 to 11 times 30 minutes.

Internal power saving options of the hard disk can be controlled with the option `-B`. Select a value from `0` to `255` for maximum saving to maximum throughput. The result depends on the hard disk used and is difficult to assess. To make a hard disk quieter, use the option `-M`. Select a value from `128` to `254` for quiet to fast.

Often, it is not so easy to put the hard disk to sleep. In Linux, numerous processes write to the hard disk, waking it up repeatedly. Therefore, it is important to understand how Linux handles data that needs to be written to the hard disk. First, all data is buffered in the RAM. This buffer is monitored by the kernel update daemon (kupdated). When the data reaches a certain age limit or when the buffer is filled to a certain degree, the buffer content is flushed to the hard disk. The buffer size is dynamic and depends on the size of the memory and the system load. By default, kupdated is set to short intervals to achieve maximum data integrity. It checks the buffer every 5 seconds and notifies the bdflush daemon when data is older than 30 seconds or the buffer reaches a fill level of 30%. The bdflush daemon then writes the data to the hard disk. It also writes independently from kupdated if, for instance, the buffer is full.

WARNING: Impairment of the Data Integrity

Changes to the kernel update daemon settings endanger the data integrity.

Apart from these processes, journaling file systems, like ReiserFS and Ext3, write their metadata independently from bdflush, which also prevents the hard disk from spinning down. To avoid this, a special kernel extension has been developed for mobile devices. See /usr/src/linux/Documentation/laptop-mode.txt for details.

Another important factor is the way active programs behave. For example, good editors regularly write hidden backups of the currently modified file to the hard disk, causing the disk to wake up. Features like this can be disabled at the expense of data integrity.

In this connection, the mail daemon postfix makes use of the variable POSTFIX_LAPTOP. If this variable is set to yes, postfix accesses the hard disk far less frequently. However, this is irrelevant if the interval for kupdated was increased.

32.4 The powersave Package

The powersave package cares about all the previously-mentioned power saving functions. Due to the increasing demand for lower energy consumption in general, some of its features are also important on workstations and servers, such as suspend, standby, or CPU frequency scaling.

This package contains all power management features of your computer. It supports hardware using ACPI, IDE hard disks, and PowerNow! or SpeedStep technologies. The functions from the packages apmd, acpid, ospmd, and cpufreqd (now cpuspeed) have been consolidated in the powersave package. Daemons from these packages, except acpid that acts as a multiplexer for ACPI events, should not be run concurrently with the powersave daemon.

Even if your system does not contain all the hardware elements listed above, use the powersave daemon for controlling the power saving function. Because ACPI and APM are mutually exclusive, you can only use one of these systems on your computer. The daemon automatically detects any changes in the hardware configuration.

32.4.1 Configuring the powersave Package

The configuration of powersave is distributed to several files. Every configuration option listed there contains additional documentation about its functionality.

`/etc/sysconfig/powersave/common`

This file contains general settings for the powersave daemon. For example, the amount of debug messages in `/var/log/messages` can be increased by increasing the value of the variable `DEBUG`.

`/etc/sysconfig/powersave/events`

The powersave daemon needs this file for processing system events. An event can be assigned external actions or actions performed by the daemon itself. For external actions, the daemon tries to run an executable file (usually a Bash script) in `/usr/lib/powersave/scripts/`. Predefined internal actions are:

- ignore
- throttle
- dethrottle
- suspend_to_disk
- suspend_to_ram
- standby
- notify
- screen_saver
- reread_cpu_capabilities

`throttle` slows down the processor by the value defined in `MAX_THROTTLING`. This value depends on the current scheme. `dethrottle` sets the processor to full performance. `suspend_to_disk`, `suspend_to_ram`, and `standby` trigger the system event for a sleep mode. These three actions are generally responsible for triggering the sleep mode, but they should always be associated with specific system events.

The directory `/usr/lib/powersave/scripts` contains scripts for processing events:

switch_vt

Useful if the screen is displaced after a suspend or standby.

wm_logout
: Saves the settings and logs out from GNOME, KDE, or other window managers.

wm_shutdown
: Saves the GNOME or KDE settings and shuts down the system.

If, for example, the variable `EVENT_GLOBAL_SUSPEND2DISK="prepare_suspend_to_disk do_suspend_to_disk"` is set, the two scripts or actions are processed in the specified order as soon as the user gives powersaved the command for the sleep mode `suspend to disk`. The daemon runs the external script `/usr/lib/powersave/scripts/prepare_suspend_to_disk`. After this script has been processed successfully, the daemon runs the internal action `do_suspend_to_disk` and sets the computer to the sleep mode after the script has unloaded critical modules and stopped services.

The actions for the event of a sleep button could be modified as in `EVENT_BUTTON_SLEEP="notify suspend_to_disk"`. In this case, the user is informed about the suspend by a pop-up window in X or a message on the console. Subsequently, the event `EVENT_GLOBAL_SUSPEND2DISK` is generated, resulting in the execution of the mentioned actions and a secure system suspend mode. The internal action `notify` can be customized using the variable `NOTIFY_METHOD` in `/etc/sysconfig/powersave/common`.

`/etc/sysconfig/powersave/cpufreq`
: Contains variables for optimizing the dynamic CPU frequency settings and whether the user space or the kernel implementation should be used.

`/etc/sysconfig/powersave/battery`
: Contains battery limits and other battery-specific settings.

`/etc/sysconfig/powersave/thermal`
: Activates cooling and thermal control. Details about this subject are available in the file `/usr/share/doc/packages/powersave/README.thermal`.

`/etc/sysconfig/powersave/scheme_*`
: These are the various schemes that adapt the power consumption to certain deployment scenarios. A number of schemes are preconfigured and can be used as they are. Custom schemes can be saved here.

32.4.2 Additional ACPI Features

If you use ACPI, you can control the response of your system to *ACPI buttons* (power, sleep, lid open, and lid closed). Configure execution of the actions in /etc/sysconfig/powersave/events. Refer to this configuration file for an explanation of the individual options.

TIP: Configuring ACPI Buttons

The settings in /etc/sysconfig/powersave/event are only taken into account if no power management applet is run on the user's desktop (i.e. KPowersave or GNOME Power Manager).

EVENT_BUTTON_POWER="wm_shutdown"
: When the power button is pressed, the system responds by shutting down the respective window manager (KDE, GNOME, fvwm, etc.).

EVENT_BUTTON_SLEEP="suspend_to_disk"
: When the sleep button is pressed, the system is set to the suspend-to-disk mode.

EVENT_BUTTON_LID_OPEN="ignore"
: Nothing happens when the lid is opened.

EVENT_BUTTON_LID_CLOSED="screen_saver"
: When the lid is closed, the screen saver is activated.

EVENT_OTHER="ignore"
: This event happens if an unknown event is encountered by the daemon. Unknown events include ACPI hot keys on some machines.

Further throttling of the CPU performance is possible if the CPU load does not exceed a specified limit for a specified time. Specify the load limit in PROCESSOR_IDLE_LIMIT and the time-out in CPU_IDLE_TIMEOUT. If the CPU load stays below the limit longer than the time-out, the event configured in EVENT_PROCESSOR_IDLE is activated. If the CPU is busy again, EVENT_PROCESSOR_BUSY is executed.

32.4.3 Troubleshooting

All error messages and alerts are logged in the file `/var/log/messages`. If you cannot find the needed information, increase the verbosity of the messages of powersave using `DEBUG` in the file `/etc/sysconfig/powersave/common`. Increase the value of the variable to `7` or even `15` and restart the daemon. The more detailed error messages in `/var/log/messages` should help you to find the error. The following sections cover the most common problems with powersave and the different sleep modes.

ACPI Activated with Hardware Support but Functions Do Not Work

If you experience problems with ACPI, use the command `dmesg|grep -i acpi` to search the output of `dmesg` for ACPI-specific messages. A BIOS update may be required to resolve the problem. Go to the home page of your laptop manufacturer, look for an updated BIOS version, and install it. Ask the manufacturer to comply with the latest ACPI specification. If the errors persist after the BIOS update, proceed as follows to replace the faulty DSDT table in your BIOS with an updated DSDT:

1. Download the DSDT for your system from `http://acpi.sourceforge.net/dsdt/index.php`. Check if the file is decompressed and compiled as shown by the file extension `.aml` (ACPI machine language). If this is the case, continue with step 3.

2. If the file extension of the downloaded table is `.asl` (ACPI source language), compile it with iasl (package `pmtools`). Enter the command `iasl -sa file.asl`. The latest version of iasl (Intel ACPI compiler) is available at `http://developer.intel.com/technology/iapc/acpi/downloads.htm`.

3. Copy the file `DSDT.aml` to any location (`/etc/DSDT.aml` is recommended). Edit `/etc/sysconfig/kernel` and adapt the path to the DSDT file accordingly. Start `mkinitrd` (package `mkinitrd`). Whenever you install the kernel and use `mkinitrd` to create an `initrd`, the modified DSDT is integrated and loaded when the system is booted.

CPU Frequency Does Not Work

Refer to the kernel sources (kernel-source) to see if your processor is supported. You may need a special kernel module or module option to activate CPU frequency control. This information is available in /usr/src/linux/Documentation/cpu-freq/*.

Suspend and Standby Do Not Work

ACPI systems may have problems with suspend and standby due to a faulty DSDT implementation (BIOS). If this is the case, update the BIOS.

When the system tries to unload faulty modules, the system is arrested or the suspend event is not triggered. The same can also happen if you do not unload modules or stop services that prevent a successful suspend. In both cases, try to identify the faulty module that prevented the sleep mode. The log file /var/log/pm-suspend.log contains detailed information about what is going on and where possible errors are. Modify the SUSPEND_MODULES variable in /usr/lib/pm-utils/defaults to unload problematic modules prior to a suspend or standby.

Refer to http://www.opensuse.org/Pm-utils and http://www.opensuse.org/S2ram to get more detailed information on how to modify the suspend and resume process.

32.4.4 For More Information

- /usr/share/doc/packages/powersave—Local Powersave daemon documentation

- http://powersave.sourceforge.net—Most recent Powersave daemon documentation

- http://www.opensuse.org/Projects_Powersave—Project page in the openSUSE wiki

- http://www.opensuse.org/S2ram—How to get Suspend to RAM working

- http://www.opensuse.org/Pm-utils—How to modify the general suspend framework

Wireless Communication 33

There are several possibilities for using your Linux system to communicate with other computers, cellular phones, or peripheral devices. WLAN (wireless LAN) can be used to network laptops. Bluetooth can be used to connect individual system components (mouse, keyboard), peripheral devices, cellular phones, PDAs, and individual computers with each other. IrDA is mostly used for communication with PDAs or cellular phones. This chapter introduces all three technologies and their configuration.

33.1 Wireless LAN

Wireless LANs have become an indispensable aspect of mobile computing. Today, most laptops have built-in WLAN cards. The 802.11 standard for the wireless communication of WLAN cards was prepared by the IEEE organization. Originally, this standard provided for a maximum transmission rate of 2 Mbit/s. Meanwhile, several supplements have been added to increase the data rate. These supplements define details such as the modulation, transmission output, and transmission rates (see Table 33.1, “Overview of Various WLAN Standards” (page 544)).

Additionally, there are proprietary standards, like the 802.11b variation of Texas Instruments with a maximum transmission rate of 22 Mbit/s (sometimes referred to as 802.11b+). However, the popularity of cards using this standard is limited.

Table 33.1 Overview of Various WLAN Standards

Name	Band (GHz)	Maximum Transmission Rate (Mbit/s)	Note
802.11	2.4	2	Outdated; virtually no end devices available
802.11b	2.4	11	Widespread
802.11a	5	54	Less common
802.11g	2.4	54	Backward-compatible with 11b

33.1.1 Hardware

802.11 cards are not supported by openSUSE®. Most cards using 802.11a, 802.11b, and 802.11g are supported. New cards usually comply with the 802.11g standard, but cards using 802.11b are still available. Normally, cards with the following chips are supported:

- ADMTek ADM8211
- Aironet 4500, 4800
- Atheros 5210, 5211, 5212
- Atmel at76c502, at76c503, at76c504, at76c506
- Broadcom BCM43xx
- Intel PRO/Wireless 2100, 2200BG, 2915ABG, 3945ABG
- Intel Wireless WiFi Link 4965GN
- Intersil Prism2/2.5/3
- Intersil PrismGT

- Lucent/Agere Hermes

- Ralink RT2400, RT2500, RT2570, RT61, RT73

- Realtek RTL8187

- Texas Instruments ACX100, ACX111

- ZyDAS zd1201

A number of older cards that are rarely used and no longer available are also supported. An extensive list of WLAN cards and the chips they use is available at the Web site of *AbsoluteValue Systems* at `http://www.linux-wlan.org/docs/wlan_adapters.html.gz`. Find an overview of the various WLAN chips at `http://wiki.uni-konstanz.de/wiki/bin/view/Wireless/ListeChipsatz`.

Some cards need a firmware image that must be loaded into the card when the driver is initialized. This is the case with Intersil PrismGT, Atmel, and TI ACX100 and ACX111. The firmware can easily be installed with the YaST Online Update. The firmware for Intel PRO/Wireless cards ships with openSUSE and is automatically installed by YaST as soon as a card of this type is detected. More information about this subject is available in the installed system in `/usr/share/doc/packages/wireless-tools/README.firmware`.

33.1.2 Function

In wireless networking, various techniques and configurations are used to ensure fast, high-quality, and secure connections. Different operating types suit different setups. It can be difficult to choose the right authentication method. The available encryption methods have different advantages and pitfalls.

Operating Mode

Basically, wireless networks can be classified as managed networks and ad-hoc networks. Managed networks have a managing element: the access point. In this mode (also referred to as infrastructure mode), all connections of the WLAN stations in the network run over the access point, which may also serve as a connection to an ethernet. Ad-hoc networks do not have an access point. The stations communicate directly with each other. The transmission range and number of participating stations are greatly limited in ad-hoc networks. Therefore, an access point is usually more efficient. It is even possible to use a WLAN card as an access point. Most cards support this functionality.

Because a wireless network is much easier to intercept and compromise than a wired network, the various standards include authentication and encryption methods. In the original version of the IEEE 802.11 standard, these are described under the term WEP. However, because WEP has proven to be insecure (see Section “Security” (page 552)), the WLAN industry (joined under the name *Wi-Fi Alliance*) has defined a new extension called WPA, which is supposed to eliminate the weaknesses of WEP. The later IEEE 802.11i standard (also referred to as WPA2, because WPA is based on a draft version 802.11i) includes WPA and some other authentication and encryption methods.

Authentication

To make sure that only authorized stations can connect, various authentication mechanisms are used in managed networks:

Open

An open system is a system that does not require authentication. Any station can join the network. Nevertheless, WEP encryption (see Section “Encryption” (page 547)) can be used.

Shared Key (according to IEEE 802.11)

In this procedure, the WEP key is used for the authentication. However, this procedure is not recommended, because it makes the WEP key more susceptible to attacks. All an attacker needs to do is to listen long enough to the communication between the station and the access point. During the authentication process, both sides exchange the same information, once in encrypted form and once in unencrypted form. This makes it possible for the key to be reconstructed with suitable tools. Because this method makes use of the WEP key for the authentication and for the encryption, it does not enhance the security of the network. A station that has the correct WEP key can authenticate, encrypt, and decrypt. A station that does not have the key cannot decrypt received packets. Accordingly, it cannot communicate, regardless of whether it had to authenticate itself.

WPA-PSK (according to IEEE 802.1x)

WPA-PSK (PSK stands for preshared key) works similarly to the Shared Key procedure. All participating stations as well as the access point need the same key. The key is 256 bits in length and is usually entered as a passphrase. This system does not need a complex key management like WPA-EAP and is more suitable for private use. Therefore, WPA-PSK is sometimes referred to as WPA “Home”.

WPA-EAP (according to IEEE 802.1x)

Actually, WPA-EAP is not an authentication system but a protocol for transporting authentication information. WPA-EAP is used to protect wireless networks in enterprises. In private networks, it is scarcely used. For this reason, WPA-EAP is sometimes referred to as WPA “Enterprise”.

WPA-EAP needs a Radius server to authenticate users. EAP offers three different methods for connecting and authenticating to the server: TLS (Transport Layer Security), TTLS (Tunneled Transport Layer Security), and PEAP (Protected Extensible Authentication Protocol). In a nutshell, these options work as follows:

EAP-TLS

TLS authentication relies on the mutual exchange of certificates both for server and client. First, the server presents its certificate to the client where it is evaluated. If the certificate is considered valid, the client in turn presents its certificate to the server. While TLS is secure, it requires a working certification management infrastructure in your network. This infrastructure is rarely found in private networks.

EAP-TTLS and PEAP

Both TTLS and PEAP are two-stage protocols. In the first stage, a secure is established and in the second one the client authentication data is exchanged. They require far less certification management overhead than TLS, if any.

Encryption

There are various encryption methods to ensure that no unauthorized person can read the data packets that are exchanged in a wireless network or gain access to the network:

WEP (defined in IEEE 802.11)

This standard makes use of the RC4 encryption algorithm, originally with a key length of 40 bits, later also with 104 bits. Often, the length is declared as 64 bits or 128 bits, depending on whether the 24 bits of the initialization vector are included. However, this standard has some weaknesses. Attacks against the keys generated by this system may be successful. Nevertheless, it is better to use WEP than not encrypt the network at all.

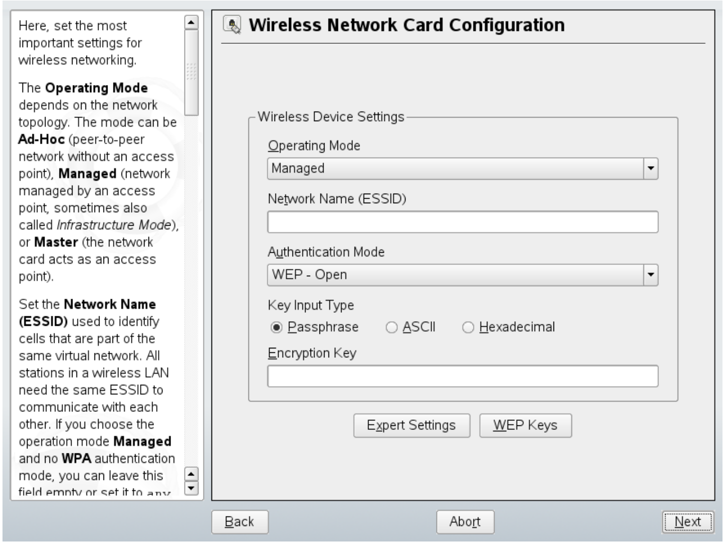
TKIP (defined in WPA/IEEE 802.11i)
: This key management protocol defined in the WPA standard uses the same encryption algorithm as WEP, but eliminates its weakness. Because a new key is generated for every data packet, attacks against these keys are in vain. TKIP is used together with WPA-PSK.

CCMP (defined in IEEE 802.11i)
: CCMP describes the key management. Usually, it is used in connection with WPA-EAP, but it can also be used with WPA-PSK. The encryption takes place according to AES and is stronger than the RC4 encryption of the WEP standard.

33.1.3 Configuration with YaST

To configure your wireless network card, start the YaST *Network Card* module. Here you can also choose whether to use YaST or NetworkManager for managing your network card. If you select YaST, select the device type *Wireless* in *Network Address Setup* and click *Next*. In *Wireless Network Card Configuration*, shown in Figure 33.1, "YaST: Configuring the Wireless Network Card" (page 548), make the basic settings for the WLAN operation:

Figure 33.1 *YaST: Configuring the Wireless Network Card*



Here, set the most important settings for wireless networking.

The **Operating Mode** depends on the network topology. The mode can be **Ad-Hoc** (peer-to-peer network without an access point), **Managed** (network managed by an access point, sometimes also called *Infrastructure Mode*), or **Master** (the network card acts as an access point).

Set the **Network Name (ESSID)** used to identify cells that are part of the same virtual network. All stations in a wireless LAN need the same ESSID to communicate with each other. If you choose the operation mode **Managed** and no **WPA** authentication mode, you can leave this field empty or set it to any

Wireless Network Card Configuration

Wireless Device Settings

Operating Mode: Managed

Network Name (ESSID):

Authentication Mode: WEP - Open

Key Input Type: ● Passphrase ○ ASCII ○ Hexadecimal

Encryption Key:

Operating Mode

A station can be integrated in a WLAN in three different modes. The suitable mode depends on the network in which to communicate: *Ad-hoc* (peer-to-peer network without access point), *Managed* (network is managed by an access point), or *Master* (your network card should be used as the access point). To use any of the WPA-PSK or WPA-EAP modes, the operating mode must be set to *managed*.

Network Name (ESSID)

All stations in a wireless network need the same ESSID for communicating with each other. If nothing is specified, the card automatically selects an access point, which may not be the one you intended to use.

Authentication Mode

Select a suitable authentication method for your network: *Open*, *Shared Key*, *WPA-PSK*, or *WPA-EAP*. If you select WPA authentication, a network name must be set.

Expert Settings

This button opens a dialog for the detailed configuration of your WLAN connection. A detailed description of this dialog is provided later.

After completing the basic settings, your station is ready for deployment in the WLAN.

IMPORTANT: Security in Wireless Networks

Be sure to use one of the supported authentication and encryption methods to protect your network traffic. Unencrypted WLAN connections allow third parties to intercept all network data. Even a weak encryption (WEP) is better than none at all. Refer to Section "Encryption" (page 547) and Section "Security" (page 552) for information.

Depending on the selected authentication method, YaST prompts you to fine-tune the settings in another dialog. For *Open*, there is nothing to configure, because this setting implements unencrypted operation without authentication.

Shared Key

Set a key input type. Choose from *Passphrase*, *ASCII*, or *Hexadecimal*. You may keep up to four different keys to encrypt the transmitted data. Click *WEP Keys* to enter the key configuration dialog. Set the length of the key: *128 bit* or *64 bit*. The default setting is *128 bit*. In the list area at the bottom of the dialog, up to four different keys can be specified for your station to use for the encryption. Press *Set as Default* to define one of them as the default key. Unless you change this, YaST uses the first entered key as the default key. If the standard key is deleted, one of the other keys must be marked manually as the default key. Click *Edit* to modify existing list entries or create new keys. In this case, a pop-up window prompts you to select an input type (*Passphrase*, *ASCII*, or *Hexadecimal*). If you select *Passphrase*, enter a word or a character string from which a key is generated according to the length previously specified. *ASCII* requests an input of 5 characters for a 64-bit key and 13 characters for a 128-bit key. For *Hexadecimal*, enter 10 characters for a 64-bit key or 26 characters for a 128-bit key in hexadecimal notation.

WPA-PSK

To enter a key for WPA-PSK, select the input method *Passphrase* or *Hexadecimal*. In the *Passphrase* mode, the input must be 8 to 63 characters. In the *Hexadecimal* mode, enter 64 characters.

WPA-EAP

Enter the credentials you have been given by your network administrator. For TLS, provide *Identity*, *Client Certificate*, *Client Key*, and *Server Certificate*. TTLS and PEAP require *Identity* and *Password*. *Server Certificate* and *Anonymous Identity* are optional. YaST searches for any certificate under `/etc/cert`, so save the certificates given to you to this location and restrict access to these files to `0600` (owner read and write).

Click *Details* to enter the advanced authentication dialog for your WPA-EAP setup. Select the authentication method for the second stage of EAP-TTLS or EAP-PEAP communication. If you selected TTLS in the previous dialog, choose `any`, `MD5`, `GTC`, `CHAP`, `PAP`, `MSCHAPv1`, or `MSCHAPv2`. If you selected PEAP, choose `any`, `MD5`, `GTC`, or `MSCHAPv2`. *PEAP version* can be used to force the use of a certain PEAP implementation if the automatically-determined setting does not work for you.

Click *Expert Settings* to leave the dialog for the basic configuration of the WLAN connection and enter the expert configuration. The following options are available in this dialog:

Channel

The specification of a channel on which the WLAN station should work is only needed in *Ad-hoc* and *Master* modes. In *Managed* mode, the card automatically searches the available channels for access points. In *Ad-hoc* mode, select one of the 12 offered channels for the communication of your station with the other stations. In *Master* mode, determine on which channel your card should offer access point functionality. The default setting for this option is *Auto*.

Bit Rate

Depending on the performance of your network, you may want to set a certain bit rate for the transmission from one point to another. In the default setting *Auto*, the system tries to use the highest possible data transmission rate. Some WLAN cards do not support the setting of bit rates.

Access Point

In an environment with several access points, one of them can be preselected by specifying the MAC address.

Use Power Management

When you are on the road, use power saving technologies to maximize the operating time of your battery. More information about power management is available in Chapter 32, *Power Management* (page 527).

33.1.4 Utilities

hostap (package `hostap`) is used to run a WLAN card as an access point. More information about this package is available at the project home page (`http://hostap.epitest.fi/`).

kismet (package `kismet`) is a network diagnosis tool with which to listen to the WLAN packet traffic. In this way, you can also detect any intrusion attempts in your network. More information is available at `http://www.kismetwireless.net/` and in the manual page.

33.1.5 Tips and Tricks for Setting Up a WLAN

These tips can help tweak speed and stability as well as security aspects of your WLAN.

Stability and Speed

The performance and reliability of a wireless network mainly depend on whether the participating stations receive a clean signal from the other stations. Obstructions like walls greatly weaken the signal. The more the signal strength sinks, the more the transmission slows down. During operation, check the signal strength with the iwconfig utility on the command line (`Link Quality` field) or with NetworkManager or KNetworkManager. If you have problems with the signal quality, try to set up the devices somewhere else or adjust the position of the antennas of your access points. Auxiliary antennas that substantially improve the reception are available for a number of PCMCIA WLAN cards. The rate specified by the manufacturer, such as 54 Mbit/s, is a nominal value that represents the theoretical maximum. In practice, the maximum data throughput is no more than half this value.

Security

If you want to set up a wireless network, remember that anybody within the transmission range can easily access it if no security measures are implemented. Therefore, be sure to activate an encryption method. All WLAN cards and access points support WEP encryption. Although this is not entirely safe, it does present an obstacle for a potential attacker. WEP is usually adequate for private use. WPA-PSK would be even better, but it is not implemented in older access points or routers with WLAN functionality. On some devices, WPA can be implemented by means of a firmware update. Furthermore, Linux does not support WPA on all hardware components. If WPA is not available, WEP is better than no encryption. In enterprises with advanced security requirements, wireless networks should only be operated with WPA.

33.1.6 Troubleshooting

If your WLAN card fails to respond, check if you have downloaded the needed firmware. Refer to Section 33.1.1, “Hardware” (page 544). The following paragraphs cover some known problems.

Multiple Network Devices

Modern laptops usually have a network card and a WLAN card. If you configured both devices with DHCP (automatic address assignment), you may encounter problems with the name resolution and the default gateway. This is evident from the fact that you can ping the router but cannot surf the Internet. The Support Database features an article on this subject at http://en.opensuse.org/SDB:Name_Resolution_Does_Not_Work_with_Several_Concurrent_DHCP_Clients.

Problems with Prism2 Cards

Several drivers are available for devices with Prism2 chips. The various cards work more or less smoothly with the various drivers. With these cards, WPA is only possible with the hostap driver. If such a card does not work properly or not at all or you want to use WPA, read /usr/share/doc/packages/wireless-tools/README.prism2.

33.1.7 For More Information

The Internet pages of Jean Tourrilhes, who developed the *Wireless Tools* for Linux, present a wealth of useful information about wireless networks. See http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html.

33.2 Bluetooth

Bluetooth is a wireless technology for connecting various devices, such as cellular phones, PDAs, peripheral devices, laptops, or system components like the keyboard or mouse. The name is derived from the Danish king Harold Bluetooth, who united various warring factions in Scandinavia. The Bluetooth logo is based on the runes for “H” (resembles a star) and “B”.

A number of important aspects distinguish Bluetooth from IrDA. First, the individual devices do not need to “see” each other directly and, second, several devices can be connected in a network. However, the maximum data rate is 2.1 Mbps (in the current version 2.0). Theoretically, Bluetooth can even communicate through walls. In practice, however, this depends on the properties of the wall and the device class. There are three device classes with transmission ranges between 10 and 100 meters.

33.2.1 Basics

The following sections outline the basic principles of how Bluetooth works. Learn which software requirements need to be met, how Bluetooth interacts with your system, and how Bluetooth profiles work.

Software

To be able to use Bluetooth, you need a Bluetooth adapter (either a built-in adapter or an external device), drivers, and a Bluetooth protocol stack. The Linux kernel already contains the basic drivers for using Bluetooth. The Bluez system is used as protocol stack. To make sure that the applications work with Bluetooth, the base packages `bluez-libs` and `bluez-utils` must be installed. These packages provide a number of needed services and utilities. Additionally, some adapters, such as Broadcom or AVM BlueFritz!, require the `bluez-firmware` package to be installed. The `bluez-cups` package enables printing over Bluetooth connections. If you need to debug problems with Bluetooth connections, install the package `bluez-hcidump` and `bluez-test`.

General Interaction

A Bluetooth system consists of four interlocked layers that provide the desired functionality:

Hardware
: The adapter and a suitable driver for support by the Linux kernel.

Configuration Files
: Used for controlling the Bluetooth system.

Daemons
: Services that are controlled by the configuration files and provide the functionality.

Applications
: The applications allow the functionality provided by the daemons to be used and controlled by the user.

When inserting a Bluetooth adapter, its driver is loaded by the hotplug system. After the driver is loaded, the system checks the configuration files to see if Bluetooth should be started. If this is the case, it determines the services to start. Based on this information, the respective daemons are started.

Profiles

In Bluetooth, services are defined by means of profiles, such as the file transfer profile, the basic printing profile, and the personal area network profile. To enable a device to use the services of another device, both must understand the same profile—a piece of information that is often missing in the device package and manual. Unfortunately, some manufacturers do not comply strictly with the definitions of the individual profiles. Despite this, communication between the devices usually works smoothly.

In the following text, local devices are those physically connected to the computer. All other devices that can only be accessed over wireless connections are referred to as remote devices.

33.2.2 Configuration

This section introduces Bluetooth configuration. Learn which configuration files are involved, which tools are needed, and how to configure Bluetooth.

The configuration files for the individual components of the Bluez system are located in the directory `/etc/bluetooth`. The only exception is the file `/etc/sysconfig/bluetooth` for starting the components.

The configuration files described below can only be modified by the user `root`. Currently, there is no graphical user interface to change all settings. Most of these settings are only interesting for experienced users with special use cases. Usually, the default settings should be adequate.

Various settings, such as the device names and the security mode, can be changed in the configuration file `/etc/bluetooth/hcid.conf`. Usually, the default settings should be adequate. The file contains comments describing the options for the various settings. However, most of the settings included in this file can also be made with kbluetooth or bluez-gnome.

Two sections in the included file are designated as `options` and `device`. The first contains general information that hcid uses for starting. The latter contains settings for the individual local Bluetooth devices.

One of the most important settings of the options section is security auto;. If set to auto, hcid tries to use the local PIN for incoming connections. If it fails, it switches to none and establishes the connection anyway. For increased security, this default setting should be set to user to make sure that the user is requested to enter a PIN every time a connection is established.

Set the name under which the computer is displayed on the other side in the device section. The device class, such as Desktop, Laptop, or Server, is defined in this section. Authentication and encryption are also enabled or disabled here.

33.2.3 System Components and Utilities

The operability of Bluetooth depends on the interaction of various services. hcid (host controller interface daemon), serves as an interface for the Bluetooth device and controls it. hcid also provides the sdpd (service discovery protocol daemon), by means of which a device can detect the services the host makes available. hcid is activated automatically when the system is started or when a Bluetooth adapter is plugged in to the system. Should this not have been the case, start the Bluetooth subsystem manually with rcbluetooth start. This command must be executed as root.

The following paragraphs briefly describe the most important shell tools that can be used for working with Bluetooth. Although various graphical components are now available for controlling Bluetooth, it can be worthwhile to check these programs.

Some of the commands can only be executed as root. This includes the command l2ping *device_address* for testing the connection to a remote device.

hcitool

Use hcitool to determine whether local and remote devices are detected. The command hcitool dev lists the local devices. The output generates a line in the form *interface_name device_address* for every detected local device.

Search for remote devices with the command hcitool inq. Three values are returned for every detected device: the device address, the clock offset, and the device class. The device address is important, because other commands use it for identifying the target device. The clock offset mainly serves a technical purpose. The class specifies the device type and the service type as a hexadecimal value.

Use hcitool name *device-address* to determine the device name of a remote device. In the case of a remote computer, the class and the device name correspond to the information in its /etc/bluetooth/hcid.conf. Local device addresses generate an error output.

hciconfig

The command /usr/sbin/hciconfig delivers further information about the local device. If hciconfig is executed without any arguments, the output shows device information, such as the device name (hciX), the physical device address (a 12-digit number in the form 00:12:34:56:78), and information about the amount of transmitted data.

hciconfig hci0 name displays the name that is returned by your computer when it receives requests from remote devices. As well as querying the settings of the local device, hciconfig can modify these settings. For example, hciconfig hci0 name TEST sets the name to TEST.

sdptool

Use sdptool to check which services are made available by a specific device. The command sdptool browse *device_address* returns all services of a device. Use sdptool search *service_code* to search for a specific service. This command scans all accessible devices for the requested service. If one of the devices offers the service, the program prints the full service name returned by the device together with a brief description. View a list of all possible service codes by entering sdptool without any parameters.

33.2.4 Graphical Applications

In Konqueror, enter the URL bluetooth:/ to list local and remote Bluetooth devices. Double-click a device for an overview of the services provided by the device. If you move across one of the specified services with the mouse, the browser's status bar shows which profile is used for the service. If you click a service, a dialog opens, asking whether to save, use the service (an application must be started to do this), or cancel the action. Mark a check box if you do not want the dialog to be displayed again but always want the selected action to be performed. For some services, support is not yet available. Other graphical front-ends to Bluetooth are included in the `kbluetooth`, `bluez-gnome`, and `gnome-bluetooth` packages.

33.2.5 Example — Establishing a Network Connection via Bluetooth

In this example, a network connection is established between the hosts *H1* and *H2*. These two hosts have the Bluetooth device addresses *baddr1* and *baddr2* (determined on both hosts with the command `hcitool dev` as described above). The hosts should be identified with the IP addresses `192.168.1.3` (*H1*) and `192.168.1.4` (*H2*).

The Bluetooth connection is established with the help of pand (personal area networking daemon). The following commands must be executed by the user `root`. The description focuses on the Bluetooth-specific actions and does not provide a detailed explanation of the network command `ip`.

Enter `pand -s` to start pand on the host *H1*. Subsequently, establish a connection on the host *H2* with `pand -c` *`baddr1`*. If you enter `ip link show` on one of the hosts to list the available network interfaces, the output should contain an entry like the following:

```
bnep0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000
link/ether 00:12:34:56:89:90 brd ff:ff:ff:ff:ff:ff
```

Instead of `00:12:34:56:89:90`, the output should contain the local device address *baddr1* or *baddr2*. Now this interface must be assigned an IP address and activated. On *H1*, do this with the following two commands:

```
ip addr add 192.168.1.3/24 dev bnep0
ip link set bnep0 up
```

On *H2*, use the following commands:

```
ip addr add 192.168.1.4/24 dev bnep0
ip link set bnep0 up
```

Now *H1* can be accessed from *H2* at the IP `192.168.1.3`. Use the command `ssh 192.168.1.4` to access *H2* from *H1*, assuming *H2* runs an sshd, which is activated by default in openSUSE®. The command `ssh 192.168.1.4` can also be run as a normal user.

33.2.6 Troubleshooting

If you have difficulties establishing a connection, proceed according to the following list. Remember that the error can be on either side of a connection or even on both sides. If possible, reconstruct the problem with another Bluetooth device to verify that the device is not defective.

Is the local device listed in the output of `hcitool dev`?

> If the local device is not listed in this output, hcid is not started or the device is not recognized as a Bluetooth device. This can have various causes. The device may be defective or the correct driver may be missing. Laptops with built-in Bluetooth often have an on and off switch for wireless devices, like WLAN and Bluetooth. Check the manual of your laptop to see if your device has such a switch. Restart the Bluetooth system with the command `rcbluetooth restart` and check if any errors are reported in `/var/log/messages`.

Does your Bluetooth adapter need a firmware file?

> If it does, install `bluez-bluefw` and restart the Bluetooth system with `rcbluetooth restart`.

Does the output of `hcitool inq` return other devices?

> Test this command more than once. The connection may have interferences, because the frequency band of Bluetooth is also used by other devices.

Can the remote device "see" your computer?

> Try to establish the connection from the remote device. Check if this device sees the computer.

Can a network connection be established (see Section 33.2.5, "Example — Establishing a Network Connection via Bluetooth" (page 558))?

> The setup described in Section 33.2.5, "Example — Establishing a Network Connection via Bluetooth" (page 558) may not work for several reasons. For example, one of the two computers may not support SSH. Try `ping 192.168.1.3` or `ping 192.168.1.4`. If this works, check if sshd is active. Another problem could be that one of the two devices already has network settings that conflict with the address `192.168.1.X` in the example. If this is the case, try different addresses, such as `10.123.1.2` and `10.123.1.3`.

If you have installed the `bluez-hcidump` package, you can use `hcidump -X` to check what is sent between the devices. Sometimes the output helps give a hint where the problem is, but be aware of the fact that it is only partly in "clear text."

33.2.7 For More Information

Some additional (last-minute) documentation can be found in `/usr/share/doc/packages/bluez-utils/` (German and English versions available).

An extensive overview of various instructions for the use and configuration of Bluetooth is available at `http://www.bluez.org/`. Other useful information and instructions:

- Official HOWTO of the Bluetooth protocol stack integrated in the kernel: `http://bluez.sourceforge.net/howto/index.html`
- Connection to PalmOS PDA: `http://www.cs.ucl.ac.uk/staff/s.zachariadis/btpalmlinux.html`

33.3 Infrared Data Transmission

IrDA (Infrared Data Association) is an industry standard for wireless communication with infrared light. Many laptops sold today are equipped with an IrDA-compatible transceiver that enables communication with other devices, such as printers, modems, LANs, or other laptops. The transfer speed ranges from 2400 bps to 4 Mbps.

There are two IrDA operation modes. The standard mode, SIR, accesses the infrared port through a serial interface. This mode works on almost all systems and is sufficient for most requirements. The faster mode, FIR, requires a special driver for the IrDA chip. Not all chip types are supported in FIR mode because of a lack of appropriate drivers. Set the desired IrDA mode in the BIOS of your computer. The BIOS also shows which serial interface is used in SIR mode.

Find information about IrDA in the IrDA how-to by Werner Heuser at `http://tuxmobil.org/Infrared-HOWTO/Infrared-HOWTO.html`. Additionally refer to the Web site of the Linux IrDA Project at `http://irda.sourceforge.net/`.

33.3.1 Software

The necessary kernel modules are included in the kernel package. The package irda provides the necessary helper applications for supporting the infrared interface. Find the documentation at /usr/share/doc/packages/irda/README after the installation of the package.

33.3.2 Configuration

The IrDA system service is not started automatically when the system is booted. Use the YaST IrDA module for activation. Only one setting can be modified in this module: the serial interface of the infrared device. The test window shows two outputs. One is the output of irdadump, which logs all sent and received IrDA packets. This output should contain the name of the computer and the names of all infrared devices in transmission range. An example for these messages is shown in Section 33.3.4, "Troubleshooting" (page 562). All devices to which an IrDA connection exists are listed in the lower part of the window.

IrDA consumes a considerable amount of battery power, because a discovery packet is sent every few seconds to detect other peripheral devices. Therefore, IrDA should only be started when necessary if you depend on battery power. Enter the command rcirda start to activate it or rcirda stop to deactivate it. All needed kernel modules are loaded automatically when the interface is activated.

If preferred, configure manually in the file /etc/sysconfig/irda. This file contains only one variable, IRDA_PORT, which determines the interface to use in SIR mode.

33.3.3 Usage

Data can be sent to the device file /dev/irlpt0 for printing. The device file /dev/irlpt0 acts just like the normal /dev/lp0 cabled interface, except the printing data is sent wirelessly with infrared light. For printing, make sure that the printer is in visual range of the computer's infrared interface and the infrared support is started.

A printer that is operated over the infrared interface can be configured with the YaST printer module. Because it is not detected automatically, configure it manually by clicking *Add* > *Directly Connected Printers*. Select *IrDA Printer* and click *Next* to configure the printer device. Usually, irlpt0 is the right connection. Click *Finish* to apply your settings. Details about operating printers in Linux are available in Chapter 7, *Printer Operation* (page 97).

Communication with other hosts and with mobile phones or other similar devices is conducted through the device file /dev/ircomm0. The Siemens S25 and Nokia 6210 mobile phones, for example, can dial and connect to the Internet with the wvdial application using the infrared interface. Synchronizing data with a Palm Pilot is also possible, provided the device setting of the corresponding application has been set to /dev/ircomm0.

If you want, you can address only devices that support the printer or IrCOMM protocols. Devices that support the IROBEX protocol, such as the 3Com Palm Pilot, can be accessed with special applications, like irobexpalm and irobexreceive. Refer to the *IR-HOWTO* (http://tldp.org/HOWTO/Infrared-HOWTO/) for information. The protocols supported by the device are listed in brackets after the name of the device in the output of irdadump. IrLAN protocol support is still a "work in progress."

33.3.4 Troubleshooting

If devices connected to the infrared port do not respond, use the command irdadump (as root) to check if the other device is recognized by the computer. Something similar to Example 33.1, "Output of irdadump" (page 562) appears regularly when a Canon BJC-80 printer is in visible range of the computer:

Example 33.1 *Output of irdadump*

```
21:41:38.435239 xid:cmd 5b62bed5 > ffffffff S=6 s=0 (14)
21:41:38.525167 xid:cmd 5b62bed5 > ffffffff S=6 s=1 (14)
21:41:38.615159 xid:cmd 5b62bed5 > ffffffff S=6 s=2 (14)
21:41:38.705178 xid:cmd 5b62bed5 > ffffffff S=6 s=3 (14)
21:41:38.795198 xid:cmd 5b62bed5 > ffffffff S=6 s=4 (14)
21:41:38.885163 xid:cmd 5b62bed5 > ffffffff S=6 s=5 (14)
21:41:38.965133 xid:rsp 5b62bed5 < 6cac38dc S=6 s=5 BJC-80
                        hint=8804 [Printer IrCOMM ] (23)
21:41:38.975176 xid:cmd 5b62bed5 > ffffffff S=6 s=* earth
                        hint=0500 [ PnP Computer ] (21)
```

Check the configuration of the interface if there is no output or the other device does not reply. Verify that the correct interface is used. The infrared interface is sometimes located at /dev/ttyS2 or at /dev/ttyS3 and an interrupt other than IRQ 3 is sometimes used. These settings can be checked and modified in the BIOS setup menu of almost every laptop.

A simple video camera can also help in determining whether the infrared LED lights up at all. Most video cameras can see infrared light; the human eye cannot.

34 Using Tablet PCs

openSUSE® comes with support for Tablet PCs with serial Wacom devices (such as IBM/Lenovo X41, ACER TM C300/C301/C302 series, Fujitsu Lifebook T series (T3010/T4010), HP Compaq TC4200, Motion M1200), with FinePoint devices (such as Gateway Tablet PCs), and Fujitsu Siemens Computers P-Series. Learn how to install and configure your Tablet PC and discover some useful Linux* applications which accept input from digital pens.

After you have installed the Tablet PC packages and configured your digitizer correctly, input with the pen, also called a stylus, can be used for the following actions and applications:

- Logging in to KDM or GDM

- Unlocking your screen on the KDE and GNOME desktops

- Actions that can also be triggered by other pointing devices (such as mouse or touch pad), for example, moving the cursor on the screen, starting applications, closing, resizing and moving windows, shifting window focus, dragging and dropping objects

- Using gesture recognition in applications of the X Window System

- Drawing with The GIMP

- Taking notes or sketching with applications like Jarnal or Xournal or editing larger amounts of text with Dasher

NOTE: Keyboard or Mouse Needed for Installation

During installation of openSUSE, the pen cannot be used as an input device. If your Tablet PC does not feature a built-in keyboard or touch pad, connect an external keyboard or mouse to your Tablet PC for installation of your system.

34.1 Installing Tablet PC Packages

The packages needed for Tablet PCs are included in the `Laptop` installation pattern—if this is selected during installation, the following packages should already be installed on your system:

- `jarnal`: a Java-based note taking application
- `xournal`: an application for note taking and sketching
- `xstroke`: a gesture recognition program for the X Window System
- `xvkbd`: a virtual keyboard for the X Window System
- `x11-input-wacom`: the X input module for Wacom tablets
- `x11-input-wacom-tools`: configuration, diagnostics, and libraries for Wacom tablets
- `x11-input-fujitsu`: the X input module for Fujitsu P-Series tablets

If these packages are not installed, manually install the packages you need from command line or select the `Laptop` pattern for installation in YaST.

34.2 Configuring Your Tablet Device

After the tablet PC packages have been installed, configure the (internal or external) tablet device with SaX2.

1. Start SaX2 from the command line or by pressing Alt + F2 and entering `sax2`.

2. If you use a Wacom device, click *Tablet* to show the *Tablet Properties*.

   If you use the Fujitsu P-Series, click *Touchscreen* instead.

3. From the list on the right, select the vendor and the name of your tablet and check *Activate This Tablet*.

4. Switch to the *Electronic Pens* tab and make sure the following options are activated: *Add Pen* and *Add Eraser*.

5. Click *OK* to save the changes.

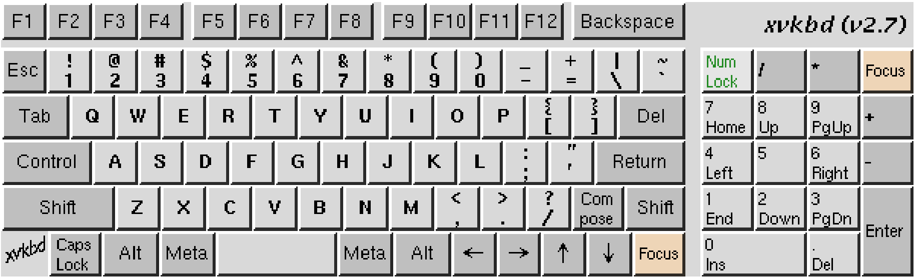
After finishing the X Window System configuration, restart your X server by logging out. Alternatively, leave the user interface and run `init 3 && init 5` in a virtual console.

After your tablet device has been configured, you can now make use of your pen as input device.

34.3 Using the Virtual Keyboard

To log in to the KDE or GNOME desktop or to unlock the screen, you can either enter your username and password as usual or via the virtual keyboard, xvkbd, displayed below the login field. To configure the keyboard or to access the integrated help, click the *xvkbd* field at the left lower corner to open the xvkbd main menu.

Figure 34.1 *xvkbd Virtual Keyboard*


xvkbd (v2.7)
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 Backspace
Esc ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 _ - + = | \ ~ `
Tab Q W E R T Y U I O P { [}] Del
Control A S D F G H J K L : ; " ' Return
Shift Z X C V B N M < , > . ? / Com pose Shift
xvkbd Caps Lock Alt Meta Meta Alt ← → ↑ ↓ Focus
Num Lock / * Focus
7 Home 8 Up 9 PgUp +
4 Left 5 6 Right -
1 End 2 Down 3 PgDn Enter
0 Ins . Del

If you want to use xvkbd after login, start it from the main menu or with `xvkbd` from a shell.

34.4 Rotating Your Display

When rotating your Tablet PC monitor, the orientation of your display and of your graphics tablet is not automatically adjusted. For the KDE desktop, use KRandRTray to rotate or resize your display manually on the fly. KRandRTray is a KDE applet for the RANDR extension of the X server.

1. Start KRandRTray from the main menu or with `krandrtray` from a shell. This adds the KRandRTray icon to your system tray.

2. To rotate your display, click the icon and select the desired orientation from the context menu. Your display is immediately tilted to the new direction. The orientation of the graphics tablet changes also so it can still interpret the movement of the pen correctly.

If you have problems changing the orientation of your desktop, refer to Section 34.7, “Troubleshooting” (page 571) for more information.

34.5 Using Gesture Recognition

With xstroke, you can use gestures with your pen or other pointing devices as input for applications on the X Window System. The xstroke alphabet is a unistroke alphabet that resembles the Graffiti* alphabet. When activated, xstroke sends the input to the currently focused window.

1. Start xstroke from the main menu or with `xstroke` from a shell. This adds a pencil icon to your system tray.

2. Start the application for which you want to create text input with the pen (for example, a terminal window, a text editor, or an OpenOffice.org Writer).

3. To activate the gesture recognition mode, click the pencil icon once.

4. Perform some gestures on the graphics tablet with the pen or another pointing device. xstroke captures the gestures and transfers them to text that appears in the application window that has the focus.

5. To switch focus to a different window, click the desired window with the pen and hold for a moment (or use the keyboard shortcut defined in your desktop's control center).

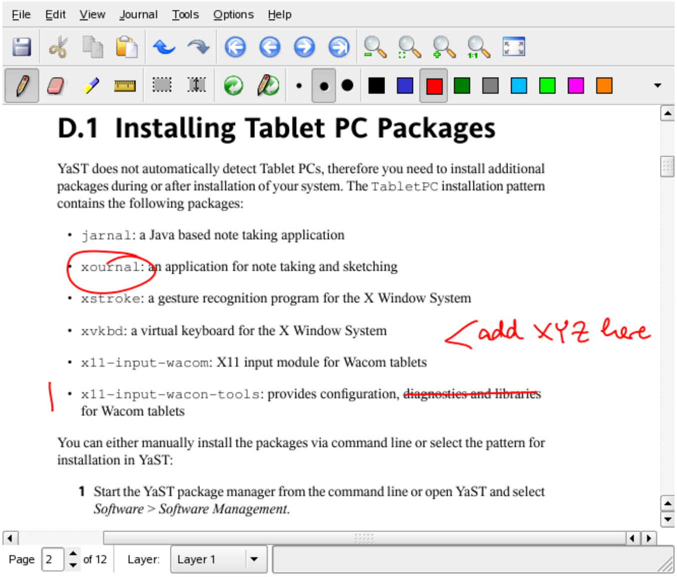
6. To deactivate the gesture recognition mode, click the pencil icon again.

34.6 Taking Notes and Sketching with the Pen

To create drawings with the pen, you can use a professional graphics editor like The GIMP or try one of the note taking applications, Xournal or Jarnal. With both Xournal and Jarnal, you can take notes, create drawings, or comment PDF files with the pen. As a Java-based application available for several platforms, Jarnal also offers basic collaboration features. For more information, refer to `http://www.dklevine.com/general/software/tc1000/jarnal-net.htm`. When saving your contents, Jarnal stores the data in an archive format (*.jaj) that also contains a file in SVG format.

Start Jarnal or Xournal from the main menu or by entering `jarnal` or `xournal` in a shell. To comment a PDF file in Xournal, for example, select *File* > *Annotate PDF* and open the PDF file from your file system. Use the pen or another pointing device to annotate the PDF then save your changes with *File* > *Print to PDF*.

Figure 34.2 Annotating a PDF with Xournal



D.1 Installing Tablet PC Packages

YaST does not automatically detect Tablet PCs, therefore you need to install additional packages during or after installation of your system. The `TabletPC` installation pattern contains the following packages:

- `jarnal`: a Java based note taking application
- `xournal`: an application for note taking and sketching
- `xstroke`: a gesture recognition program for the X Window System
- `xvkbd`: a virtual keyboard for the X Window System <add XYZ here
- `x11-input-wacom`: X11 input module for Wacom tablets
- `x11-input-wacon-tools`: provides configuration, ~~diagnostics and libraries~~ for Wacom tablets

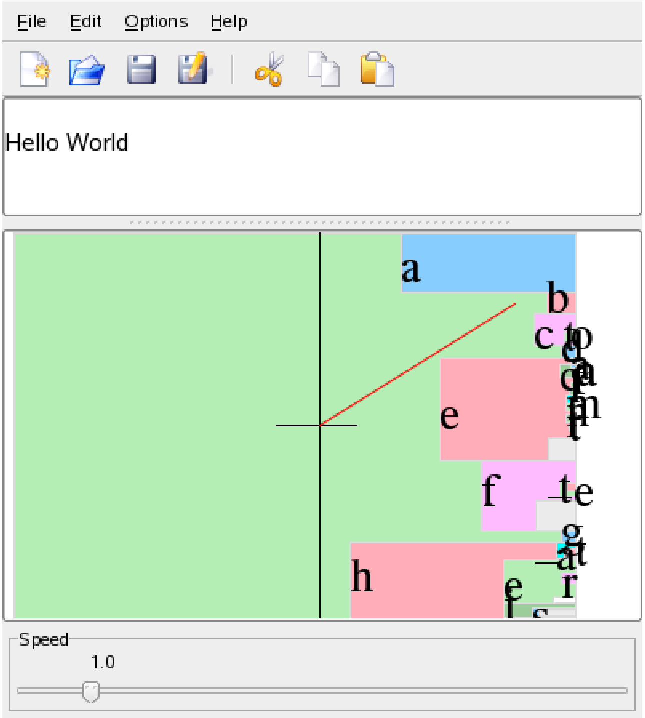
You can either manually install the packages via command line or select the pattern for installation in YaST:

1 Start the YaST package manager from the command line or open YaST and select *Software > Software Management*.

Dasher is another useful application. It was designed for situations where keyboard input is impractical or unavailable. With a bit of training, you can rapidly enter larger amounts of text using only the pen (or other input devices—it can even be driven with an eye tracker).

Start Dasher from the main menu or with `dasher` from a shell. Move your pen in one direction and the application starts to zoom into the letters on the right side. From the letters passing the cross hairs in the middle, the text is created or predicted and is printed to the upper part of the window. To stop or start writing, click the display once with the pen. Modify the zooming speed at the bottom of the window.

Figure 34.3 *Editing Texts with Dasher*



Hello World

Speed

1.0

The Dasher concept works for many languages. For more information, refer to the Dasher Web site, which offers comprehensive documentation, demonstrations and training texts. Find it at `http://www.inference.phy.cam.ac.uk/dasher/`

34.7 Troubleshooting

Virtual Keyboard Does Not Appear on Login Screen

> Occasionally, the virtual keyboard is not displayed on the login screen. To solve this, restart the X server by pressing Ctrl + Alt + <— or press the appropriate key on your Tablet PC (if you use a slate model without integrated keyboard). If the virtual keyboard still does not show, connect an external keyboard to your slate model and log in using the hardware keyboard.

Orientation of the Graphics Tablets Does Not Change in GNOME

With the `xrandr` command, you can change the orientation of your display from within a shell. Enter `xrandr --help` to view the options available. To simultaneously change the orientation of your graphics tablet, the command needs to be modified as described below:

- For normal orientation (0° rotation):

  ```
  xrandr -o 0 && xsetwacom set "Mouse[7]" Rotate 0
  ```

- For 90° rotation (clockwise, portrait):

  ```
  xrandr -o 3 && xsetwacom set "Mouse[7]" Rotate 1
  ```

- For 180° rotation (landscape):

  ```
  xrandr -o 2 && xsetwacom set "Mouse[7]" Rotate 3
  ```

- For 270° rotation (counterclockwise, portrait):

  ```
  xrandr -o 1 && xsetwacom set "Mouse[7]" Rotate 2
  ```

Note that the commands above depend on the contents of your `/etc/X11/xorg.conf` configuration file. If you have configured your device with SaX2 as described in Section 34.2, "Configuring Your Tablet Device" (page 567), the commands should work as they are written. If you have changed the `Identifier` of the tablet stylus input device in `xorg.conf` manually, replace `"Mouse[7]"` with the new `Identifier`.

34.8 For More Information

Some of the applications mentioned here do not offer integrated online help, but you can find some useful information about usage and configuration in your installed system in `/usr/share/doc/package/`*`packagename`* or on the Web:

- For the Xournal manual, refer to `http://xournal.sourceforge.net/manual.html`
- The Jarnal documentation is located at `http://www.dklevine.com/general/software/tc1000/jarnal.htm#documentation`
- Find the xstroke man page at `http://davesource.com/Projects/xstroke/xstroke.txt`
- Find a HOWTO for configuring X on the Linux Wacom Web site: `http://linuxwacom.sourceforge.net/index.php/howto/x11`
- Find a very informative Web site about the Dasher project at `http://www.inference.phy.cam.ac.uk/dasher/`

Part VI. Security

Masquerading and Firewalls 35

Whenever Linux is used in a networked environment, you can use the kernel functions that allow the manipulation of network packets to maintain a separation between internal and external network areas. The Linux netfilter framework provides the means to establish an effective firewall that keeps different networks apart. With the help of iptables—a generic table structure for the definition of rule sets—precisely control the packets allowed to pass a network interface. Such a packet filter can be set up quite easily with the help of SuSEfirewall2 and the corresponding YaST module.

35.1 Packet Filtering with iptables

The components netfilter and iptables are responsible for the filtering and manipulation of network packets as well as for network address translation (NAT). The filtering criteria and any actions associated with them are stored in chains, which must be matched one after another by individual network packets as they arrive. The chains to match are stored in tables. The `iptables` command allows you to alter these tables and rule sets.

The Linux kernel maintains three tables, each for a particular category of functions of the packet filter:

filter
: This table holds the bulk of the filter rules, because it implements the *packet filtering* mechanism in the stricter sense, which determines whether packets are let through (`ACCEPT`) or discarded (`DROP`), for example.

nat
: This table defines any changes to the source and target addresses of packets. Using these functions also allows you to implement *masquerading*, which is a special case of NAT used to link a private network with the Internet.

mangle
: The rules held in this table make it possible to manipulate values stored in IP headers (such as the type of service).

These tables contain several predefined chains to match packets:

PREROUTING
: This chain is applied to incoming packets.

INPUT
: This chain is applied to packets destined for the system's internal processes.

FORWARD
: This chain is applied to packets that are only routed through the system.

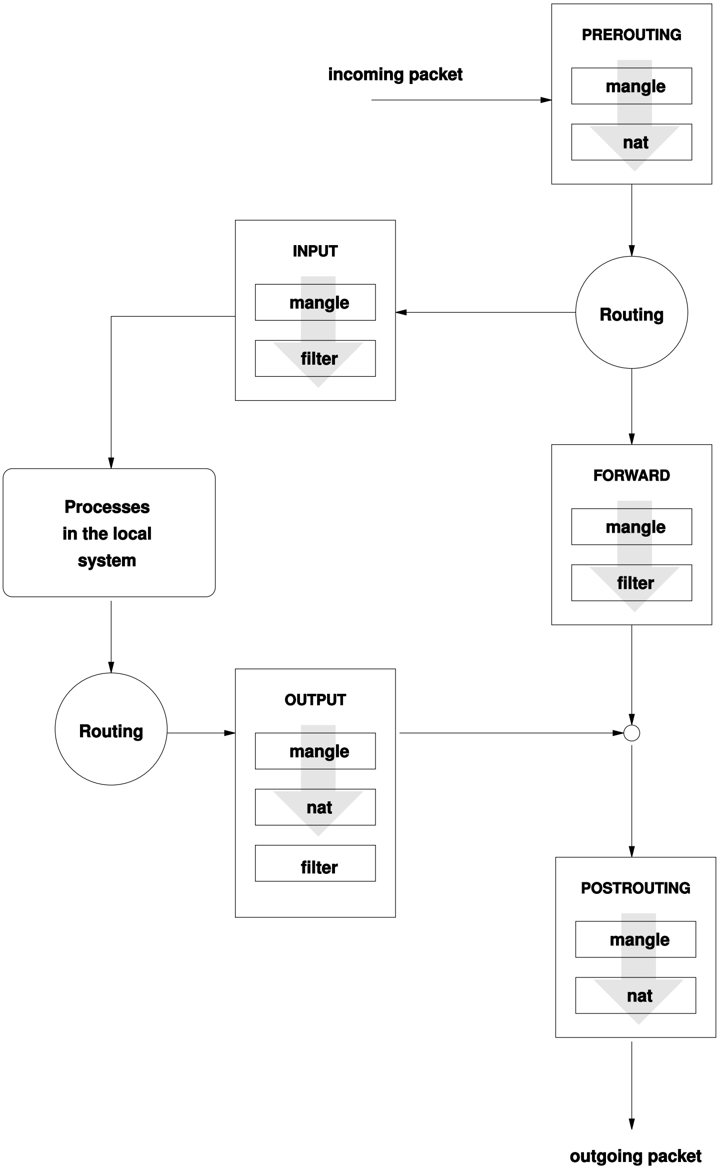
OUTPUT
: This chain is applied to packets originating from the system itself.

POSTROUTING
: This chain is applied to all outgoing packets.

Figure 35.1, "iptables: A Packet's Possible Paths" (page 579) illustrates the paths along which a network packet may travel on a given system. For the sake of simplicity, the figure lists tables as parts of chains, but in reality these chains are held within the tables themselves.

In the simplest of all possible cases, an incoming packet destined for the system itself arrives at the `eth0` interface. The packet is first referred to the `PREROUTING` chain of the `mangle` table then to the `PREROUTING` chain of the `nat` table. The following step, concerning the routing of the packet, determines that the actual target of the packet is a process of the system itself. After passing the `INPUT` chains of the `mangle` and the `filter` table, the packet finally reaches its target, provided that the rules of the `filter` table are actually matched.

Figure 35.1 *iptables: A Packet's Possible Paths*



35.2 Masquerading Basics

Masquerading is the Linux-specific form of NAT (network address translation). It can be used to connect a small LAN (where hosts use IP addresses from the private range—see Section 20.1.2, “Netmasks and Routing” (page 303)) with the Internet (where official IP addresses are used). For the LAN hosts to be able to connect to the Internet, their private addresses are translated to an official one. This is done on the router, which acts as the gateway between the LAN and the Internet. The underlying principle is a simple one: The router has more than one network interface, typically a network card and a separate interface connecting with the Internet. While the latter links the router with the outside world, one or several others link it with the LAN hosts. With these hosts in the local network connected to the network card (such as `eth0`) of the router, they can send any packets not destined for the local network to their default gateway or router.

IMPORTANT: Using the Correct Network Mask

When configuring your network, make sure both the broadcast address and the netmask are the same for all local hosts. Failing to do so prevents packets from being routed properly.

As mentioned, whenever one of the LAN hosts sends a packet destined for an Internet address, it goes to the default router. However, the router must be configured before it can forward such packets. For security reasons, this is not enabled in a default installation. To enable it, set the variable `IP_FORWARD` in the file `/etc/sysconfig/sysctl` to `IP_FORWARD=yes`.

The target host of the connection can see your router, but knows nothing about the host in your internal network where the packets originated. This is why the technique is called masquerading. Because of the address translation, the router is the first destination of any reply packets. The router must identify these incoming packets and translate their target addresses, so packets can be forwarded to the correct host in the local network.

With the routing of inbound traffic depending on the masquerading table, there is no way to open a connection to an internal host from the outside. For such a connection, there would be no entry in the table. In addition, any connection already established has a status entry assigned to it in the table, so the entry cannot be used by another connection.

As a consequence of all this, you might experience some problems with a number of application protocols, such as ICQ, cucme, IRC (DCC, CTCP), and FTP (in PORT mode). Web browsers, the standard FTP program, and many other programs use the PASV mode. This passive mode is much less problematic as far as packet filtering and masquerading are concerned.

35.3 Firewalling Basics

Firewall is probably the term most widely used to describe a mechanism that provides and manages a link between networks while also controlling the data flow between them. Strictly speaking, the mechanism described in this section is called a *packet filter*. A packet filter regulates the data flow according to certain criteria, such as protocols, ports, and IP addresses. This allows you to block packets that, according to their addresses, are not supposed to reach your network. To allow public access to your Web server, for example, explicitly open the corresponding port. However, a packet filter does not scan the contents of packets with legitimate addresses, such as those directed to your Web server. For example, if incoming packets were intended to compromise a CGI program on your Web server, the packet filter would still let them through.

A more effective but more complex mechanism is the combination of several types of systems, such as a packet filter interacting with an application gateway or proxy. In this case, the packet filter rejects any packets destined for disabled ports. Only packets directed to the application gateway are accepted. This gateway or proxy pretends to be the actual client of the server. In a sense, such a proxy could be considered a masquerading host on the protocol level used by the application. One example for such a proxy is Squid, an HTTP proxy server. To use Squid, the browser must be configured to communicate via the proxy. Any HTTP pages requested are served from the proxy cache and pages not found in the cache are fetched from the Internet by the proxy. As another example, the SUSE proxy suite (`proxy-suite`) provides a proxy for the FTP protocol.

The following section focuses on the packet filter that comes with openSUSE. For further information about packet filtering and firewalling, read the Firewall HOWTO included in the `howto` package. If this package is installed, read the HOWTO with `less /usr/share/doc/howto/en/txt/Firewall-HOWTO.gz`.

35.4 SuSEfirewall2

SuSEfirewall2 is a script that reads the variables set in `/etc/sysconfig/SuSEfirewall2` to generate a set of iptables rules. It defines three security zones, although only the first and the second one are considered in the following sample configuration:

External Zone
: Given that there is no way to control what is happening on the external network, the host needs to be protected from it. In most cases, the external network is the Internet, but it could be another insecure network, such as a WLAN.

Internal Zone
: This refers to the private network, in most cases the LAN. If the hosts on this network use IP addresses from the private range (see Section 20.1.2, “Netmasks and Routing” (page 303)), enable network address translation (NAT), so hosts on the internal network can access the external one.

Demilitarized Zone (DMZ)
: While hosts located in this zone can be reached both from the external and the internal network, they cannot access the internal network themselves. This setup can be used to put an additional line of defense in front of the internal network, because the DMZ systems are isolated from the internal network.

Any kind of network traffic not explicitly allowed by the filtering rule set is suppressed by iptables. Therefore, each of the interfaces with incoming traffic must be placed into one of the three zones. For each of the zones, define the services or protocols allowed. The rule set is only applied to packets originating from remote hosts. Locally generated packets are not captured by the firewall.

The configuration can be performed with YaST (see Section 35.4.1, “Configuring the Firewall with YaST” (page 583)). It can also be made manually in the file `/etc/sysconfig/SuSEfirewall2`, which is well commented. Additionally, a number of example scenarios are available in `/usr/share/doc/packages/SuSEfirewall2/EXAMPLES`.

35.4.1 Configuring the Firewall with YaST

IMPORTANT: Automatic Firewall Configuration

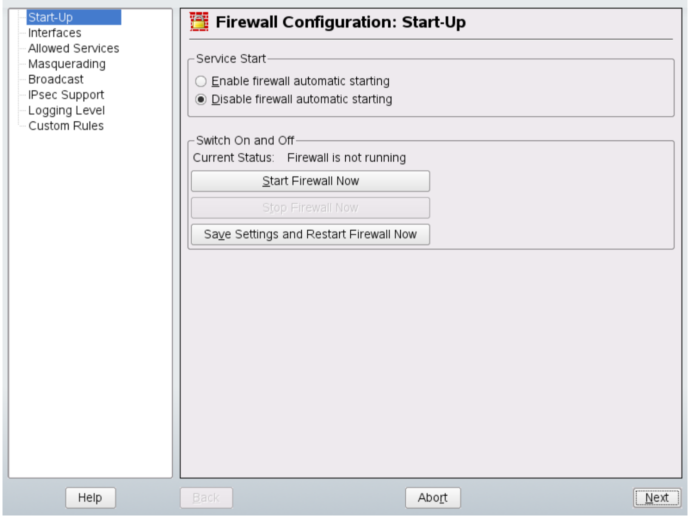
After the installation, YaST automatically starts a firewall on all configured interfaces. If a server is configured and activated on the system, YaST can modify the automatically-generated firewall configuration with the options *Open Ports on Selected Interface in Firewall* or *Open Ports on Firewall* in the server configuration modules. Some server module dialogs include a *Firewall Details* button for activating additional services and ports. The YaST firewall configuration module can be used to activate, deactivate, or reconfigure the firewall.

The YaST dialogs for the graphical configuration can be accessed from the YaST Control Center. Select *Security and Users* > *Firewall*. The configuration is divided into seven sections that can be accessed directly from the tree structure on the left side.

Start-Up

Set the start-up behavior in this dialog. In a default installation, SuSEfirewall2 is started automatically. You can also start and stop the firewall here. To implement your new settings in a running firewall, use *Save Settings and Restart Firewall Now*.

Figure 35.2 *The YaST Firewall Configuration*



Firewall Configuration: Start-Up

Service Start

○ Enable firewall automatic starting
● Disable firewall automatic starting

Switch On and Off

Current Status: Firewall is not running

Interfaces

All known network interfaces are listed here. To remove an interface from a zone, select the interface, press *Change*, and choose *No Zone Assigned*. To add an interface to a zone, select the interface, press *Change* and choose any of the available zones. You may also create a special interface with your own settings by using *Custom*.

Allowed Services

You need this option to offer services from your system to a zone from which it is protected. By default, the system is only protected from external zones. Explicitly allow the services that should be available to external hosts. After selecting the desired zone in *Allowed Services for Selected Zone*, activate the services from the list.

Masquerading

Masquerading hides your internal network from external networks, such as the Internet, while enabling hosts in the internal network to access the external network transparently. Requests from the external network to the internal one are blocked and requests from the internal network seem to be issued by the masquerading server when seen externally. If special services of an internal machine need to be available to the external network, add special redirect rules for the service.

Broadcast

In this dialog, configure the UDP ports that allow broadcasts. Add the required port numbers or services to the appropriate zone, separated by spaces. See also the file `/etc/services`.

The logging of broadcasts that are not accepted can be enabled here. This may be problematic, because Windows hosts use broadcasts to know about each other and so generate many packets that are not accepted.

IPsec Support

Configure whether the IPsec service should be available to the external network in this dialog. Configure which packets are trusted under *Details*.

Logging Level

There are two rules for the logging: accepted and not accepted packets. Packets that are not accepted are DROPPED or REJECTED. Select from *Log All*, *Log Only Critical*, or *Do Not Log Any* for both of them.

Custom Rules

Here set special firewall rules that allow connections, matching specified citeria.

When completed with the firewall configuration, exit this dialog with *Next*. A zone-oriented summary of your firewall configuration then opens. In it, check all settings. All services, ports, and protocols that have been allowed are listed in this summary. To modify the configuration, use *Back*. Press *Accept* to save your configuration.

35.4.2 Configuring Manually

The following paragraphs provide step-by-step instructions for a successful configuration. Each configuration item is marked as to whether it is relevant to firewalling or masquerading. Aspects related to the DMZ (demilitarized zone) as mentioned in the configuration file are not covered here. They are applicable only to a more complex network infrastructure found in larger organizations (corporate networks), which require extensive configuration and in-depth knowledge about the subject.

First, use the YaST module System Services (Runlevel) to enable SuSEfirewall2 in your runlevel (3 or 5 most likely). It sets the symlinks for the SuSEfirewall2_* scripts in the `/etc/init.d/rc?.d/` directories.

`FW_DEV_EXT` (firewall, masquerading)
: The device linked to the Internet. For a modem connection, enter `ppp0`. For an ISDN link, use `ippp0`. DSL connections use `dsl0`. Specify `auto` to use the interface that corresponds to the default route.

`FW_DEV_INT` (firewall, masquerading)
: The device linked to the internal, private network (such as `eth0`). Leave this blank if there is no internal network and the firewall protects only the host on which it runs.

`FW_ROUTE` (firewall, masquerading)
: If you need the masquerading function, set this to `yes`. Your internal hosts will not be visible to the outside, because their private network addresses (e.g., `192.168.x.x`) are ignored by Internet routers.

For a firewall without masquerading, only set this to `yes` if you want to allow access to the internal network. Your internal hosts need to use officially registered IP addresses in this case. Normally, however, you should *not* allow access to your internal network from the outside.

FW_MASQUERADE (masquerading)
Set this to yes if you need the masquerading function. This provides a virtually direct connection to the Internet for the internal hosts. It is more secure to have a proxy server between the hosts of the internal network and the Internet. Masquerading is not needed for services a proxy server provides.

FW_MASQ_NETS (masquerading)
Specify the hosts or networks to masquerade, leaving a space between the individual entries. For example:

```
FW_MASQ_NETS="192.168.0.0/24 192.168.10.1"
```

FW_PROTECT_FROM_INT (firewall)
Set this to yes to protect your firewall host from attacks originating in your internal network. Services are only available to the internal network if explicitly enabled. Also see FW_SERVICES_INT_TCP and FW_SERVICES_INT_UDP.

FW_SERVICES_EXT_TCP (firewall)
Enter the TCP ports that should be made available. Leave this blank for a normal workstation at home that should not offer any services.

FW_SERVICES_EXT_UDP (firewall)
Leave this blank unless you run a UDP service and want to make it available to the outside. The services that use UDP include include DNS servers, IPsec, TFTP, DHCP and others. In that case, enter the UDP ports to use.

FW_SERVICES_INT_TCP (firewall)
With this variable, define the services available for the internal network. The notation is the same as for FW_SERVICES_EXT_TCP, but the settings are applied to the *internal* network. The variable only needs to be set if FW_PROTECT_FROM_INT is set to yes.

FW_SERVICES_INT_UDP (firewall)
See FW_SERVICES_INT_TCP.

After configuring the firewall, test your setup. The firewall rule sets are created by entering SuSEfirewall2 start as root. Then use telnet, for example, from an external host to see whether the connection is actually denied. After that, review /var/log/messages, where you should see something like this:

```
Mar 15 13:21:38 linux kernel: SFW2-INext-DROP-DEFLT IN=eth0
OUT= MAC=00:80:c8:94:c3:e7:00:a0:c9:4d:27:56:08:00 SRC=192.168.10.0
```

```
DST=192.168.10.1 LEN=60 TOS=0x10 PREC=0x00 TTL=64 ID=15330 DF PROTO=TCP
SPT=48091 DPT=23 WINDOW=5840 RES=0x00 SYN URGP=0
OPT (020405B40402080A061AFEBC0000000001030300)
```

Other packages to test your firewall setup are nmap or nessus. The documentation of `nmap` is found at `/usr/share/doc/packages/nmap` and the documentation of `nessus` resides in the directory `/usr/share/doc/packages/nessus-core` after installing the respective package.

35.5 For More Information

The most up-to-date information and other documentation about the `SuSEfirewall2` package is found in `/usr/share/doc/packages/SuSEfirewall2`. The home page of the netfilter and iptables project, http://www.netfilter.org, provides a large collection of documents in many languages.

SSH—Secure Network Operations 36

With more and more computers installed in networked environments, it often becomes necessary to access hosts from a remote location. This normally means that a user sends login and password strings for authentication purposes. As long as these strings are transmitted as plain text, they could be intercepted and misused to gain access to that user account without the authorized user even knowing about it. Apart from the fact that this would open all the user's files to an attacker, the illegal account could be used to obtain administrator or `root` access or to penetrate other systems. In the past, remote connections were established with telnet, which offers no guards against eavesdropping in the form of encryption or other security mechanisms. There are other unprotected communication channels, like the traditional FTP protocol and some remote copying programs.

The SSH suite provides the necessary protection by encrypting the authentication strings (usually a login name and a password) and all the other data exchanged between the hosts. With SSH, the data flow could still be recorded by a third party, but the contents are encrypted and cannot be reverted to plain text unless the encryption key is known. So SSH enables secure communication over insecure networks, such as the Internet. The SSH flavor that comes with openSUSE is OpenSSH.

36.1 The OpenSSH Package

openSUSE installs the package OpenSSH by default. The programs ssh, scp, and sftp are then available as secure alternatives to telnet, rlogin, rsh, rcp, and ftp. In the default configuration, system access of a openSUSE system is only possible with the OpenSSH utilities and only if the firewall permits access.

36.2 The ssh Program

Using the ssh program, it is possible to log in to remote systems and work interactively. It replaces both telnet and rlogin. The slogin program is just a symbolic link pointing to ssh. For example, log in to the host `venus` with the command `ssh venus`. The host then prompts for the password on `venus`.

After successful authentication, you can work on the remote command line or use interactive applications, such as YaST. If the local username is different from the remote username, you can log in using a different login name with `ssh -l augustine venus` or `ssh augustine@venus`.

Furthermore, ssh offers the possibility to run commands on remote systems, as known from rsh. In the following example, run the command `uptime` on the host `venus` and create a directory with the name `tmp`. The program output is displayed on the local terminal of the host `jupiter`.

```
ssh venus "uptime; mkdir tmp"
Password:
1:21pm  up  2:17,  9 users,  load average: 0.15, 0.04, 0.02
```

Quotation marks are necessary here to send both instructions with one command. It is only by doing this that the second command is executed on `venus`.

36.3 scp—Secure Copy

scp copies files to a remote machine. It is a secure and encrypted substitute for rcp. For example, `scp MyLetter.tex venus:` copies the file `MyLetter.tex` from the host `jupiter` to the host `venus`. If the username on `jupiter` is different than the username on `venus`, specify the latter using the `username@host` format. Do not use `-l` option for specifying username with this command, it has a different meaning.

After the correct password is entered, scp starts the data transfer. The program displays the estimated time of arrival to the right of the progress bar. Suppress all output by giving the option `-q`.

scp also provides a recursive copying feature for entire directories. The command `scp -r src/ venus:backup/` copies the entire contents of the directory `src` including all subdirectories to the `backup` directory on the host `venus`. If this subdirectory does not exist yet, it is created automatically.

The option `-p` tells scp to leave the time stamp of files unchanged. `-C` compresses the data transfer. This minimizes the data volume to transfer, but creates a heavier burden on the processor.

36.4 sftp—Secure File Transfer

The sftp program can be used instead of scp for secure file transfer. During an sftp session, you can use many of the commands known from ftp. The sftp program may be a better choice than scp, especially when transferring data for which the filenames are unknown.

36.5 The SSH Daemon (sshd)—Server-Side

To work with the SSH client programs ssh and scp, a server, the SSH daemon, must be running in the background, listening for connections on TCP/IP port 22. The daemon generates three key pairs when starting for the first time. Each key pair consists of a private and a public key. Therefore, this procedure is referred to as public key–based. To guarantee the security of the communication via SSH, access to the private key files must be restricted to the system administrator. The file permissions are set accordingly by the default installation. The private keys are only required locally by the SSH daemon and must not be given to anyone else. The public key components (recognizable by the name extension `.pub`) are sent to the client requesting the connection. They are readable for all users.

A connection is initiated by the SSH client. The waiting SSH daemon and the requesting SSH client exchange identification data to compare the protocol and software versions and to prevent connections through the wrong port. Because a child process of the original SSH daemon replies to the request, several SSH connections can be made simultaneously.

For the communication between SSH server and SSH client, OpenSSH supports versions 1 and 2 of the SSH protocol. Version 2 of the SSH protocol is used by default.

Override this to use version 1 of the protocol with the -1 switch. To continue using version 1 after a system update, follow the instructions in /usr/share/doc/packages/openssh/README.SuSE. This document also describes how an SSH 1 environment can be transformed into a working SSH 2 environment with just a few steps.

When using version 1 of SSH, the server sends its public host key and a server key, which is regenerated by the SSH daemon every hour. Both allow the SSH client to encrypt a freely chosen session key, which is sent to the SSH server. The SSH client also tells the server which encryption method (cipher) to use.

Version 2 of the SSH protocol does not require a server key. Both sides use an algorithm according to Diffie-Helman to exchange their keys.

The private host and server keys are absolutely required to decrypt the session key and cannot be derived from the public parts. Only the SSH daemon contacted can decrypt the session key using its private keys (see man /usr/share/doc/packages/openssh/RFC.nroff). This initial connection phase can be watched closely by turning on the verbose debugging option -v of the SSH client.

The client stores all public host keys in ~/.ssh/known_hosts after its first contact with a remote host. This prevents any man-in-the-middle attacks—attempts by foreign SSH servers to use spoofed names and IP addresses. Such attacks are detected either by a host key that is not included in ~/.ssh/known_hosts or by the server's inability to decrypt the session key in the absence of an appropriate private counterpart.

It is recommended to back up the private and public keys stored in /etc/ssh/ in a secure, external location. This way, key modifications can be detected and the old ones can be used again after a reinstallation. This spares users any unsettling warnings. If it is verified that, despite the warning, it is indeed the correct SSH server, the existing entry for the system must be removed from ~/.ssh/known_hosts.

36.6 SSH Authentication Mechanisms

Now the actual authentication takes place, which, in its simplest form, consists of entering a password as mentioned above. The goal of SSH was to introduce a secure software that is also easy to use. Because it is meant to replace rsh and rlogin, SSH must also be able to provide an authentication method appropriate for daily use. SSH accomplishes this by way of another key pair, which is generated by the user. The SSH package provides a helper program for this: ssh-keygen. After entering `ssh-keygen -t rsa` or `ssh-keygen -t dsa`, the key pair is generated and you are prompted for the base filename in which to store the keys.

Confirm the default setting and answer the request for a passphrase. Even if the software suggests an empty passphrase, a text from 10 to 30 characters is recommended for the procedure described here. Do not use short and simple words or phrases. Confirm by repeating the passphrase. Subsequently, you will see where the private and public keys are stored, in this example, the files `id_rsa` and `id_rsa.pub`.

Use `ssh-keygen -p -t rsa` or `ssh-keygen -p -t dsa` to change your old passphrase. Copy the public key component (`id_rsa.pub` in the example) to the remote machine and save it to `~/.ssh/authorized_keys`. You will be asked to authenticate yourself with your passphrase the next time you establish a connection. If this does not occur, verify the location and contents of these files.

In the long run, this procedure is more troublesome than giving your password each time. Therefore, the SSH package provides another tool, ssh-agent, which retains the private keys for the duration of an X session. The entire X session is started as a child process of ssh-agent. The easiest way to do this is to set the variable `usessh` at the beginning of the `.xsession` file to `yes` and log in via a display manager, such as KDM or XDM. Alternatively, enter `ssh-agent startx`.

Now you can use ssh or scp as usual. If you have distributed your public key as described above, you are no longer prompted for your password. Take care of terminating your X session or locking it with a password protection application, such as xlock.

All the relevant changes that resulted from the introduction of version 2 of the SSH protocol are also documented in the file `/usr/share/doc/packages/openssh/README.SuSE`.

36.7 X, Authentication, and Forwarding Mechanisms

Beyond the previously described security-related improvements, SSH also simplifies the use of remote X applications. If you run `ssh` with the option `-X`, the `DISPLAY` variable is automatically set on the remote machine and all X output is exported to the remote machine over the existing SSH connection. At the same time, X applications started remotely and locally viewed with this method cannot be intercepted by unauthorized individuals.

By adding the option `-A`, the ssh-agent authentication mechanism is carried over to the next machine. This way, you can work from different machines without having to enter a password, but only if you have distributed your public key to the destination hosts and properly saved it there.

Both mechanisms are deactivated in the default settings, but can be permanently activated at any time in the systemwide configuration file `/etc/ssh/sshd_config` or the user's `~/.ssh/config`.

ssh can also be used to redirect TCP/IP connections. In the examples below, SSH is told to redirect the SMTP and the POP3 port, respectively:

```
ssh -L 25:mail.example.com:25 jupiter.example.com
```

With this command, any connection directed to `jupiter.example.com` port 25 (SMTP) is redirected to the SMTP port on `mail.example.com` via an encrypted channel. This is especially useful for those using SMTP servers without SMTP-AUTH or POP-before-SMTP features. From any arbitrary location connected to a network, e-mail can be transferred to the "home" mail server for delivery. Similarly, all POP3 requests (port 110) on `jupiter.example.com` can be forwarded to the POP3 port of `mail.example.com` with this command:

```
ssh -L 110:mail.example.com:110 jupiter.example.com
```

Both commands must be executed as `root`, because the connection is made to privileged local ports. E-mail is sent and retrieved by normal users in an existing SSH connection. The SMTP and POP3 host must be set to `localhost` for this to work. Additional information can be found in the manual pages for each of the programs described above and also in the files under `/usr/share/doc/packages/openssh`.

36.8 Configuring An SSH Daemon with YaST

To configure an sshd server with YaST run YaST and choose *Network Services* > *SSHD Configuration*. Then proceed as follows:

1. Select the ports sshd should listen on in the *SSHD TCP Ports* table. The default port number is 22. Multiple ports are allowed. To add a new port, click *Add*, enter the port number and click *OK*. To delete port, select it in the table, click *Delete* and confirm.

2. Select the features the sshd daemon should support. To disable TCP forwarding, uncheck *Allow TCP Forwarding*. Disabling TCP forwarding does not improve security unless users are also denied shell access, as they can always install their own forwarders. See Section 36.7, "X, Authentication, and Forwarding Mechanisms" (page 593) for more information about TCP forwarding.

   To disable X forwarding, uncheck *Allow X11 Forwarding*. If this option is disabled, any X11 forward requests by the client will return an error. However users can always install their own forwarders. See Section 36.7, "X, Authentication, and Forwarding Mechanisms" (page 593) for more information about X forwarding.

   In *Allow Compression* determine, whether the connection between the server and clients should be compressed. After setting these options, click *Next*.

3. In *Print Message of the day After Login* determine, whether sshd should print message from `/etc/motd` when a user logs in interactively. If you want to disable connection of a user `root`, uncheck *Permit Root Login*.

   In *Maximum Authentication Tries* enter the maximum allowed number of authentication attempts per connection. *Password Authentication* specifies whether password authentication is allowed. *RSA Authentication* specifies whether pure RSA authentication is allowed. This option applies to SSH protocol version 1 only. *Public Key Authentication* specifies whether public key authentication is allowed. This option applies to protocol version 2 only.

4. Click *Accept* to save the configuration.

Managing X.509 Certification 37

An increasing number of authentication mechanisms are based on cryptographic procedures. Digital certificates that assign cryptographic keys to their owners play an important role in this context. These certificates are used for communication and can also be found, for example, on company ID cards. The generation and administration of certificates is mostly handled by official institutions that offer this as a commercial service. In some cases, however, it may make sense to carry out these tasks yourself, for example, if a company does not wish to pass personal data to third parties.

YaST provides two modules for certification, which offer basic management functions for digital X.509 certificates. The following sections explain the basics of digital certification and how to use YaST to create and administer certificates of this type. For more detailed information, refer to `http://www.ietf.org/html.charters/pkix-charter.html`.

37.1 The Principles of Digital Certification

Digital certification uses cryptographic processes to encrypt data, protecting the data from access by unauthorized people. The user data is encrypted using a second data record, or *key*. The key is applied to the user data in a mathematical process, producing an altered data record in which the original content can no longer be identified. Asymmetrical encryption is now in general use (*public key method*). Keys always occur in pairs:

Private Key
The private key must be kept safely by the key owner. Accidental publication of the private key compromises the key pair and renders it useless.

Public Key
The key owner circulates the public key for use by third parties.

37.1.1 Key Authenticity

Because the public key process is in widespread use, there are many public keys in circulation. Successful use of this system requires that every user be sure that a public key actually belongs to the assumed owner. The assignment of users to public keys is confirmed by trustworthy organizations with public key certificates. Such certificates contain the name of the key owner, the corresponding public key, and the electronic signature of the person issuing the certificate.

Trustworthy organizations that issue and sign public key certificates are usually part of a certification infrastructure that is also responsible for the other aspects of certificate management, such as publication, withdrawal, and renewal of certificates. An infrastructure of this kind is generally referred to as a *public key infrastructure* or *PKI*. One familiar PKI is the *OpenPGP* standard in which users publish their certificates themselves without central authorization points. These certificates become trustworthy when signed by other parties in the "web of trust."

The *X.509 Public Key Infrastructure* (PKIX) is an alternative model defined by the *IETF* (Internet Engineering Task Force) that serves as a model for almost all publicly-used PKIs today. In this model, authentication is made by *certificate authorities* (CA) in a hierarchical tree structure. The root of the tree is the root CA, which certifies all sub-CAs. The lowest level of sub-CAs issue user certificates. The user certificates are trustworthy by certification that can be traced to the root CA.

The security of such a PKI depends on the trustworthiness of the CA certificates. To make certification practices clear to PKI customers, the PKI operator defines a *certification practice statement* (CPS) that defines the procedures for certificate management. This should ensure that the PKI only issues trustworthy certificates.

37.1.2 X.509 Certificates

An X.509 certificate is a data structure with several fixed fields and, optionally, additional extensions. The fixed fields mainly contain the name of the key owner, the public key, and the data relating to the issuing CA (name and signature). For security reasons, a certificate should only have a limited period of validity, so a field is also provided for this date. The CA guarantees the validity of the certificate in the specified period. The CPS usually requires the PKI (the issuing CA) to create and distribute a new certificate before expiration.

The extensions can contain any additional information. An application is only required to be able to evaluate an extension if it is identified as *critical*. If an application does not recognize a critical extension, it must reject the certificate. Some extensions are only useful for a specific application, such as signature or encryption.

Table 37.1 shows the fields of a basic X.509 certificate in version 3.

Table 37.1 *X.509v3 Certificate*

Field	Content
Version	The version of the certificate, for example, v3
Serial Number	Unique certificate ID (an integer)
Signature	The ID of the algorithm used to sign the certificate
Issuer	Unique name (DN) of the issuing authority (CA)
Validity	Period of validity
Subject	Unique name (DN) of the owner
Subject Public Key Info	Public key of the owner and the ID of the algorithm
Issuer Unique ID	Unique ID of the issuing CA (optional)
Subject Unique ID	Unique ID of the owner (optional)

Field	Content
Extensions	Optional additional information, such as “KeyUsage” or “BasicConstraints”

37.1.3 Blocking X.509 Certificates

If a certificate becomes untrustworthy before it has expired, it must be blocked immediately. This can be needed if, for example, the private key has accidentally been made public. Blocking certificates is especially important if the private key belongs to a CA rather than a user certificate. In this case, all user certificates issued by the relevant CA must be blocked immediately. If a certificate is blocked, the PKI (the responsible CA) must make this information available to all those involved using a *certificate revocation list* (CRL).

These lists are supplied by the CA to public CRL distribution points (CDPs) at regular intervals. The CDP can optionally be named as an extension in the certificate, so a checker can fetch a current CRL for validation purposes. One way to do this is the *online certificate status protocol* (OCSP). The authenticity of the CRLs is ensured with the signature of the issuing CA. Table 37.2, “X.509 Certificate Revocation List (CRL)” (page 600) shows the basic parts of a X.509 CRL.

Table 37.2 *X.509 Certificate Revocation List (CRL)*

Field	Content
Version	The version of the CRL, such as v2
Signature	The ID of the algorithm used to sign the CRL
Issuer	Unique name (DN) of the publisher of the CRL (usually the issuing CA)
This Update	Time of publication (date, time) of this CRL
Next Update	Time of publication (date, time) of the next CRL

Field	Content
List of revoked certificates	Every entry contains the serial number of the certificate, the time of revocation, and optional extensions (CRL entry extensions)
Extensions	Optional CRL extensions

37.1.4 Repository for Certificates and CRLs

The certificates and CRLs for a CA must be made publicly accessible using a *repository*. Because the signature protects the certificates and CRLs from being forged, the repository itself does not need to be secured in a special way. Instead, it tries to grant the simplest and fastest access possible. For this reason, certificates are often provided on an LDAP or HTTP server. Find explanations about LDAP in Chapter 26, *LDAP—A Directory Service* (page 409). Chapter 30, *The Apache HTTP Server* (page 481) contains information about the HTTP server.

37.1.5 Proprietary PKI

YaST contains modules for the basic management of X.509 certificates. This mainly involves the creation of CAs, sub-CAs, and their certificates. The services of a PKI go far beyond simply creating and distributing certificates and CRLs. The operation of a PKI requires a well-conceived administrative infrastructure allowing continuous update of certificates and CRLs. This infrastructure is provided by commercial PKI products and can also be partly automated. YaST provides tools for creating and distributing CAs and certificates, but cannot currently offer this background infrastructure. To set up a small PKI, you can use the available YaST modules. However, you should use commercial products to set up an “official” or commercial PKI.

37.2 YaST Modules for CA Management

YaST provides two modules for basic CA management. The primary management tasks with these modules are explained here.

37.2.1 Creating a Root CA

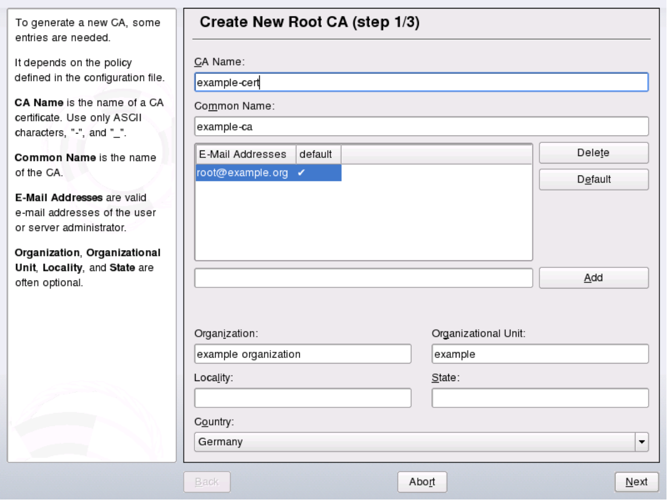
The first step when setting up a PKI is to create a root CA. Do the following:

1. Start YaST and go to *Security and Users* > *CA Management*.

2. Click *Create Root CA*.

3. Enter the basic data for the CA in the first dialog, shown in Figure 37.1, "YaST CA Module—Basic Data for a Root CA" (page 602). The text fields have the following meanings:

Figure 37.1 *YaST CA Module—Basic Data for a Root CA*



To generate a new CA, some entries are needed.

It depends on the policy defined in the configuration file.

CA Name is the name of a CA certificate. Use only ASCII characters, "-", and "_".

Common Name is the name of the CA.

E-Mail Addresses are valid e-mail addresses of the user or server administrator.

Organization, **Organizational Unit**, **Locality**, and **State** are often optional.

Create New Root CA (step 1/3)

CA Name: example-cert

Common Name: example-ca

E-Mail Addresses	default
root@example.org	✓

Organization: example organization

Organizational Unit: example

Locality:

State:

Country: Germany

CA Name
Enter the technical name of the CA. Directory names, among other things, are derived from this name, which is why only the characters listed in the help can be used. The technical name is also displayed in the overview when the module is started.

Common Name
Enter the name to use to refer to the CA.

E-Mail Addresses
Several e-mail addresses can be entered that can be seen by the CA user. This can be helpful for inquiries.

Country
Select the country where the CA is operated.

Organisation, *Organisational Unit*, *Locality*, *State*
Optional values

4 Click *Next*.

5 Enter a password in the second dialog. This password is always required when using the CA—when creating a sub-CA or generating certificates. The text fields have the following meaning:

Key Length
Key Length contains a meaningful default and does not generally need to be changed unless an application cannot deal with this key length.

Valid Period (days)
The *Valid Period* in the case of a CA defaults to 3650 days (roughly ten years). This long period makes sense because the replacement of a deleted CA involves an enormous administrative effort.

Clicking *Advanced Options* opens a dialog for setting different attributes from the X.509 extensions (Figure 37.4, “YaST CA Module—Extended Settings” (page 608)). These values have rational default settings and should only be changed if you are really sure of what you are doing.

6 YaST displays the current settings for confirmation. Click *Create*. The root CA is created then appears in the overview.

TIP

In general, it is best not to allow user certificates to be issued by the root CA. It is better to create at least one sub-CA and create the user certificates from there. This has the advantage that the root CA can be kept isolated and secure, for example, on an isolated computer on secure premises. This makes it very difficult to attack the root CA.

37.2.2 Creating or Revoking a Sub-CA

A sub-CA is created in exactly the same way as a root CA. Do the following:

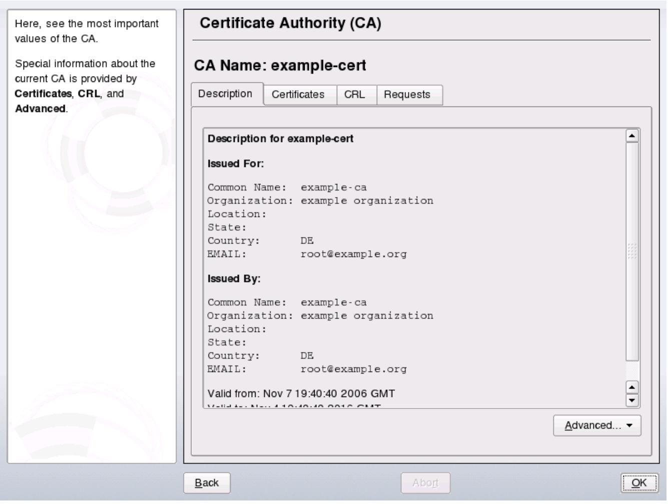
1. Start YaST and open the CA module.
2. Select the required CA and click *Enter CA*.

   NOTE

   The validity period for a sub-CA must be fully within the validity period of the “parent” CA. Because a sub-CA is always created after the “parent” CA, the default value leads to an error message. To avoid this, enter a permissible value for the period of validity.

3. Enter the password if you entered a CA the first time. YaST displays the CA key information in the tab *Description* (see Figure 37.2).

Figure 37.2 *YaST CA Module—Using a CA*



Here, see the most important values of the CA.

Special information about the current CA is provided by **Certificates**, **CRL**, and **Advanced**.

Certificate Authority (CA)

CA Name: example-cert

Description for example-cert

Issued For:

```
Common Name:  example-ca
Organization: example organization
Location:
State:
Country:      DE
EMAIL:        root@example.org
```

Issued By:

```
Common Name:  example-ca
Organization: example organization
Location:
State:
Country:      DE
EMAIL:        root@example.org
```

Valid from: Nov 7 19:40:40 2006 GMT

Valid to: Nov 4 19:40:40 2016 GMT

4 Click *Advanced* and select *Create SubCA*. This opens the same dialog as for creating a root CA.

5 Proceed as described in Section 37.2.1, “Creating a Root CA” (page 602).

6 Select the tab *Certificates*. Reset compromised or otherwise unwanted sub-CAs here using *Revoke*. Revocation is not enough to deactivate a sub-CA on its own. Also publish revoked sub-CAs in a CRL. The creation of CRLs is described in Section 37.2.5, “Creating CRLs ” (page 609).

7 Finish with *Ok*

37.2.3 Creating or Revoking User Certificates

Creating client and server certificates is very similar to the one for creating CAs in Section 37.2.1, “Creating a Root CA” (page 602). The same principles apply here. In certificates intended for e-mail signature, the e-mail address of the sender (the private key owner) should be contained in the certificate to enable the e-mail program to assign the correct certificate. For certificate assignment during encryption, it is necessary for the e-mail address of the recipient (the public key owner) to be included in the certificate. In the case of server and client certificates, the hostname of the server must be entered in the *Common Name* field. The default validity period for certificates is 365 days.

To create client and server certificates, do the following:

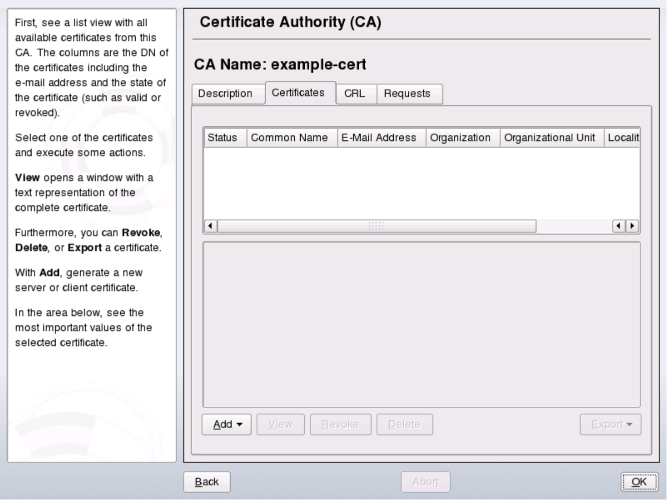
1. Start YaST and open the CA module.

2. Select the required CA and click *Enter CA*.

3. Enter the password if entering a CA for the first time. YaST displays the CA key information in the *Description* tab.

4. Click *Certificates* (see Figure 37.3, “Certificates of a CA” (page 606)).

Figure 37.3 *Certificates of a CA*



First, see a list view with all available certificates from this CA. The columns are the DN of the certificates including the e-mail address and the state of the certificate (such as valid or revoked).

Select one of the certificates and execute some actions.

View opens a window with a text representation of the complete certificate.

Furthermore, you can **Revoke**, **Delete**, or **Export** a certificate.

With **Add**, generate a new server or client certificate.

In the area below, see the most important values of the selected certificate.

Certificate Authority (CA)

CA Name: example-cert

5. Click *Add* > *Add Server Certificate* and create a server certificate.

6. Click *Add* > *Add Client Certificate* and create a client certificate. Do not forget to enter an e-mail address.

7. Finish with *Ok*

To revoke compromised or otherwise unwanted certificates, do the following:

1 Start YaST and open the CA module.

2 Select the required CA and click *Enter CA*.

3 Enter the password if entering a CA the first time. YaST displays the CA key information in the *Description* tab.

4 Click *Certificates* (see Section 37.2.2, "Creating or Revoking a Sub-CA" (page 604).)

5 Select the certificate to revoke and click *Revoke*.

6 Choose a reason to revoke this certificate

7 Finish with *Ok*.

NOTE

Revocation alone is not enough to deactivate a certificate. Also publish revoked certificates in a CRL. Section 37.2.5, "Creating CRLs " (page 609) explains how to create CRLs. Revoked certificates can be completely removed after publication in a CRL with *Delete*.

37.2.4 Changing Default Values

The previous sections explained how to create sub-CAs, client certificates, and server certificates. Special settings are used in the extensions of the X.509 certificate. These settings have been given rational defaults for every certificate type and do not normally need to be changed. However, it may be that you have special requirements for these extensions. In this case, it may make sense to adjust the defaults. Otherwise, start from scratch every time you create a certificate.

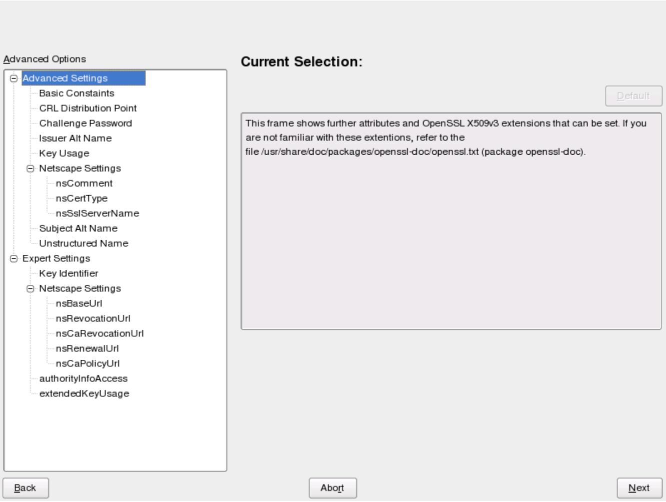
1 Start YaST and open the CA module.

2 Enter the required CA, as described in Section 37.2.2, "Creating or Revoking a Sub-CA" (page 604).

3 Click *Advanced* > *Edit Defaults*.

4 Choose the type the settings to change. The dialog for changing the defaults, shown in Figure 37.4, “YaST CA Module—Extended Settings” (page 608), then opens.

Figure 37.4 *YaST CA Module—Extended Settings*



This frame shows further attributes and OpenSSL X509v3 extensions that can be set. If you are not familiar with these extentions, refer to the file /usr/share/doc/packages/openssl-doc/openssl.txt (package openssl-doc).

5 Change the associated value on the right side and set or delete the critical setting with *critical*.

6 Click *Next* to see a short summary.

7 Finish your changes with *Save*.

TIP

All changes to the defaults only affect objects created after this point. Already existing CAs and certificates remain unchanged.

37.2.5 Creating CRLs

If compromised or otherwise unwanted certificates should be excluded from further use, they must first be revoked. The procedure for this is explained in Section 37.2.2, "Creating or Revoking a Sub-CA" (page 604) (for sub-CAs) and Section 37.2.3, "Creating or Revoking User Certificates" (page 605) (for user certificates). After this, a CRL must be created and published with this information.

The system maintains only one CRL for each CA. To create or update this CRL, do the following:

1. Start YaST and open the CA module.
2. Enter the required CA, as described in Section 37.2.2, "Creating or Revoking a Sub-CA" (page 604).
3. Click *CRL*. The dialog that opens displays a summary of the last CRL of this CA.
4. Create a new CRL with *Generate CRL* if you have revoked new sub-CAs or certificates since its creation.
5. Specify the period of validity for the new CRL (default: 30 days).
6. Click *OK* to create and display the CRL. Afterwards, you must publish this CRL.

TIP

Applications that evaluate CRLs reject every certificate if CRL is not available or expired. As a PKI provider, it is your duty always to create and publish a new CRL before the current CRL expires (period of validity). YaST does not provide a function for automating this procedure.

37.2.6 Exporting CA Objects to LDAP

The executing computer should be configured with the YaST LDAP client for LDAP export. This provides LDAP server information at runtime that can be used when completing dialog fields. Otherwise, although export may be possible, all LDAP data must be entered manually. You must always enter several passwords (see Table 37.3, "Passwords during LDAP Export" (page 610)).

***Table 37.3** Passwords during LDAP Export*

Password	Meaning
LDAP Password	Authorizes the user to make entries in the LDAP tree.
Certificate Password	Authorizes the user to export the certificate.
New Certificate Password	The PKCS12 format is used during LDAP export. This format forces the assignment of a new password for the exported certificate.

Certificates, CAs, and CRLs can be exported to LDAP.

Exporting a CA to LDAP

To export a CA, enter the CA as described in Section 37.2.2, "Creating or Revoking a Sub-CA" (page 604). Select *Extended* > *Export to LDAP* in the subsequent dialog, which opens the dialog for entering LDAP data. If your system has been configured with the YaST LDAP client, the fields are already partly completed. Otherwise, enter all the data manually. Entries are made in LDAP in a separate tree with the attribute "caCertificate".

Exporting a Certificate to LDAP

Enter the CA containing the certificate to export then select *Certificates*. Select the required certificate from the certificate list in the upper part of the dialog and select *Export* > *Export to LDAP*. The LDAP data is entered here in the same way as for CAs. The certificate is saved with the corresponding user object in the LDAP tree with the attributes "userCertificate" (PEM format) and "userPKCS12" (PKCS12 format).

Exporting a CRL to LDAP

Enter the CA containing the CRL to export and select *CRL*. If desired, then create a new CRL and export this with *Export* > *To LDAP*. The LDAP data is also entered here in the same way as with CAs. Entries are then made in the LDAP at the same point as the associated CA, but using the "certificateRevocationList" attribute.

37.2.7 Exporting CA Objects as a File

If you have set up a repository on the computer for administering CAs, you can use this option to create the CA objects directly as a file at the correct location. Different output formats are available, such as PEM, DER, and PKCS12. In the case of PEM, it is also possible to choose whether a certificate should be exported with or without key and whether the key should be encrypted. In the case of PKCS12, it is also possible to export the certification path.

Export a file in the same way for certificates, CAs, and CRLs as with LDAP, described in Section 37.2.6, "Exporting CA Objects to LDAP " (page 609), except you should select *Export as File* instead of *Export to LDAP*. This then takes you to a dialog for selecting the required output format and entering the password and filename. The certificate is stored at the required location after you click *OK*.

TIP

You can select any storage location in the file system. This option can also be used to save CA objects on a transport medium, such as a USB stick. The `/media` directory generally holds any type of drive except the hard drive of your system.

37.2.8 Importing Common Server Certificates

If you have exported a server certificate with YaST to your media on an isolated CA management computer, you can import this certificate on a server as a *common server certificate*. Do this during installation or at a later point with YaST.

NOTE

You need one of the PKCS12 formats to import your certificate successfully.

The general server certificate is stored in `/etc/ssl/servercerts` and can be used there by any CA-supported service. When this certificate expires, it can easily be replaced using the same mechanisms. To get things functioning with the replaced certificate, restart the participating services.

TIP

If you select *Import* here, you can select the source in the file system. This option can also be used to import certificates from a transport medium, such as a USB stick.

To import a common server certificate, do the following:

1. Start YaST and open *Common Server Certificate* under *Security and Users*

2. View the data for the current certificate in the description field after YaST has been started.

3. Select *Import* and the certificate file.

4. Enter the password and click *Next*. The certificate is imported then displayed in the description field.

5. Close YaST with *Finish*.

Network Authentication—Kerberos 38

An open network provides no means to ensure that a workstation can identify its users properly except the usual password mechanisms. In common installations, the user must enter the password each time a service inside the network is accessed. Kerberos provides an authentication method with which a user registers once then is trusted in the complete network for the rest of the session. To have a secure network, the following requirements must be met:

- Have all users prove their identity for each desired service and make sure that no one can take the identity of someone else.

- Make sure that each network server also proves its identity. Otherwise an attacker might be able to impersonate the server and obtain sensitive information transmitted to the server. This concept is called *mutual authentication*, because the client authenticates to the server and vice versa.

Kerberos helps you meet these requirements by providing strongly encrypted authentication. The following shows how this is achieved. Only the basic principles of Kerberos are discussed here. For detailed technical instruction, refer to the documentation provided with your implementation of Kerberos.

38.1 Kerberos Terminology

The following glossary defines some Kerberos terminology.

credential
: Users or clients need to present some kind of credentials that authorize them to request services. Kerberos knows two kinds of credentials—tickets and authenticators.

ticket
: A ticket is a per-server credential used by a client to authenticate at a server from which it is requesting a service. It contains the name of the server, the client's name, the client's Internet address, a time stamp, a lifetime, and a random session key. All this data is encrypted using the server's key.

authenticator
: Combined with the ticket, an authenticator is used to prove that the client presenting a ticket is really the one it claims to be. An authenticator is built of the client's name, the workstation's IP address, and the current workstation's time all encrypted with the session key only known to the client and the server from which it is requesting a service. An authenticator can only be used once, unlike a ticket. A client can build an authenticator itself.

principal
: A Kerberos principal is a unique entity (a user or service) to which it can assign a ticket. A principal consists of the following components:

- **Primary**—the first part of the principal, which can be the same as your username in the case of a user.

- **Instance**—some optional information characterizing the primary. This string is separated from the primary by a `/`.

- **Realm**—this specifies your Kerberos realm. Normally, your realm is your domain name in uppercase letters.

mutual authentication
: Kerberos ensures that both client and server can be sure of each others identity. They share a session key, which they can use to communicate securely.

session key
: Session keys are temporary private keys generated by Kerberos. They are known to the client and used to encrypt the communication between the client and the server for which it requested and received a ticket.

replay

Almost all messages sent in a network can be eavesdropped, stolen, and resent. In the Kerberos context, this would be most dangerous if an attacker manages to obtain your request for a service containing your ticket and authenticator. He could then try to resend it (*replay*) to impersonate you. However, Kerberos implements several mechanisms to deal with that problem.

server or service

Service is used to refer to a specific action to perform. The process behind this action is referred to as a *server*.

38.2 How Kerberos Works

Kerberos is often called a third party trusted authentication service, which means all its clients trust Kerberos's judgment of another client's identity. Kerberos keeps a database of all its users and their private keys.

To ensure Kerberos is worth all the trust put in it, run both the authentication and ticket-granting server on a dedicated machine. Make sure that only the administrator can access this machine physically and over the network. Reduce the (networking) services run on it to the absolute minimum—do not even run sshd.

38.2.1 First Contact

Your first contact with Kerberos is quite similar to any login procedure at a normal networking system. Enter your username. This piece of information and the name of the ticket-granting service are sent to the authentication server (Kerberos). If the authentication server knows about your existence, it generates a random session key for further use between your client and the ticket-granting server. Now the authentication server prepares a ticket for the ticket-granting server. The ticket contains the following information—all encrypted with a session key only the authentication server and the ticket-granting server know:

- The names both of the client and the ticket-granting server

- The current time

- A lifetime assigned to this ticket

- The client's IP address
- The newly-generated session key

This ticket is then sent back to the client together with the session key, again in encrypted form, but this time the private key of the client is used. This private key is only known to Kerberos and the client, because it is derived from your user password. Now that the client has received this response, you are prompted for your password. This password is converted into the key that can decrypt the package sent by the authentication server. The package is “unwrapped” and password and key are erased from the workstation's memory. As long as the lifetime given to the ticket used to obtain other tickets does not expire, your workstation can prove your identity.

38.2.2 Requesting a Service

To request a service from any server in the network, the client application needs to prove its identity to the server. Therefore, the application generates an authenticator. An authenticator consists of the following components:

- The client's principal
- The client's IP address
- The current time
- A checksum (chosen by the client)

All this information is encrypted using the session key that the client has already received for this special server. The authenticator and the ticket for the server are sent to the server. The server uses its copy of the session key to decrypt the authenticator, which gives it all information needed about the client requesting its service to compare it to that contained in the ticket. The server checks if the ticket and the authenticator originate from the same client.

Without any security measures implemented on the server side, this stage of the process would be an ideal target for replay attacks. Someone could try to resend a request stolen off the net some time before. To prevent this, the server does not accept any request with a time stamp and ticket received previously. In addition to that, a request with a time stamp differing too much from the time the request is received is ignored.

38.2.3 Mutual Authentication

Kerberos authentication can be used in both directions. It is not only a question of the client being the one it claims to be. The server should also be able to authenticate itself to the client requesting its service. Therefore, it sends some kind of authenticator itself. It adds one to the checksum it received in the client's authenticator and encrypts it with the session key, which is shared between it and the client. The client takes this response as a proof of the server's authenticity and they both start cooperating.

38.2.4 Ticket Granting—Contacting All Servers

Tickets are designed to be used for one server at a time. This implies that you have to get a new ticket each time you request another service. Kerberos implements a mechanism to obtain tickets for individual servers. This service is called the “ticket-granting service”. The ticket-granting service is a service just like any other service mentioned before, so uses the same access protocols that have already been outlined. Any time an application needs a ticket that has not already been requested, it contacts the ticket-granting server. This request consists of the following components:

- The requested principal

- The ticket-granting ticket

- An authenticator

Like any other server, the ticket-granting server now checks the ticket-granting ticket and the authenticator. If they are considered valid, the ticket-granting server builds a new session key to be used between the original client and the new server. Then the ticket for the new server is built, containing the following information:

- The client's principal

- The server's principal

- The current time

- The client's IP address

- The newly-generated session key

The new ticket is assigned a lifetime, which is the lesser of the remaining lifetime of the ticket-granting ticket and the default for the service. The client receives this ticket and the session key, which are sent by the ticket-granting service, but this time the answer is encrypted with the session key that came with the original ticket-granting ticket. The client can decrypt the response without requiring the user's password when a new service is contacted. Kerberos can thus acquire ticket after ticket for the client without bothering the user more than once at login time.

38.2.5 Compatibility to Windows 2000

Windows 2000 contains a Microsoft implementation of Kerberos 5. Because openSUSE® uses the MIT implementation of Kerberos 5, find useful information and guidance in the MIT documentation. See Section 38.4, “For More Information” (page 619).

38.3 Users' View of Kerberos

Ideally, a user's one and only contact with Kerberos happens during login at the workstation. The login process includes obtaining a ticket-granting ticket. At logout, a user's Kerberos tickets are automatically destroyed, which makes it difficult for anyone else to impersonate this user. The automatic expiration of tickets can lead to a somewhat awkward situation when a user's login session lasts longer than the maximum lifespan given to the ticket-granting ticket (a reasonable setting is 10 hours). However, the user can get a new ticket-granting ticket by running `kinit`. Enter the password again and Kerberos obtains access to desired services without additional authentication. To get a list of all the tickets silently acquired for you by Kerberos, run `klist`.

Here is a short list of some applications that use Kerberos authentication. These applications can be found under `/usr/lib/mit/bin` or `/usr/lib/mit/sbin`. They all have the full functionality of their common UNIX and Linux brothers plus the additional bonus of transparent authentication managed by Kerberos:

- telnet, telnetd
- rlogin

- rsh, rcp, rshd
- ftp, ftpd
- ksu

You no longer have to enter your password for using these applications because Kerberos has already proven your identity. ssh, if compiled with Kerberos support, can even forward all the tickets acquired for one workstation to another one. If you use ssh to log in to another workstation, ssh makes sure that the encrypted contents of the tickets are adjusted to the new situation. Simply copying tickets between workstations is not sufficient because the ticket contains workstation-specific information (the IP address). XDM, GDM, and KDM offer Kerberos support, too. Read more about the Kerberos network applications in *Kerberos V5 UNIX User's Guide* at `http://web.mit.edu/kerberos`

38.4 For More Information

The official site of the MIT Kerberos is `http://web.mit.edu/kerberos`. There, find links to any other relevant resource concerning Kerberos, including Kerberos installation, user, and administration guides.

The paper at `ftp://athena-dist.mit.edu/pub/kerberos/doc/usenix.PS` gives quite an extensive insight to the basic principles of Kerberos without being too difficult to read. It also provides a lot of opportunities for further investigation and reading about Kerberos.

The official Kerberos FAQ is available at `http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html`. The book *Kerberos—A Network Authentication System* by Brian Tung (ISBN 0-201-37924-4) offers extensive information.

Installing and Administering Kerberos

39

A Kerberos environment as described in Chapter 38, *Network Authentication—Kerberos* (page 613) consists of several different components. A key distribution center (KDC) holds the central database with all Kerberos-relevant data. All clients rely on the KDC for proper authentication across the network. Both the KDC and the clients need to be configured to match your setup:

General Preparations
: Check your network setup and make sure it meets the minimum requirements outlined in Section 39.1, "Kerberos Network Topology" (page 622). Choose an appropriate realm for your Kerberos setup, see Section 39.2, "Choosing the Kerberos Realms" (page 623). Carefully set up the machine that is to serve as the KDC and apply tight security, see Section 39.3, "Setting Up the KDC Hardware" (page 624). Set up a reliable time source in your network to make sure all tickets contain valid timestamps, see Section 39.4, "Configuring Time Synchronization" (page 625).

Basic Configuration
: Configure the KDC and the clients, see Section 39.5, "Configuring the KDC" (page 626) and Section 39.6, "Configuring Kerberos Clients" (page 628). Enable remote administration for your Kerberos service, so you do not need physical access to your KDC machine, see Section 39.7, "Configuring Remote Kerberos Administration" (page 634). Create service principals for every service in your realm, see Section 39.8, "Creating Kerberos Service Principals" (page 636).

Enabling Kerberos Authentication
: Various services in your network can make use of Kerberos. To add Kerberos password-checking to applications using PAM, proceed as outlined in Section 39.9, "Enabling PAM Support for Kerberos" (page 637). To configure SSH or LDAP

with Kerberos authentication, proceed as outlined in Section 39.10, “Configuring SSH for Kerberos Authentication” (page 638) and Section 39.11, “Using LDAP and Kerberos” (page 639).

39.1 Kerberos Network Topology

Any Kerberos environment must meet the following requirements to be fully functional:

- Provide a DNS server for name resolution across your network, so clients and servers can locate each other. Refer to Chapter 22, *The Domain Name System* (page 353) for information on DNS setup.

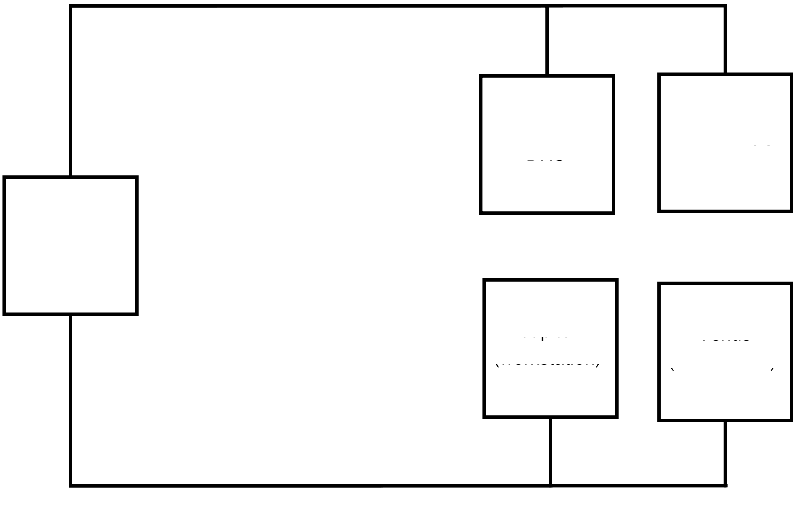
- Provide a time server in your network. Using exact time stamps is crucial to a Kerberos setup, because valid Kerberos tickets must contain correct time stamps. Refer to Chapter 24, *Time Synchronization with NTP* (page 393) for information on NTP setup.

- Provide a key distribution center (KDC) as the center piece of the Kerberos architecture. It holds the Kerberos database. Use the tightest possible security policy on this machine to prevent any attacks on this machine compromising your entire infrastructure.

- Configure the client machines to use Kerberos authentication.

The following figure depicts a simple example network with just the minimum components needed to build a Kerberos infrastructure. Depending on the size and topology of your deployment, you might need to use a different setup.

Figure 39.1 Kerberos Network Topology



TIP: Configuring Subnet Routing

For a setup similar to the one in Figure 39.1, “Kerberos Network Topology” (page 623), configure routing between the two subnets (192.168.1.0/24 and 192.168.2.0/24). Refer to Section “Configuring Routing” (page 320) for more information on configuring routing with YaST.

39.2 Choosing the Kerberos Realms

The domain of a Kerberos installation is called a realm and is identified by a name, such as `EXAMPLE.COM` or simply `ACCOUNTING`. Kerberos is case-sensitive, so `example.com` is actually a different realm than `EXAMPLE.COM`. Use the case you prefer. It is common practice, however, to use uppercase realm names.

It is also a good idea to use your DNS domain name (or a subdomain, such as `ACCOUNTING.EXAMPLE.COM`). As shown below, your life as an administrator can be much easier if you configure your Kerberos clients to locate the KDC and other

Kerberos services via DNS. To do so, it is helpful if your realm name is a subdomain of your DNS domain name.

Unlike the DNS name space, Kerberos is not hierarchical. You cannot set up a realm named `EXAMPLE.COM`, have two “subrealms” named `DEVELOPMENT` and `ACCOUNTING` underneath it, and expect the two subordinate realms to somehow inherit principals from `EXAMPLE.COM`. Instead, you would have three separate realms for which you would have to configure crossrealm authentication for users from one realm to interact with servers or other users from another realm.

For the sake of simplicity, assume you are setting up just one realm for your entire organization. For the remainder of this section, the realm name `EXAMPLE.COM` is used in all examples.

39.3 Setting Up the KDC Hardware

The first thing required to use Kerberos is a machine that acts as the key distribution center, or KDC for short. This machine holds the entire Kerberos user database with passwords and all information.

The KDC is the most important part of your security infrastructure—if someone breaks into it, all user accounts and all of your infrastructure protected by Kerberos is compromised. An attacker with access to the Kerberos database can impersonate any principal in the database. Tighten security for this machine as much as possible:

1. Put the server machine into a physically secured location, such as a locked server room to which only a very few people have access.

2. Do not run any network applications on it except the KDC. This includes servers and clients—for example, the KDC should not import any file systems via NFS or use DHCP to retrieve its network configuration.

3. Install a minimal system first then check the list of installed packages and remove any unneeded packages. This includes servers, such as inetd, portmap, and cups, as well as anything X-based. Even installing an SSH server should be considered a potential security risk.

4. No graphical login is provided on this machine as an X server is a potential security risk. Kerberos provides its own administration interface.

5. Configure /etc/nsswitch.conf to use only local files for user and group lookup. Change the lines for passwd and group to look like this:

   ```
   passwd:         files
   group:          files
   ```

   Edit the passwd, group, and shadow files in /etc and remove the lines that start with a + character (these are for NIS lookups).

6. Disable all user accounts except root's account by editing /etc/shadow and replacing the hashed passwords with * or ! characters.

39.4 Configuring Time Synchronization

To use Kerberos successfully, make sure that all system clocks within your organization are synchronized within a certain range. This is important because Kerberos protects against replayed credentials. An attacker might be able to observe Kerberos credentials on the network and reuse them to attack the server. Kerberos employs several defenses to prevent this. One of them is that it puts time stamps into its tickets. A server receiving a ticket with a time stamp that differs from the current time rejects the ticket.

Kerberos allows a certain leeway when comparing time stamps. However, computer clocks can be very inaccurate in keeping time—it is not unheard of for PC clocks to lose or gain half an hour over the course of a week. For this reason, configure all hosts on the network to synchronize their clocks with a central time source.

A simple way to do so is by installing an NTP time server on one machine and having all clients synchronize their clocks with this server. Do this either by running an NTP daemon in client mode on all these machines or by running ntpdate once a day from all clients (this solution probably works for a small number of clients only). The KDC itself needs to be synchronized to the common time source as well. Because running an NTP daemon on this machine would be a security risk, it is probably a good idea to do this by running ntpdate via a cron entry. To configure your machine as an NTP client, proceed as outlined in Section 24.1, “Configuring an NTP Client with YaST” (page 393).

It is also possible to adjust the maximum deviation Kerberos allows when checking time stamps. This value (called *clock skew*) can be set in the `krb5.conf` file as described in Section "Adjusting the Clock Skew" (page 633).

39.5 Configuring the KDC

This section covers the initial configuration and installation of the KDC, including the creation of an administrative principal. This procedure is consists of several steps:

1. **Install the RPMs** On a machine designated as the KDC, install special software packages. Use YaST to install the `krb5`, `krb5-server` and `krb5-client` packages.

2. **Adjust the Configuration Files** The configuration files `/etc/krb5.conf` and `/var/lib/kerberos/krb5kdc/kdc.conf` must be adjusted for your scenario. These files contain all information on the KDC.

3. **Create the Kerberos Database** Kerberos keeps a database of all principal identifiers and the secret keys of all principals that need to be authenticated. Refer to Section 39.5.1, "Setting Up the Database" (page 627) for details.

4. **Adjust the ACL Files: Add Administrators** The Kerberos database on the KDC can be managed remotely. To prevent unauthorized principals from tampering with the database, Kerberos uses access control lists. You must explicitly enable remote access for the administrator principal to enable him to manage the database. The Kerberos ACL file is located under `/var/lib/kerberos/krb5kdc/kadm5.acl`. Refer to Section 39.7, "Configuring Remote Kerberos Administration" (page 634) for details.

5. **Adjust the Kerberos Database: Add Administrators** You need at least one administrative principal to run and administer Kerberos. This principal must be added before starting the KDC. Refer to Section 39.5.2, "Creating a Principal" (page 628) for details.

6. **Start the Kerberos Daemon** Once the KDC software is installed and properly configured, start the Kerberos daemon to provide Kerberos service for your realm. Refer to Section 39.5.3, "Starting the KDC" (page 628) for details.

7. **Create a Principal for Yourself** You need a principal for yourself. Refer to Section 39.5.2, “Creating a Principal” (page 628) for details.

39.5.1 Setting Up the Database

Your next step is to initialize the database where Kerberos keeps all information about principals. Set up the database master key, which is used to protect the database from accidental disclosure, in particular when it is backed up to a tape. The master key is derived from a pass phrase and is stored in a file called the stash file. This is so you do not need to enter the password every time the KDC is restarted. Make sure that you choose a good pass phrase, such as a sentence from a book opened to a random page.

When you make tape backups of the Kerberos database (`/var/lib/kerberos/krb5kdc/principal`), do not back up the stash file (which is in `/var/lib/kerberos/krb5kdc/.k5.EXAMPLE.COM`). Otherwise, everyone able to read the tape could also decrypt the database. Therefore, it is also a good idea to keep a copy of the pass phrase in a safe or some other secure location, because you need it to restore your database from backup tape after a crash.

To create the stash file and the database, run:

```
$> kdb5_util create -r EXAMPLE.COM -s
Initializing database '/var/lib/kerberos/krb5kdc/principal' for realm
'EXAMPLE.COM',
master key name 'K/M@EXAMPLE.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:  <= Type the master password.
Re-enter KDC database master key to verify:  <= Type it again.
$>
```

To verify that it did anything, use the list command:

```
$>kadmin.local
kadmin> listprincs
K/M@EXAMPLE.COM
kadmin/admin@EXAMPLE.COM
kadmin/changepw@EXAMPLE.COM
krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

This shows that there are now a number of principals in the database. All of these are for internal use by Kerberos.

39.5.2 Creating a Principal

Next, create two Kerberos principals for yourself: one normal principal for your everyday work and one for administrative tasks relating to Kerberos. Assuming your login name is `newbie`, proceed as follows:

```
kadmin.local

kadmin> ank newbie
newbie@EXAMPLE.COM's Password: <type password here>
Verifying password: <re-type password here>
```

Next, create another principal named `newbie/admin` by typing `ank newbie/admin` at the `kadmin` prompt. The `admin` suffixed to your username is a *role*. Later, use this role when administering the Kerberos database. A user can have several roles for different purposes. Roles are basically completely different accounts with similar names.

39.5.3 Starting the KDC

Start the KDC daemon and the kadmin daemon. To start the daemons manually, enter `rckrb5kdc start` and `rckadmind start`. Also make sure that KDC and kadmind are started by default when the server machine is rebooted with the command `insserv krb5kdc` and `insserv kadmind`.

39.6 Configuring Kerberos Clients

Once the supporting infrastructure is in place (DNS, NTP) and the KDC has been properly configured and started, configure the clients machines. You can either use YaST to configure a Kerberos client or use one of the two manual approaches described below.

39.6.1 Configuring a Kerberos Client with YaST

Rather than manually editing all relevant configuration files when configuring a Kerberos client, let YaST do the job for you. You can either perform the client configuration during the installation of your machine or in the installed system:

1 Log in as `root` and select *Network Services* > *Kerberos Client*.

2 Select *Use Kerberos*.

3 To configure a DNS-based Kerberos client, proceed as follows:

3a Confirm the *Basic Kerberos Settings* that are displayed.

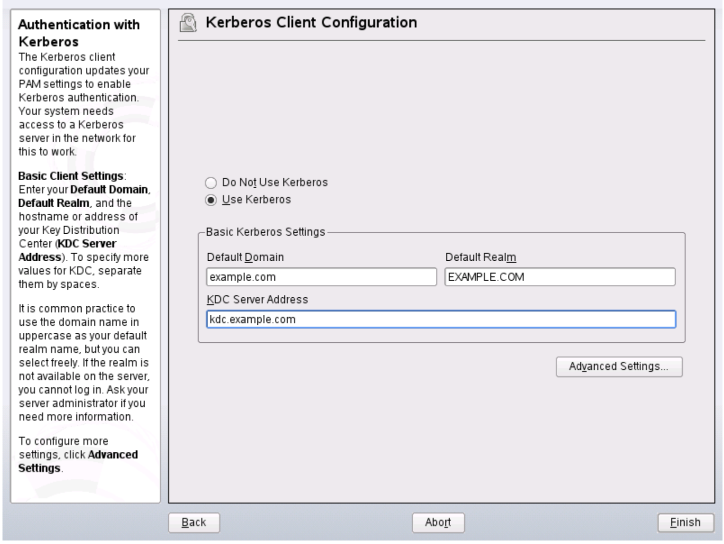
3b Click *Advanced Settings* to configure details on ticket-related issues, OpenSSH support, and time synchronization.

4 To configure a static Kerberos client, proceed as follows:

4a Set *Default Domain*, *Default Realm*, and *KDC Server Address* to the values that match your setup.

4b Click *Advanced Settings* to configure details on ticket-related issues, OpenSSH support, and time synchronization.

Figure 39.2 *YaST: Basic Configuration of a Kerberos Client*



Authentication with Kerberos
The Kerberos client configuration updates your PAM settings to enable Kerberos authentication. Your system needs access to a Kerberos server in the network for this to work.

Basic Client Settings: Enter your **Default Domain**, **Default Realm**, and the hostname or address of your Key Distribution Center (**KDC Server Address**). To specify more values for KDC, separate them by spaces.

It is common practice to use the domain name in uppercase as your default realm name, but you can select freely. If the realm is not available on the server, you cannot log in. Ask your server administrator if you need more information.

To configure more settings, click **Advanced Settings**.

Kerberos Client Configuration

○ Do Not Use Kerberos
● Use Kerberos

Basic Kerberos Settings

Default Domain: example.com

Default Realm: EXAMPLE.COM

KDC Server Address: kdc.example.com

To configure ticket-related options in the *Advanced Settings* dialog, choose from the following options:

- Specify the *Default Ticket Lifetime* and the *Default Renewable Lifetime* in days, hours, or minutes (using the units of measurement *d*, *h*, and *m*, with no blank space between the value and the unit).

- To forward your complete identity to use your tickets on other hosts, select *Forwardable*.

- Enable the transfer of certain tickets by selecting *Proxiable*.

- Keep tickets available with a PAM module even after a session has ended by enabling *Retained*.

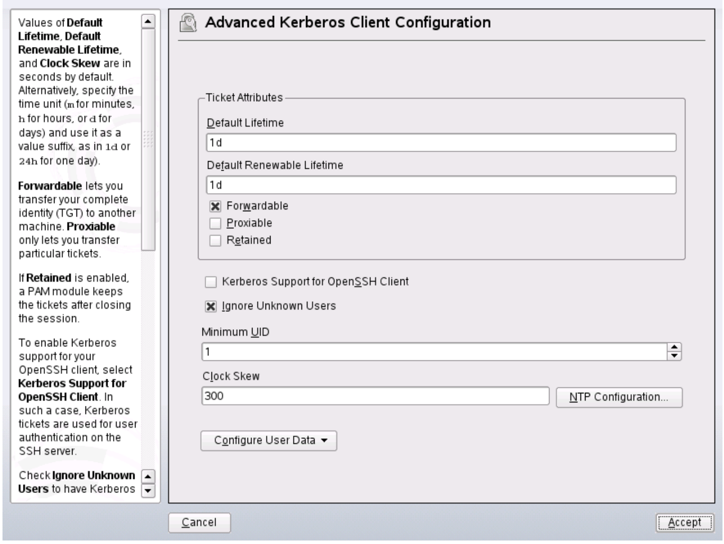
- Enable Kerberos authentication support for your OpenSSH client by selecting the corresponding check box. The client then uses Kerberos tickets to authenticate with the SSH server.

- Exclude a range of user accounts from using Kerberos authentication by providing a value for the *Minimum UID* that a user of this feature must have. For instance, you may want to exclude the system administrator (`root`).

- Use *Clock Skew* to set a value for the allowable difference between the time stamps and your host's system time.

- To keep the system time in sync with an NTP server, you can also set up the host as an NTP client by selecting *NTP Configuration*, which opens the YaST NTP client dialog that is described in Section 24.1, “Configuring an NTP Client with YaST” (page 393). After finishing the configuration, YaST performs all the necessary changes and the Kerberos client is ready for use.

Figure 39.3 *YaST: Advanced Configuration of a Kerberos Client*



Values of **Default Lifetime**, **Default Renewable Lifetime**, and **Clock Skew** are in seconds by default. Alternatively, specify the time unit (`m` for minutes, `h` for hours, or `d` for days) and use it as a value suffix, as in `1d` or `24h` for one day).

Forwardable lets you transfer your complete identity (TGT) to another machine. **Proxiable** only lets you transfer particular tickets.

If **Retained** is enabled, a PAM module keeps the tickets after closing the session.

To enable Kerberos support for your OpenSSH client, select **Kerberos Support for OpenSSH Client**. In such a case, Kerberos tickets are used for user authentication on the SSH server.

Check **Ignore Unknown Users** to have Kerberos

Advanced Kerberos Client Configuration

Ticket Attributes

Default Lifetime: 1d

Default Renewable Lifetime: 1d

☒ Forwardable
☐ Proxiable
☐ Retained

☐ Kerberos Support for OpenSSH Client

☒ Ignore Unknown Users

Minimum UID: 1

Clock Skew: 300

39.6.2 Manually Configuring Kerberos Clients

When configuring Kerberos, there are basically two approaches you can take—static configuration in the `/etc/krb5.conf` file or dynamic configuration with DNS. With DNS configuration, Kerberos applications try to locate the KDC services using DNS records. With static configuration, add the hostnames of your KDC server to `krb5.conf` (and update the file whenever you move the KDC or reconfigure your realm in other ways).

DNS-based configuration is generally a lot more flexible and the amount of configuration work per machine is a lot less. However, it requires that your realm name is either the same as your DNS domain or a subdomain of it. Configuring Kerberos via DNS also creates a minor security issue—an attacker can seriously disrupt your infrastructure through your DNS (by shooting down the name server, spoofing DNS records, etc.). However, this amounts to a denial of service at most. A similar scenario applies to the static configuration case unless you enter IP addresses in `krb5.conf` instead of hostnames.

Static Configuration

One way to configure Kerberos is to edit the configuration file /etc/krb5.conf. The file installed by default contains various sample entries. Erase all of these entries before starting. krb5.conf is made up of several sections, each introduced by the section name included in brackets like [this].

To configure your Kerberos clients, add the following stanza to krb5.conf (where kdc.example.com is the hostname of the KDC):

```
[libdefaults]
        default_realm = EXAMPLE.COM

[realms]
        EXAMPLE.COM = {
                kdc = kdc.example.com
                admin_server = kdc.example.com
        }
```

The default_realm line sets the default realm for Kerberos applications. If you have several realms, just add additional statements to the [realms] section.

Also add a statement to this file that tells applications how to map hostnames to a realm. For example, when connecting to a remote host, the Kerberos library needs to know in which realm this host is located. This must be configured in the [domain_realms] section:

```
[domain_realm]
        .example.com = EXAMPLE.COM
        www.foobar.com = EXAMPLE.COM
```

This tells the library that all hosts in the example.com DNS domains are in the EXAMPLE.COM Kerberos realm. In addition, one external host named www.foobar.com should also be considered a member of the EXAMPLE.COM realm.

DNS-Based Configuration

DNS-based Kerberos configuration makes heavy use of SRV records. See *(RFC2052) A DNS RR for specifying the location of services* at http://www.ietf.org. These records are not supported in earlier implementations of the BIND name server. At least BIND version 8 is required for this.

The name of an SRV record, as far as Kerberos is concerned, is always in the format _service._proto.realm, where realm is the Kerberos realm. Domain names in DNS are case insensitive, so case-sensitive Kerberos realms would break when using this configuration method. _service is a service name (different names are used when trying to contact the KDC or the password service, for example). _proto can be either _udp or _tcp, but not all services support both protocols.

The data portion of SRV resource records consists of a priority value, a weight, a port number, and a hostname. The priority defines the order in which hosts should be tried (lower values indicate a higher priority). The weight is there to support some sort of load balancing among servers of equal priority. You probably do not need any of this, so it is okay to set these to zero.

MIT Kerberos currently looks up the following names when looking for services:

_kerberos

This defines the location of the KDC daemon (the authentication and ticket granting server). Typical records look like this:

```
_kerberos._udp.EXAMPLE.COM.  IN  SRV    0 0 88 kdc.example.com.
_kerberos._tcp.EXAMPLE.COM.  IN  SRV    0 0 88 kdc.example.com.
```

_kerberos-adm

This describes the location of the remote administration service. Typical records look like this:

```
_kerberos-adm._tcp.EXAMPLE.COM. IN  SRV    0 0 749 kdc.example.com.
```

Because kadmind does not support UDP, there should be no _udp record.

As with the static configuration file, there is a mechanism to inform clients that a specific host is in the EXAMPLE.COM realm, even if it is not part of the example.com DNS domain. This can be done by attaching a TXT record to _keberos.hostname, as shown here:

```
_keberos.www.foobar.com.  IN TXT "EXAMPLE.COM"
```

Adjusting the Clock Skew

The *clock skew* is the tolerance for accepting tickets with time stamps that do not exactly match the host's system clock. Usually, the clock skew is set to 300 seconds (five minutes). This means a ticket can have a time stamp somewhere between five minutes ago and five minutes in the future from the server's point of view.

When using NTP to synchronize all hosts, you can reduce this value to about one minute. The clock skew value can be set in `/etc/krb5.conf` like this:

```
[libdefaults]
        clockskew = 120
```

39.7 Configuring Remote Kerberos Administration

To be able to add and remove principals from the Kerberos database without accessing the KDC's console directly, tell the Kerberos administration server which principals are allowed to do what. Do this by editing the file `/var/lib/kerberos/krb5kdc/kadm5.acl`. The ACL (access control list) file allows you to specify privileges with a fine degree of control. For details, refer to the manual page with `man 8 kadmind`.

Right now, just grant yourself the privilege to do anything you want with the database by putting the following line into the file:

```
newbie/admin              *
```

Replace the username `newbie` with your own. Restart kadmind for the change to take effect.

You should now be able to perform Kerberos administration tasks remotely using the kadmin tool. First, obtain a ticket for your admin role and use that ticket when connecting to the kadmin server:

```
kadmin -p newbie/admin
Authenticating as principal newbie/admin@EXAMPLE.COM with password.
Password for newbie/admin@EXAMPLE.COM:
kadmin:  getprivs
current privileges: GET ADD MODIFY DELETE
kadmin:
```

Using the `getprivs` command, verify which privileges you have. The list shown above is the full set of privileges.

As an example, modify the principal `newbie`:

```
kadmin -p newbie/admin
Authenticating as principal newbie/admin@EXAMPLE.COM with password.
Password for newbie/admin@EXAMPLE.COM:

kadmin:  getprinc newbie
Principal: newbie@EXAMPLE.COM
Expiration date: [never]
Last password change: Wed Jan 12 17:28:46 CET 2005
Password expiration date: [none]
Maximum ticket life: 0 days 10:00:00
Maximum renewable life: 7 days 00:00:00
Last modified: Wed Jan 12 17:47:17 CET 2005 (admin/admin@EXAMPLE.COM)
Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 2
Key: vno 1, Triple DES cbc mode with HMAC/sha1, no salt
Key: vno 1, DES cbc mode with CRC-32, no salt
Attributes:
Policy: [none]

kadmin:  modify_principal -maxlife "8 hours" newbie
Principal "newbie@EXAMPLE.COM" modified.
kadmin:  getprinc joe
Principal: newbie@EXAMPLE.COM
Expiration date: [never]
Last password change: Wed Jan 12 17:28:46 CET 2005
Password expiration date: [none]
Maximum ticket life: 0 days 08:00:00
Maximum renewable life: 7 days 00:00:00
Last modified: Wed Jan 12 17:59:49 CET 2005 (newbie/admin@EXAMPLE.COM)
Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 2
Key: vno 1, Triple DES cbc mode with HMAC/sha1, no salt
Key: vno 1, DES cbc mode with CRC-32, no salt
Attributes:
Policy: [none]
kadmin:
```

This changes the maximum ticket life time to eight hours. For more information about the `kadmin` command and the options available, refer to `http://web.mit.edu/kerberos/www/krb5-1.4/krb5-1.4/doc/krb5-admin.html#Kadmin%20Options` or look at `man 8 kadmin`.

39.8 Creating Kerberos Service Principals

So far, only user credentials have been discussed. However, Kerberos-compatible services usually need to authenticate themselves to the client user, too. Therefore, special service principals must be present in the Kerberos database for each service offered in the realm. For example, if ldap.example.com offers an LDAP service, you need a service principal, `ldap/ldap.example.com@EXAMPLE.COM`, to authenticate this service to all clients.

The naming convention for service principals is *`service/hostname@REALM`*, where `hostname` is the host's fully qualified hostname.

Valid service descriptors are:

Service Descriptor	Service
`host`	Telnet, RSH, SSH
`nfs`	NFSv4 (with Kerberos support)
`HTTP`	HTTP (with Kerberos authentication)
`imap`	IMAP
`pop`	POP3
`ldap`	LDAP

Service principals are similar to user principals, but have significant differences. The main difference between a user principal and a service principal is that the key of the former is protected by a password—when a user obtains a ticket-granting ticket from the KDC, he needs to type his password so Kerberos can decrypt the ticket. It would be quite inconvenient for the system administrator if he had to obtain new tickets for the SSH daemon every eight hours or so.

Instead, the key required to decrypt the initial ticket for the service principal is extracted by the administrator from the KDC once and stored in a local file called the *keytab*. Services such the SSH daemon read this key and use it to obtain new tickets automatically when needed. The default keytab file resides in /etc/krb5.keytab.

To create a host service principal for jupiter.example.com enter the following commands during your kadmin session:

```
kadmin -p newbie/admin
Authenticating as principal newbie/admin@EXAMPLE.COM with password.
Password for newbie/admin@EXAMPLE.COM:
kadmin:  addprinc -randkey host/jupiter.example.com
WARNING: no policy specified for host/jupiter.example.com@EXAMPLE.COM;
defaulting to no policy
Principal "host/jupiter.example.com@EXAMPLE.COM" created.
```

Instead of setting a password for the new principal, the -randkey flag tells kadmin to generate a random key. This is used here because no user interaction is wanted for this principal. It is a server account for the machine.

Finally, extract the key and store it in the local keytab file /etc/krb5.keytab. This file is owned by the superuser, so you must be root to execute the next command in the kadmin shell:

```
kadmin:  ktadd host/jupiter.example.com
Entry for principal host/jupiter.example.com with kvno 3, encryption type
Triple
DES cbc mode with HMAC/sha1 added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/jupiter.example.com with kvno 3, encryption type DES
cbc mode with CRC-32 added to keytab WRFILE:/etc/krb5.keytab.
kadmin:
```

When completed, make sure that you destroy the admin ticket obtained with kinit above with kdestroy.

39.9 Enabling PAM Support for Kerberos

openSUSE® comes with a PAM module named pam_krb5, which supports Kerberos login and password update. This module can be used by applications, such as console login, su, and graphical login applications like KDM, where the user presents a password and would like the authenticating application to obtain an initial Kerberos ticket on his behalf. To configure PAM support for Kerberos, use the following command:

```
pam-config --add --krb5
```

The above command adds the pam_krb5 module to the existing PAM configuration files and makes sure it is called in the right order. To make fine adjustments to the way in which pam_krb5 is used, edit the file /etc/krb5.conf and add default applications to pam. For details, refer to the manual page with man 5 pam_krb5.

The pam_krb5 module was specifically not designed for network services that accept Kerberos tickets as part of user authentication. This is an entirely different matter, which is discussed below.

39.10 Configuring SSH for Kerberos Authentication

OpenSSH supports Kerberos authentication in both protocol version 1 and 2. In version 1, there are special protocol messages to transmit Kerberos tickets. Version 2 does not use Kerberos directly anymore, but relies on GSSAPI, the General Security Services API. This is a programming interface that is not specific to Kerberos—it was designed to hide the peculiarities of the underlying authentication system, be it Kerberos, a public-key authentication system like SPKM, or others. The GSSAPI library included supports only Kerberos, however.

To use sshd with Kerberos authentication, edit /etc/ssh/sshd_config and set the following options:

```
# These are for protocol version 1
#
# KerberosAuthentication yes
# KerberosTicketCleanup yes

# These are for version 2 - better to use this
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
```

Then restart your SSH daemon using rcsshd restart.

To use Kerberos authentication with protocol version 2, enable it on the client side as well. Do this either in the systemwide configuration file /etc/ssh/ssh_config or on a per-user level by editing ~/.ssh/config. In both cases, add the option GSSAPIAuthentication yes.

You should now be able to connect using Kerberos authentication. Use klist to verify that you have a valid ticket then connect to the SSH server. To force SSH protocol version 1, specify the -1 option on the command line.

TIP: Additional Information

The file /usr/share/doc/packages/openssh/README.kerberos discusses the interaction of OpenSSH and Kerberos in more detail.

39.11 Using LDAP and Kerberos

When using Kerberos, one way to distribute the user information (such as user ID, groups,and home directory) in your local network is to use LDAP. This requires a strong authentication mechanism that prevents packet spoofing and other attacks. One solution is to use Kerberos for LDAP communication, too.

OpenLDAP implements most authentication flavors through SASL, the simple authentication session layer. SASL is basically a network protocol designed for authentication. The SASL implementation is cyrus-sasl, which supports a number of different authentication flavors. Kerberos authentication is performed through GSSAPI (General Security Services API). By default, the SASL plug-in for GSSAPI is not installed. Install it manually with rpm -ivh cyrus-sasl-gssapi-*.rpm.

To enable Kerberos to bind to the OpenLDAP server, create a principal ldap/earth.example.com and add that to the keytab.

By default, the LDAP server slapd runs as user and group ldap, while the keytab file is readable by root only. Therefore, either change the LDAP configuration so the server runs as root or make the keytab file readable by the group ldap. The latter is done automatically by the OpenLDAP start script (/etc/init.d/ldap) if the keytab file has been specified in the OPENLDAP_KRB5_KEYTAB variable in /etc/sysconfig/openldap and the OPENLDAP_CHOWN_DIRS variable is set to yes, which is the default setting. If OPENLDAP_KRB5_KEYTAB is left empty, the default keytab under /etc/krb5.keytab is used and you must adjust the privileges yourself as described below.

To run slapd as `root`, edit `/etc/sysconfig/openldap`. Disable the `OPENLDAP_USER` and `OPENLDAP_GROUP` variables by putting a comment character in front of them.

To make the keytab file readable by group LDAP, execute

```
chgrp ldap /etc/krb5.keytab
```

```
chmod 640 /etc/krb5.keytab
```

A third, and maybe the best solution, is to tell OpenLDAP to use a special keytab file. To do this, start kadmin, and enter the following command after you have added the principal ldap/earth.example.com:

```
ktadd -k /etc/openldap/ldap.keytab ldap/earth.example.com@EXAMPLE.COM
```

Then, on the shell, run:

```
chown ldap.ldap /etc/openldap/ldap.keytab
chmod 600 /etc/openldap/ldap.keytab
```

To tell OpenLDAP to use a different keytab file, change the following variable in `/etc/sysconfig/openldap`:

```
OPENLDAP_KRB5_KEYTAB="/etc/openldap/ldap.keytab"
```

Finally, restart the LDAP server using `rcldap restart`.

39.11.1 Using Kerberos Authentication with LDAP

You should now be able to use tools, such as ldapsearch, with Kerberos authentication automatically.

```
ldapsearch -b ou=people,dc=example,dc=com '(uid=newbie)'

SASL/GSSAPI authentication started
SASL SSF: 56
SASL installing layers
[...]

# newbie, people, example.com
dn: uid=newbie,ou=people,dc=example,dc=com
uid: newbie
cn: Olaf Kirch
[...]
```

As you can see, ldapsearch prints a message that it started GSSAPI authentication. The next message is very cryptic, but it shows that the *security strength factor* (SSF for short) is 56 (The value 56 is somewhat arbitrary. Most likely it was chosen because this is the number of bits in a DES encryption key). What this tells you is that GSSAPI authentication was successful and that encryption is being used to provide integrity protection and confidentiality for the LDAP connection.

In Kerberos, authentication is always mutual. This means that not only have you authenticated yourself to the LDAP server, but also the LDAP server authenticated itself to you. In particular, this means communication is with the desired LDAP server, rather than some bogus service set up by an attacker.

39.11.2 Kerberos Authentication and LDAP Access Control

Now, allow each user to modify the login shell attribute of their LDAP user record. Assuming you have a schema where the LDAP entry of user `joe` is located at `uid=joe,ou=people,dc=example,dc=com`, set up the following access controls in `/etc/openldap/slapd.conf`:

```
# This is required for things to work _at all_
access to dn.base="" by * read
# Let each user change their login shell
access to dn="*,ou=people,dc=example,dc=com" attrs=loginShell
       by self write
# Every user can read everything
access to *
       by users read
```

The second statement gives authenticated users write access to the `loginShell` attribute of their own LDAP entry. The third statement gives all authenticated users read access to the entire LDAP directory.

There is one minor piece of the puzzle missing—how the LDAP server can find out that the Kerberos user `joe@EXAMPLE.COM` corresponds to the LDAP distinguished name `uid=joe,ou=people,dc=example,dc=com`. This sort of mapping must be configured manually using the `saslExpr` directive. In this example, add the following to `slapd.conf`:

```
authz-regexp
   uid=(.*),cn=GSSAPI,cn=auth
   uid=$1,ou=people,dc=example,dc=com
```

To understand how this works, you need to know that when SASL authenticates a user, OpenLDAP forms a distinguished name from the name given to it by SASL (such as joe) and the name of the SASL flavor (GSSAPI). The result would be uid=joe,cn=GSSAPI,cn=auth.

If a authz-regexp has been configured, it checks the DN formed from the SASL information using the first argument as a regular expression. If this regular expression matches, the name is replaced with the second argument of the authz-regexp statement. The placeholder $1 is replaced with the substring matched by the (.*) expression.

More complicated match expressions are possible. If you have a more complicated directory structure or a schema in which the username is not part of the DN, you can even use search expressions to map the SASL DN to the user DN.

40 Encrypting Partitions and Files

Every user has some confidential data that third parties should not be able to access. The more you rely on mobile computing and on working in different environments and networks, the more carefully you should handle your data. The encryption of files or entire partitions is recommended if others have network or physical access to your system. Laptops or removable media, such as external hard disks or USB sticks, are prone to being lost or stolen. Thus, it is recommended to encrypt the parts of your file that hold confidential data.

There are several ways to protect your data by means of encryption:

Encrypting a Hard Disk Partition

You can create an encrypted partition with YaST during installation or in an already installed system. Refer to Section 40.1.1, “Creating an Encrypted Partition during Installation” (page 645) and Section 40.1.2, “Creating an Encrypted Partition on a Running System” (page 646) for details. This option can also be used for removable media, such as external hard disks, as described in Section 40.1.4, “Encrypting the Content of Removable Media” (page 647).

Creating an Encrypted File as Container

You can create an encrypted file on your hard disk or on a removable medium with YaST at any time. The encrypted file can then be used to *store* other files or folders. For more information, refer to Section 40.1.3, “Creating an Encrypted File as a Container” (page 646).

Encrypting Home Directories

With openSUSE, you can also create encrypted home directories for users. When the user logs in to the system, the encrypted home directory is mounted and the contents are made available to the user. Refer to Section 40.2, "Using Encrypted Home Directories" (page 647) for more information.

Encrypting Single Files

If you only have a small number of files that hold sensitive or confidential data, you can encrypt them individually and protect them with a password using the vi editor. Refer to Section 40.3, "Using vi to Encrypt Single Files" (page 649) for more information.

WARNING: Encrypted Media Offers Limited Protection

The methods described in this chapter offer only limited protection. You cannot protect your running system from being compromised. After the encrypted medium is successfully mounted, everybody with appropriate permissions has access to it. However, encrypted media are useful in case of loss or theft of your computer or to prevent unauthorized individuals from reading your confidential data.

40.1 Setting Up an Encrypted File System with YaST

Use YaST to encrypt partitions or parts of your file system during installation or in an already installed system. However, encrypting a partition in an already installed system is more difficult, because you have to resize and change existing partitions. In such cases, it may be more convenient to create an encrypted file of a defined size in which to *store* other files or parts of your file system. To encrypt an entire partition, dedicate a partition for encryption in the partition layout. The standard partitioning proposal as suggested by YaST does not, by default, include an encrypted partition. Add it manually in the partitioning dialog.

40.1.1 Creating an Encrypted Partition during Installation

WARNING: Password Input

Make sure to memorize the password for your encrypted partitions well. Without that password you cannot access or restore the encrypted data.

The YaST expert dialog for partitioning offers the options needed for creating an encrypted partition. To create a new encrypted partition proceed as follows:

1 Run the YaST Partitioner from the YaST Control Center with *System* > *Partitioner*

2 Click *Create* and select a primary or a logical partition.

3 Select the desired file system, size and mount point of this partition.

4 If the encrypted file system should only be mounted when necessary, enable *Do Not Mount at System Start-up* in the *Fstab Options*.

5 Activate the *Encrypt file system* checkbox.

6 Click *OK*. You will be prompted for a password that is used to encrypt this partition. This password is not displayed. To prevent typing errors, enter the password twice.

7 Complete the process by clicking *OK*. The new encrypted partition is now created.

Unless *Do Not Mount at System Start-up* was selected, the operating system requests the password while booting before mounting the partition. The partition is available to all users once it has been mounted.

To skip mounting the encrypted partition during start-up, click Enter when prompted for the password. Then decline the offer to enter the password again. In this case, the encrypted file system is not mounted and the operating system continues booting, blocking access to your data.

To access an encrypted partition that is not mounted during boot, mount the partition manually by entering `mount  name_of_partition mount_point`. Enter the password when prompted for it. After you are done with working on this partition, unmount it with `umount name_of_partition` to protect it from access by other users.

When you are installing your system on a machine where several partitions already exist, you can also decide to encrypt an existing partition during installation. In this case follow the description in Section 40.1.2, "Creating an Encrypted Partition on a Running System" (page 646) and be aware that this action destroys all data on the existing partition to encrypt.

40.1.2 Creating an Encrypted Partition on a Running System

WARNING: Activating Encryption in a Running System

It is also possible to create encrypted partitions on a running system. However, encrypting an existing partition destroys all data on it and requires resizing and restructuring of existing partitions.

On a running system, select *System* > *Partitioning* in the YaST Control Center. Click *Yes* to proceed. In the *Expert Partitioner*, select the partition to encrypt and click *Edit*. The rest of the procedure is the same as described in Section 40.1.1, "Creating an Encrypted Partition during Installation" (page 645).

40.1.3 Creating an Encrypted File as a Container

Instead of using a partition, it is possible to create an encrypted file of a certain size that can then hold other files or folders containing confidential data. Such container files are created from the YaST Expert Partitioner dialog. Select *Crypt File* and enter the full path to the file and its size. Accept the proposed formatting settings and the file system type. Specify the mount point and decide whether the encrypted file system should be mounted at system boot.

The advantage of encrypted container files over encrypted partitions is that they can be added without repartitioning the hard disk. They are mounted with the help of a loop device and behave just like normal partitions.

40.1.4 Encrypting the Content of Removable Media

YaST treats removable media like external hard disks or USB flash drives the same as any other hard disk. Container files or partitions on such media can be encrypted as described above. However, enable *Do Not Mount During Booting* in the *Fstab Options* dialog, because removable media are usually only connected while the system is running.

If you have encrypted your removable device with LUKS (Linux Unified Key Setup), the KDE and GNOME desktops automatically recognize this and prompt for the password when the device is detected. If you have formatted your removable medium with a FAT file system, the user logged in to the desktop and who enters the decryption password automatically becomes the owner of the device and can read and write files. For devices with a file system other than FAT, change the ownership explicitly for users other than `root` to enable these users to read or write files on the device.

40.2 Using Encrypted Home Directories

To protect data in home directories against theft and hard disk removal, use the LUKS framework to encrypt user home directories. LUKS creates an image and an image key. The image key is protected with the user's login password. By default, the image and the image key are located in the respective user's home directory. The key can also be located anywhere in the file system—for example, on a removable device that can be mounted manually.

Use the YaST user management module or the `cryptconfig` command line tool to enable encryption of home directories. You can create encrypted home directories for new or existing users. To encrypt or modify encrypted home directories of already existing users, enter the user's current login password. For information about user management with YaST, refer to Chapter 5, *Managing Users with YaST* (↑Start-Up).

WARNING: Security Restrictions

Encrypting a user's home directory does not provide strong security from other users. If strong security is required, the system should not be shared physically.

To enhance security, also encrypt the swap partition and the /tmp and /var/tmp directories, because these may contain temporary images of critical data.

You can encrypt swap, /tmp, and /var/tmp with the YaST partitioner as described in Section 40.1.1, "Creating an Encrypted Partition during Installation" (page 645) and Section 40.1.3, "Creating an Encrypted File as a Container" (page 646). In addition to the options YaST, you can use the cryptconfig command line tool for some special tasks.

For example, you can create and add an additional key to the image.

1. Log in to a shell as root.

2. Run

   ```
   cryptconfig create-key admin.key
   ```

   to create a key for administrators.

3. To create an encrypted home directory for user tux and to add the administration key to it, enter

   ```
   cryptconfig make-ehd --extra-key-file=admin.key tux 200
   ```

   This creates a home directory with the initial size of 200 MB.

4. To change the size of the home directory at any time, use

   ```
   cryptconfig enlarge-size image
        size_to_add_in_MB
   ```

For more information about the command line tool, run the cryptconfig --help command.

Internally, the home directory is provided by means of the pam module pam_mount. If you need to add an additional login method that provides encrypted home directories, you may have to add this module to the respective configuration file in /etc/pam.d/

. For more information see also Chapter 18, *Authentication with PAM* (page 263) and the man page of `pam_mount`.

40.3 Using vi to Encrypt Single Files

The disadvantage of using encrypted partitions is that while the partition is mounted, at least `root` can access the data. To prevent this, vi can be used in encrypted mode.

Use `vi -x` *`filename`* to edit a new file. vi prompts you to set a password, after which it encrypts the content of the file. Whenever you access this file, vi requests the correct password.

For even more security, you can place the encrypted text file in an encrypted partition. This is recommended because the encryption used in vi is not very strong.

Confining Privileges with AppArmor

41

Many security vulnerabilities result from bugs in *trusted* programs. A trusted program runs with privilege that some attacker would like to have. The program fails to keep that trust if there is a bug in the program that allows the attacker to acquire that privilege.

Novell® AppArmor is an application security solution designed specifically to provide least privilege confinement to suspect programs. AppArmor allows the administrator to specify the domain of activities the program can perform by developing a security *profile* for that application—a listing of files that the program may access and the operations the program may perform.

Effective hardening of a computer system requires minimizing the number of programs that mediate privilege then securing the programs as much as possible. With Novell AppArmor, you only need to profile the programs that are exposed to attack in your environment, which drastically reduces the amount of work required to harden your computer. AppArmor profiles enforce policies to make sure that programs do what they are supposed to do, but nothing else.

Administrators only need to care about the applications that are vulnerable to attacks and generate profiles for these. Hardening a system thus comes down to building and maintaining the AppArmor profile set and monitoring any policy violations or exceptions logged by AppArmor's reporting facility.

Building AppArmor profiles to confine an application is very straightforward and intuitive. AppArmor ships with several tools that assist in profile creation. It does not require you to do any programming or script handling. The only task that is required from the administrator is to determine a policy of strictest access and execute permissions for each application that needs to be hardened.

Updates or modifications to the application profiles are only required if the software configuration or the desired range of activities changes. AppArmor offers intuitive tools to handle profile updates or modifications.

Users should not notice AppArmor at all. It runs “behind the scenes” and does not require any user interaction. Performance is not affected noticeably by AppArmor. If some activity of the application is not covered by an AppArmor profile or if some activity of the application is prevented by AppArmor, the administrator needs to adjust the profile of this application to cover this kind of behavior.

This guide outlines the basic tasks that need to be performed with AppArmor to effectively harden a system. For more in-depth information, refer to *Novell AppArmor Administration Guide*.

41.1 Installing Novell AppArmor

Novell AppArmor is installed and running by default on any installation of openSUSE® regardless of what patterns are installed. The packages listed below are needed for a fully functional instance of AppArmor

- `apparmor-parser`
- `libapparmor`
- `apparmor-docs`
- `yast2-apparmor`
- `apparmor-profiles`
- `apparmor-utils`
- `audit`

41.2 Enabling and Disabling Novell AppArmor

Novell AppArmor is configured to run by default on any fresh installation of openSUSE. There are two ways of toggling the status of AppArmor:

Using YaST System Services (Runlevel)
: Disable or enable AppArmor by removing or adding its boot script to the sequence of scripts executed on system boot. Status changes are applied at the next system boot.

Using Novell AppArmor Control Panel
: Toggle the status of Novell AppArmor in a running system by switching it off or on using the YaST Novell AppArmor Control Panel. Changes made here are applied instantaneously. The Control Panel triggers a stop or start event for AppArmor and removes or adds its boot script in the system's boot sequence.

To disable AppArmor permanently by removing it from the sequence of scripts executed on system boot, proceed as follows:

1. Log in as `root` and start YaST.

2. Select *System* > *System Services (Runlevel)*.

3. Select *Expert Mode*.

4. Select `boot.apparmor` and click *Set/Reset* > *Disable the service*.

5. Exit the YaST Runlevel tool with *Finish*.

AppArmor will not be initialized on the next system boot and stays inactive until you explicitly reenable it. Reenabling a service using the YaST Runlevel tool is similar to disabling it.

Toggle the status of AppArmor in a running system by using the AppArmor Control Panel. These changes take effect as soon as you apply them and survive a reboot of the system. To toggle AppArmor's status, proceed as follows:

1. Log in as `root` and start YaST.

2. Select *Novell AppArmor* > *AppArmor Control Panel*.

3. Select *Enable AppArmor*. To disable AppArmor, uncheck this option.

4. Exit the AppArmor Control Panel with *Done*.

41.3 Getting Started with Profiling Applications

Prepare a successful deployment of Novell AppArmor on your system by carefully considering the following items:

1. Determine the applications to profile. Read more on this in Section 41.3.1, “Choosing the Applications to Profile” (page 654).

2. Build the needed profiles as roughly outlined in Section 41.3.2, “Building and Modifying Profiles” (page 655). Check the results and adjust the profiles when necessary.

3. Keep track of what is happening on your system by running AppArmor reports and dealing with security events. Refer to Section 41.3.3, “Configuring Novell AppArmor Event Notification and Reports” (page 658).

4. Update your profiles whenever your environment changes or you need to react to security events logged by AppArmor's reporting tool. Refer to Section 41.3.4, “Updating Your Profiles” (page 659).

41.3.1 Choosing the Applications to Profile

You only need to protect the programs that are exposed to attacks in your particular setup, so only use profiles for those applications you really run. Use the following list to determine the most likely candidates:

Network Agents

Programs (servers and clients) that have open network ports. User clients, such as mail clients and Web browsers, mediate privilege. These programs run with the privilege to write to the user's home directory and they process input from potentially hostile remote sources, such as hostile Web sites and e-mailed malicious code.

Web Applications

Programs that can be invoked through a Web browser, including CGI Perl scripts, PHP pages, and more complex Web applications.

Cron Jobs
: Programs that the cron daemon periodically run read input from a variety of sources.

To find out which processes are currently running with open network ports and might need a profile to confine them, run `aa-unconfined` as `root`.

Example 41.1 *Output of aa-unconfined*

```
19848 /usr/sbin/cupsd not confined
19887 /usr/sbin/sshd not confined
19947 /usr/lib/postfix/master not confined
29205 /usr/sbin/sshd confined by '/usr/sbin/sshd (enforce)'
```

Each of the processes in the above example labeled `not confined` might need a custom profile to confine it. Those labeled `confined by` are already protected by AppArmor.

TIP: For More Information

For more information about choosing the the right applications to profile, refer to Section “Determining Programs to Immunize” (Chapter 1, *Immunizing Programs*, ↑Novell AppArmor Administration Guide).

41.3.2 Building and Modifying Profiles

Novell AppArmor on openSUSE ships with a preconfigured set of profiles for the most important applications. In addition to that, you can use AppArmor to create your own profiles for any application you want.

There are two ways of managing profiles. One is to use the graphical front-end provided by the YaST Novell AppArmor modules and the other is to use the command line tools provided by the AppArmor suite itself. Both methods basically work the same way.

Running aa-unconfined as described in Section 41.3.1, “Choosing the Applications to Profile” (page 654) identifies a list of applications that may need a profile to run in a safe mode.

For each application, perform the following steps to create a profile:

1 As `root`, let AppArmor create a rough outline of the application's profile by running `aa-genprof` *`programname`*

or

Outline the basic profile by running *YaST* > *Novell AppArmor* > *Add Profile Wizard* and specifying the complete path of the application to profile.

A basic profile is outlined and AppArmor is put into learning mode, which means that it logs any activity of the program you are executing but does not yet restrict it.

2. Run the full range of the application's actions to let AppArmor get a very specific picture of its activities.

3. Let AppArmor analyze the log files generated in Step 2 (page 656) by running typing `S` in aa-genprof.

   or

   Analyze the logs by clicking *Scan system log for AppArmor events* in the *Add Profile Wizard* and following the instructions given in the wizard until the profile is completed.

   AppArmor scans the logs it recorded during the application's run and asks you to set the access rights for each event that was logged. Either set them for each file or use globbing.

4. Depending on the complexity of your application, it might be necessary to repeat Step 2 (page 656) and Step 3 (page 656). Confine the application, exercise it under the confined conditions, and process any new log events. To properly confine the full range of an application's capabilities, you might be required to repeat this procedure often.

5. Once all access permissions are set, your profile is set to enforce mode. The profile is applied and AppArmor restricts the application according to the profile just created.

   If you started aa-genprof on an application that had an existing profile that was in complain mode, this profile remains in learning mode upon exit of this learning cycle. For more information about changing the mode of a profile, refer to Section “aa-complain—Entering Complain or Learning Mode” (Chapter 4, *Building Profiles from the Command Line*, ↑Novell AppArmor Administration Guide) and Section “aa-enforce—Entering Enforce Mode” (Chapter 4, *Building Profiles from the Command Line*, ↑Novell AppArmor Administration Guide).

Test your profile settings by performing every task you need with the application you just confined. Normally, the confined program runs smoothly and you do not notice AppArmor activities at all. However, if you notice certain misbehavior with your application, check the system logs and see if AppArmor is too tightly confining your application. Depending on the log mechanism used on your system, there are several places to look for AppArmor log entries:

`/var/log/audit/audit.log`
: If the `audit` package is installed and auditd is running, AppArmor events are logged as follows:

```
type=APPARMOR msg=audit(1140325305.502:1407): REJECTING w access to
/usr/lib/firefox/update.test (firefox-bin(9469) profile
/usr/lib/firefox/firefox-bin active /usr/lib/firefox/firefox-bin)
```

`/var/log/messages`
: If auditd is not used, AppArmor events are logged in the standard system log under `/var/log/messages`. An example entry would look like the following:

```
Feb 22 18:29:14 dhcp-81 klogd: audit(1140661749.146:3): REJECTING w access
to /dev/console (mdnsd(3239) profile /usr/sbin/mdnsd active
/usr/sbin/mdnsd)
```

`dmesg`
: If auditd is not running, AppArmor events can also be checked using the `dmesg` command:

```
audit(1140661749.146:3): REJECTING w access to /dev/console (mdnsd(3239)
profile /usr/sbin/mdnsd active /usr/sbin/mdnsd)
```

To adjust the profile, analyze the log messages relating to this application again as described in Step 3 (page 656). Determine the access rights or restrictions when prompted.

TIP: For More Information

For more information about profile building and modification, refer to Chapter 2, *Profile Components and Syntax* (↑Novell AppArmor Administration Guide), Chapter 3, *Building and Managing Profiles with YaST* (↑Novell AppArmor Administration Guide), and Chapter 4, *Building Profiles from the Command Line* (↑Novell AppArmor Administration Guide).

41.3.3 Configuring Novell AppArmor Event Notification and Reports

Set up event notification in Novell AppArmor so you can review security events. Event Notification is an Novell AppArmor feature that informs a specified e-mail recipient when systemic Novell AppArmor activity occurs under the chosen severity level. This feature is currently available in the YaST interface.

To set up event notification in YaST, proceed as follows:

1. Make sure that a mail server is running on your system to deliver the event notifications.

2. Log in as `root` and start YaST. Then select *Novell AppArmor* > *AppArmor Control Panel*).

3. In *Enable Security Event Notification*, select *Configure*.

4. For each record type (*Terse*, *Summary*, and *Verbose*), set a report frequency, enter the e-mail address that should receive the reports, and determine the severity of events to log. To include unknown events in the event reports, check *Include Unknown Severity Events*.

   NOTE: Selecting Events to Log

   Unless you are familiar with AppArmor's event categorization, choose to be notified about events for all security levels.

5. Leave this dialog with *OK* > *Done* to apply your settings.

Using Novell AppArmor reports, you can read important Novell AppArmor security events reported in the log files without manually sifting through the cumbersome messages only useful to the aa-logprof tool. You can decrease the size of the report by filtering by date range or program name.

To configure the AppArmor reports, proceed as follows:

1. Log in as `root` and start YaST. Select *Novell AppArmor* > *AppArmor Reports*.

2. Select the type of report to examine or configure from *Executive Security Summary*, *Applications Audit*, and *Security Incident Report*.

3. Edit the report generation frequency, e-mail address, export format, and location of the reports by selecting *Edit* and providing the requested data.

4. To run a report of the selected type, click *Run Now*.

5. Browse through the archived reports of a given type by selecting *View Archive* and specifying the report type.

   or

   Delete unneeded reports or add new ones.

TIP: For More Information

For more information about configuring event notification in Novell AppArmor, refer to Section "Configuring Security Event Notification" (Chapter 6, *Managing Profiled Applications*, ↑Novell AppArmor Administration Guide). Find more information about report configuration in Section "Configuring Reports" (Chapter 6, *Managing Profiled Applications*, ↑Novell AppArmor Administration Guide).

41.3.4 Updating Your Profiles

Software and system configurations change over time. As a result of that, your profile setup for AppArmor might need some fine-tuning from time to time. AppArmor checks your system log for policy violations or other AppArmor events and lets you adjust your profile set accordingly. Any application behavior that is outside of any profile definition can also be addressed using the *Update Profile Wizard*.

To update your profile set, proceed as follows:

1. Log in as `root` and start YaST.

2. Start *Novell AppArmor* > *Update Profile Wizard*.

3. Adjust access or execute rights to any resource or for any executable that has been logged when prompted.

4 Leave YaST after you answer all questions. Your changes are applied to the respective profiles.

TIP: For More Information

For more information about updating your profiles from the system logs, refer to Section “Updating Profiles from Log Entries” (Chapter 3, *Building and Managing Profiles with YaST*, ↑Novell AppArmor Administration Guide).

42 Security and Confidentiality

One of the main characteristics of a Linux or UNIX system is its ability to handle several users at the same time (multiuser) and to allow these users to perform several tasks (multitasking) on the same computer simultaneously. Moreover, the operating system is network transparent. The users often do not know whether the data and applications they are using are provided locally from their machine or made available over the network.

With the multiuser capability, the data of different users must be stored separately. Security and privacy need to be guaranteed. Data security was already an important issue, even before computers could be linked through networks. Just like today, the most important concern was the ability to keep data available in spite of a lost or otherwise damaged data medium, a hard disk in most cases.

This section is primarily focused on confidentiality issues and on ways to protect the privacy of users, but it cannot be stressed enough that a comprehensive security concept should always include procedures to have a regularly updated, workable, and tested backup in place. Without this, you could have a very hard time getting your data back—not only in the case of some hardware defect, but also if the suspicion arises that someone has gained unauthorized access and tampered with files.

42.1 Local Security and Network Security

There are several ways of accessing data:

- personal communication with people who have the desired information or access to the data on a computer

- directly from the console of a computer (physical access)

- over a serial line

- using a network link

In all these cases, a user should be authenticated before accessing the resources or data in question. A Web server might be less restrictive in this respect, but you still would not want it to disclose all your personal data to any surfer.

In the list above, the first case is the one where the highest amount of human interaction is involved, such as when you are contacting a bank employee and are required to prove that you are the person owning that bank account. Then you are asked to provide a signature, a PIN, or a password to prove that you are the person you claim to be. In some cases, it might be possible to elicit some intelligence from an informed person just by mentioning known bits and pieces to win the confidence of that person by using clever rhetoric. The victim could be led to reveal gradually more information, maybe without even becoming aware of it. Among hackers, this is called *social engineering*. You can only guard against this by educating people and by dealing with language and information in a conscious way. Before breaking into computer systems, attackers often try to target receptionists, service people working with the company, or even family members. In many cases, such an attack based on social engineering is only discovered at a much later time.

A person wanting to obtain unauthorized access to your data could also use the traditional way and try to get at your hardware directly. Therefore, the machine should be protected against any tampering so that no one can remove, replace, or cripple its components. This also applies to backups and even any network cable or the power cord. Also secure the boot procedure, because there are some well-known key combinations that might provoke unusual behavior. Protect yourself against this by setting passwords for the BIOS and the boot loader.

Serial terminals connected to serial ports are still used in many places. Unlike network interfaces, they do not rely on a network protocol to communicate with the host. A simple cable or an infrared port is used to send plain characters back and forth between the devices. The cable itself is the weakest point of such a system: with an older printer connected to it, it is easy to record anything that runs over the wires. What can be achieved with a printer can also be accomplished in other ways, depending on the effort that goes into the attack.

Reading a file locally on a host requires other access rules than opening a network connection with a server on a different host. There is a distinction between local security and network security. The line is drawn where data must be put into packets to be sent somewhere else.

42.1.1 Local Security

Local security starts with the physical environment in the location where the computer is running. Set up your machine in a place where security is in line with your expectations and needs. The main goal of local security is to keep users separate from each other, so no user can assume the permissions or the identity of another. This is a general rule to be observed, but it is especially true for the user `root`, who holds the supreme power on the system. `root` can take on the identity of any other local user without being prompted for the password and read any locally stored file.

42.1.2 Passwords

On a Linux system, passwords are not stored as plain text and the text string entered is not simply matched with the saved pattern. If this were the case, all accounts on your system would be compromised as soon as someone got access to the corresponding file. Instead, the stored password is encrypted and, each time it is entered, is encrypted again and the two encrypted strings are compared. This only provides more security if the encrypted password cannot be reverse-computed into the original text string.

This is actually achieved by a special kind of algorithm, also called *trapdoor algorithm*, because it only works in one direction. An attacker who has obtained the encrypted string is not able to get your password by simply applying the same algorithm again. Instead, it would be necessary to test all the possible character combinations until a combination is found that looks like your password when encrypted. With passwords eight characters long, there are quite a number of possible combinations to calculate.

In the seventies, it was argued that this method would be more secure than others due to the relative slowness of the algorithm used, which took a few seconds to encrypt just one password. In the meantime, however, PCs have become powerful enough to do several hundred thousand or even millions of encryptions per second. Because of this, encrypted passwords should not be visible to regular users (`/etc/shadow` cannot be read by normal users). It is even more important that passwords are not easy to guess, in case the password file becomes visible due to some error. Consequently, it is not really useful to "translate" a password like "tantalize" into "t@nt@1lz3".

Replacing some letters of a word with similar looking numbers is not safe enough. Password cracking programs that use dictionaries to guess words also play with substitutions like that. A better way is to make up a word with no common meaning, something that only makes sense to you personally, like the first letters of the words of a sentence or the title of a book, such as "The Name of the Rose" by Umberto Eco. This would give the following safe password: "TNotRbUE9". In contrast, passwords like "beerbuddy" or "jasmine76" are easily guessed even by someone who has only some casual knowledge about you.

42.1.3 The Boot Procedure

Configure your system so it cannot be booted from a floppy or from CD, either by removing the drives entirely or by setting a BIOS password and configuring the BIOS to allow booting from a hard disk only. Normally, a Linux system is started by a boot loader, allowing you to pass additional options to the booted kernel. Prevent others from using such parameters during boot by setting an additional password in `/boot/grub/menu.lst` (see Chapter 13, *The Boot Loader* (page 195)). This is crucial to your system's security. Not only does the kernel itself run with `root` permissions, but it is also the first authority to grant `root` permissions at system start-up.

42.1.4 File Permissions

As a general rule, always work with the most restrictive privileges possible for a given task. For example, it is definitely not necessary to be `root` to read or write e-mail. If the mail program has a bug, this bug could be exploited for an attack that acts with exactly the permissions of the program when it was started. By following the above rule, minimize the possible damage.

The permissions of all files included in the openSUSE distribution are carefully chosen. A system administrator who installs additional software or other files should take great care when doing so, especially when setting the permission bits. Experienced and security-conscious system administrators always use the `-l` option with the command `ls` to get an extensive file list, which allows them to detect any incorrect file permissions immediately. An incorrect file attribute does not only mean that files could be changed or deleted. These modified files could be executed by `root` or, in the case of configuration files, programs could use such files with the permissions of `root`. This significantly increases the possibilities of an attacker. Attacks like this are called cuckoo eggs, because the program (the egg) is executed (hatched) by a different user (bird), just like a cuckoo tricks other birds into hatching its eggs.

A openSUSE® system includes the files `permissions`, `permissions.easy`, `permissions.secure`, and `permissions.paranoid`, all in the directory `/etc`. The purpose of these files is to define special permissions, such as world-writable directories or, for files, the setuser ID bit (programs with the setuser ID bit set do not run with the permissions of the user that has launched it, but with the permissions of the file owner, in most cases `root`). An administrator can use the file `/etc/permissions.local` to add his own settings.

To define which of the above files is used by openSUSE's configuration programs to set permissions accordingly, select *Local Security* in the *Security and Users* section of YaST. To learn more about the topic, read the comments in `/etc/permissions` or consult the manual page of `chmod` (`man chmod`).

42.1.5 Buffer Overflows and Format String Bugs

Special care must be taken whenever a program is supposed to process data that can or could be changed by a user, but this is more of an issue for the programmer of an application than for regular users. The programmer must make sure that his application interprets data in the correct way, without writing it into memory areas that are too small to hold it. Also, the program should hand over data in a consistent manner, using the interfaces defined for that purpose.

A *buffer overflow* can happen if the actual size of a memory buffer is not taken into account when writing to that buffer. There are cases where this data (as generated by the user) uses up some more space than what is available in the buffer. As a result, data is written beyond the end of that buffer area, which, under certain circumstances, makes it possible for a program to execute program sequences influenced by the user (and not by the programmer), rather than just processing user data. A bug of this kind may have serious consequences, especially if the program is being executed with special privileges (see Section 42.1.4, “File Permissions” (page 664)).

Format string bugs work in a slightly different way, but again it is the user input that could lead the program astray. In most cases, these programming errors are exploited with programs executed with special permissions—setuid and setgid programs—which also means that you can protect your data and your system from such bugs by removing the corresponding execution privileges from programs. Again, the best way is to apply a policy of using the lowest possible privileges (see Section 42.1.4, “File Permissions” (page 664)).

Given that buffer overflows and format string bugs are bugs related to the handling of user data, they are not only exploitable if access has been given to a local account. Many of the bugs that have been reported can also be exploited over a network link. Accordingly, buffer overflows and format string bugs should be classified as being relevant for both local and network security.

42.1.6 Viruses

Contrary to what some people say, there are viruses that run on Linux. However, the viruses that are known were released by their authors as a *proof of concept* to prove that the technique works as intended. None of these viruses have been spotted *in the wild* so far.

Viruses cannot survive and spread without a host on which to live. In this case, the host would be a program or an important storage area of the system, such as the master boot record, which needs to be writable for the program code of the virus. Owing to its multiuser capability, Linux can restrict write access to certain files, especially important with system files. Therefore, if you did your normal work with `root` permissions, you would increase the chance of the system being infected by a virus. In contrast, if you follow the principle of using the lowest possible privileges as mentioned above, chances of getting a virus are slim.

Apart from that, you should never rush into executing a program from some Internet site that you do not really know. openSUSE's RPM packages carry a cryptographic signature as a digital label that the necessary care was taken to build them. Viruses are a typical sign that the administrator or the user lacks the required security awareness, putting at risk even a system that should be highly secure by its very design.

Viruses should not be confused with worms, which belong to the world of networks entirely. Worms do not need a host to spread.

42.1.7 Network Security

Network security is important for protecting from an attack that is started outside. The typical login procedure requiring a username and a password for user authentication is still a local security issue. In the particular case of logging in over a network, differentiate between the two security aspects. What happens until the actual authentication is network security and anything that happens afterwards is local security.

42.1.8 X Window System and X Authentication

As mentioned at the beginning, network transparency is one of the central characteristics of a UNIX system. X, the windowing system of UNIX operating systems, can make use of this feature in an impressive way. With X, it is basically no problem to log in at a remote host and start a graphical program that is then sent over the network to be displayed on your computer.

When an X client should be displayed remotely using an X server, the latter should protect the resource managed by it (the display) from unauthorized access. In more concrete terms, certain permissions must be given to the client program. With the X Window System, there are two ways to do this, called host-based access control and cookie-based access control. The former relies on the IP address of the host where the client should run. The program to control this is xhost. xhost enters the IP address of a legitimate client into a tiny database belonging to the X server. However, relying on IP addresses for authentication is not very secure. For example, if there were a second user working on the host sending the client program, that user would have access to the X server as well—just like someone stealing the IP address. Because of these shortcomings, this authentication method is not described in more detail here, but you can learn about it with `man xhost`.

In the case of cookie-based access control, a character string is generated that is only known to the X server and to the legitimate user, just like an ID card of some kind. This cookie (the word goes back not to ordinary cookies, but to Chinese fortune cookies, which contain an epigram) is stored on login in the file `.Xauthority` in the user's home directory and is available to any X client wanting to use the X server to display a window. The file `.Xauthority` can be examined by the user with the tool `xauth`. If you were to rename `.Xauthority` or if you deleted the file from your home directory by accident, you would not be able to open any new windows or X clients.

SSH (secure shell) can be used to encrypt a network connection completely and forward it to an X server transparently without the encryption mechanism being perceived by the user. This is also called X forwarding. X forwarding is achieved by simulating an X server on the server side and setting a DISPLAY variable for the shell on the remote host. Further details about SSH can be found in Chapter 36, *SSH—Secure Network Operations* (page 589).

WARNING

If you do not consider the host where you log in to be a secure host, do not use X forwarding. With X forwarding enabled, an attacker could authenticate via your SSH connection to intrude on your X server and sniff your keyboard input, for instance.

42.1.9 Buffer Overflows and Format String Bugs

As discussed in Section 42.1.5, "Buffer Overflows and Format String Bugs" (page 665), buffer overflows and format string bugs should be classified as issues concerning both local and network security. As with the local variants of such bugs, buffer overflows in network programs, when successfully exploited, are mostly used to obtain `root` permissions. Even if that is not the case, an attacker could use the bug to gain access to an unprivileged local account to exploit any other vulnerabilities that might exist on the system.

Buffer overflows and format string bugs exploitable over a network link are certainly the most frequent form of remote attacks in general. Exploits for these—programs to exploit these newly-found security holes—are often posted on the security mailing lists. They can be used to target the vulnerability without knowing the details of the code.

Over the years, experience has shown that the availability of exploit codes has contributed to more secure operating systems, obviously due to the fact that operating system makers were forced to fix the problems in their software. With free software, anyone has access to the source code (openSUSE comes with all available source codes) and anyone who finds a vulnerability and its exploit code can submit a patch to fix the corresponding bug.

42.1.10 Denial of Service

The purpose of a denial of service (DoS) attack is to block a server program or even an entire system, something that could be achieved by various means: overloading the server, keeping it busy with garbage packets, or exploiting a remote buffer overflow. Often a DoS attack is made with the sole purpose of making the service disappear. However, once a given service has become unavailable, communications could become vulnerable to *man-in-the-middle attacks* (sniffing, TCP connection hijacking, spoofing) and DNS poisoning.

42.1.11 Man in the Middle: Sniffing, Hijacking, Spoofing

In general, any remote attack performed by an attacker who puts himself between the communicating hosts is called a *man-in-the-middle attack*. What almost all types of man-in-the-middle attacks have in common is that the victim is usually not aware that there is something happening. There are many possible variants, for example, the attacker could pick up a connection request and forward that to the target machine. Now the victim has unwittingly established a connection with the wrong host, because the other end is posing as the legitimate destination machine.

The simplest form of a man-in-the-middle attack is called *sniffer*—the attacker is “just” listening to the network traffic passing by. As a more complex attack, the “man in the middle” could try to take over an already established connection (hijacking). To do so, the attacker would need to analyze the packets for some time to be able to predict the TCP sequence numbers belonging to the connection. When the attacker finally seizes the role of the target host, the victims notice this, because they get an error message saying the connection was terminated due to a failure. The fact that there are protocols not secured against hijacking through encryption, which only perform a simple authentication procedure upon establishing the connection, makes it easier for attackers.

Spoofing is an attack where packets are modified to contain counterfeit source data, usually the IP address. Most active forms of attack rely on sending out such fake packets—something that, on a Linux machine, can only be done by the superuser (`root`).

Many of the attacks mentioned are carried out in combination with a DoS. If an attacker sees an opportunity to bring down a certain host abruptly, even if only for a short time, it makes it easier for him to push the active attack, because the host will not be able to interfere with the attack for some time.

42.1.12 DNS Poisoning

DNS poisoning means that the attacker corrupts the cache of a DNS server by replying to it with spoofed DNS reply packets, trying to get the server to send certain data to a victim who is requesting information from that server. Many servers maintain a trust relationship with other hosts, based on IP addresses or hostnames. The attacker needs a good understanding of the actual structure of the trust relationships among hosts to disguise itself as one of the trusted hosts. Usually, the attacker analyzes some packets received from the server to get the necessary information. The attacker often needs to target a well-timed DoS attack at the name server as well. Protect yourself by using encrypted connections that are able to verify the identity of the hosts to which to connect.

42.1.13 Worms

Worms are often confused with viruses, but there is a clear difference between the two. Unlike viruses, worms do not need to infect a host program to live. Instead, they are specialized to spread as quickly as possible on network structures. The worms that appeared in the past, such as Ramen, Lion, or Adore, make use of well-known security holes in server programs like bind8 or lprNG. Protection against worms is relatively easy. Given that some time elapses between the discovery of a security hole and the moment the worm hits your server, there is a good chance that an updated version of the affected program is available on time. That is only useful if the administrator actually installs the security updates on the systems in question.

42.2 Some General Security Tips and Tricks

To handle security competently, it is important to keep up with new developments and stay informed about the latest security issues. One very good way to protect your systems against problems of all kinds is to get and install the updated packages recommended by security announcements as quickly as possible. SUSE security announcements are published on the list opensuse-security-announce@opensuse.org. It is a first-hand source of information regarding updated packages and includes members of SUSE's security team among its active contributors. You can subscribe to this list on page http://en.opensuse.org/Communicate#Mailing_Lists.

The mailing list opensuse-security@opensuse.org is a good place to discuss any security issues of interest. Subscribe to it on the same Web page.

bugtraq@securityfocus.com is one of the best-known security mailing lists worldwide. Reading this list, which receives between 15 and 20 postings per day, is recommended. More information can be found at http://www.securityfocus.com.

The following is a list of rules you may find useful in dealing with basic security concerns:

- According to the rule of using the most restrictive set of permissions possible for every job, avoid doing your regular jobs as `root`. This reduces the risk of getting a cuckoo egg or a virus and protects you from your own mistakes.

- If possible, always try to use encrypted connections to work on a remote machine. Using `ssh` (secure shell) to replace `telnet`, `ftp`, `rsh`, and `rlogin` should be standard practice.

- Avoid using authentication methods based on IP addresses alone.

- Try to keep the most important network-related packages up-to-date and subscribe to the corresponding mailing lists to receive announcements on new versions of such programs (bind, postfix, ssh, etc.). The same should apply to software relevant to local security.

- Change the `/etc/permissions` file to optimize the permissions of files crucial to your system's security. If you remove the setuid bit from a program, it might well be that it cannot do its job anymore in the intended way. On the other hand, consider that, in most cases, the program will also have ceased to be a potential security risk. You might take a similar approach with world-writable directories and files.

- Disable any network services you do not absolutely require for your server to work properly. This makes your system safer. Open ports, with the socket state LISTEN, can be found with the program netstat. As for the options, it is recommended to use `netstat -ap` or `netstat -anp`. The `-p` option allows you to see which process is occupying a port under which name.

  Compare the netstat results with those of a thorough port scan done from outside your host. An excellent program for this job is nmap, which not only checks out the ports of your machine, but also draws some conclusions as to which services are waiting behind them. However, port scanning may be interpreted as an aggressive act, so do not do this on a host without the explicit approval of the administrator. Finally, remember that it is important not only to scan TCP ports, but also UDP ports (options `-sS` and `-sU`).

- To monitor the integrity of the files of your system in a reliable way, use the program AIDE (Advanced Intrusion Detection Environment), available on openSUSE. Encrypt the database created by AIDE to prevent someone from tampering with it. Furthermore, keep a backup of this database available outside your machine, stored on an external data medium not connected to it by a network link.

- Take proper care when installing any third-party software. There have been cases where a hacker had built a trojan horse into the tar archive of a security software package, which was fortunately discovered very quickly. If you install a binary package, have no doubts about the site from which you downloaded it.

  SUSE's RPM packages are gpg-signed. The key used by SUSE for signing is:

  ID:9C800ACA 2000-10-19 SUSE Package Signing Key <build@suse.de>

  Key fingerprint = 79C1 79B2 E1C8 20C1 890F 9994 A84E DAE8 9C80 0ACA

  The command `rpm --checksig package.rpm` shows whether the checksum and the signature of an uninstalled package are correct. Find the key on the first CD of the distribution and on most key servers worldwide.

- Check your backups of user and system files regularly. Consider that if you do not test whether the backup works, it might actually be worthless.

- Check your log files. Whenever possible, write a small script to search for suspicious entries. Admittedly, this is not exactly a trivial task. In the end, only you can know which entries are unusual and which are not.

- Use `tcp_wrapper` to restrict access to the individual services running on your machine, so you have explicit control over which IP addresses can connect to a service. For further information regarding `tcp_wrapper`, consult the manual pages of tcpd and hosts_access (`man 8 tcpd`, `man hosts_access`).

- Use SuSEfirewall to enhance the security provided by tcpd (tcp_wrapper).

- Design your security measures to be redundant: a message seen twice is much better than no message at all.

42.3 Using the Central Security Reporting Address

If you discover a security-related problem (please check the available update packages first), write an e-mail to security@suse.de. Please include a detailed description of the problem and the version number of the package concerned. SUSE will try to send a reply as soon as possible. You are encouraged to pgp encrypt your e-mail messages. SUSE's pgp key is:

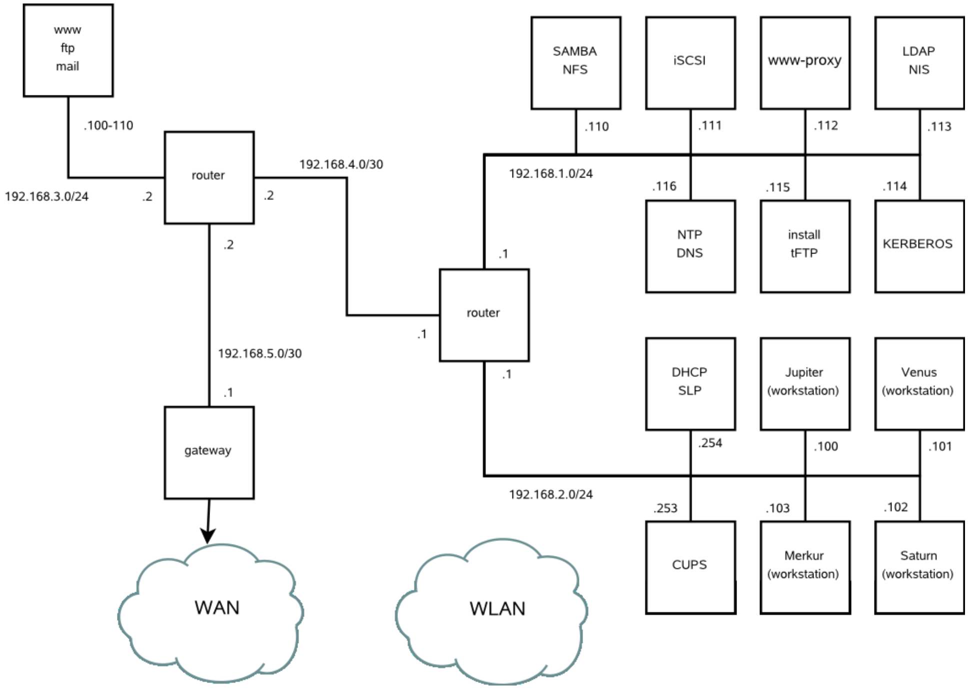
```
ID:3D25D3D9 1999-03-06 SUSE Security Team <security@suse.de>
Key fingerprint = 73 5F 2E 99 DF DB 94 C4 8F 5A A3 AE AF 22 F2 D5
```

This key is also available for download from http://www.novell.com/linux/security/securitysupport.html.

A

An Example Network

This example network is used across all network-related chapters of the openSUSE® documentation.



B

GNU Licenses

This appendix contains the GNU General Public License and the GNU Free Documentation License.

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and an idea of what it does. Copyright (C) yyyy name of author

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.


This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.


You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'. This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program `Gnomovision'
(which makes passes at compilers) written
by James Hacker.


signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License [http://www.fsf.org/licenses/lgpl.html] instead of this License.

GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document’s license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c) YEAR YOUR NAME.
```

```
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
only as published by the Free Software Foundation;
with the Invariant Section being this copyright notice and license.
A copy of the license is included in the section entitled “GNU
Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Index

Symbols

64-bit Linux, 175
- kernel specifications, 178
- runtime support, 175
- software development, 176

A

access permissions (see permissions)
ACLs, 251-262
- access, 253, 256
- check algorithm, 261
- default, 254, 259
- definitions, 253
- effects, 259
- handling, 254
- masks, 258
- permission bits, 255
- structure, 254
- support, 262

add-on medium
- language support, 84

Apache, 481-520
- CGI scripts, 507
- configuring, 483
  - files, 483
  - manually, 483-490
  - virtual host, 486
  - YaST, 490-497
- installing, 482
- modules, 499-506
  - available, 500
  - building, 506
  - external, 504
  - installing, 500
  - multiprocessing, 503
- quick start, 481
- security, 515
- SSL, 509-515
  - configure Apache with SSL, 514
  - creating an SSL certificate, 510
- starting, 497
- stopping, 497
- troubleshooting, 517

authentication
- PAM, 263-270

B

Bash
- .bashrc, 216
- .profile, 216
- profile, 215

BIND, 361-372
Bluetooth, 553
- hciconfig, 557
- hcitool, 556
- pand, 558
- sdptool, 557

booting, 179
- boot sectors, 195, 196
- configuring
  - YaST, 205
- graphic, 210
- GRUB, 195, 196
- initramfs, 181
- initrd, 181

C

cards
- graphics, 122
- network, 315, 316

cat, 286
cd, 282
chgrp, 280, 283
chmod, 279, 283
chown, 280, 282

CJK, 223
clear, 291
commands, 280-291
  cat, 286
  cd, 282
  chgrp, 280, 283
  chmod, 279, 283
  chown, 280, 282
  clear, 291
  cp, 281
  date, 289
  df, 288
  diff, 287
  du, 288
  file, 286
  find, 285
  fonts-config, 124
  free, 220, 288
  getfacl, 257
  grep, 286
  grub, 196
  gzip, 284
  halt, 291
  ifconfig, 343
  ip, 340
  kadmin, 627
  kill, 289
  killall, 289
  kinit, 634
  ktadd, 637
  ldapadd, 436
  ldapdelete, 439
  ldapmodify, 437
  ldapsearch, 438, 640
  less, 286
  ln, 282
  locate, 285
  lp, 104
  ls, 281
  man, 280
  mkdir, 282
  mount, 287
  mv, 282
  nslookup, 290
  passwd, 291
  ping, 290, 341
  ps, 289
  reboot, 291
  rm, 282
  rmdir, 282
  route, 344
  rpm, 85
  rpmbuild, 85
  scp, 590
  setfacl, 257
  sftp, 591
  slptool, 350
  smbpasswd, 479
  ssh, 590
  ssh-agent, 593
  ssh-keygen, 592
  su, 291
  tar, 284
  telnet, 290
  top, 289
  umount, 287
  updatedb, 285
configuration files, 333
  .bashrc, 216, 219
  .emacs, 221
  .profile, 216
  .xsession, 593
  acpi, 529
  crontab, 216
  csh.cshrc, 225
  dhclient.conf, 387
  dhcp, 334
  dhcpd.conf, 388
  fstab, 46, 287
  group, 76

grub.conf, 203
host.conf, 336
HOSTNAME, 340
hosts, 315, 336
ifcfg-*, 333
inittab, 183, 185, 222
inputrc, 222
irda, 561
kernel, 181
krb5.conf, 631, 632, 634, 638
krb5.keytab, 637
language, 223, 225
logrotate.conf, 217
menu.lst, 198
named.conf, 362, 363-372
network, 334
networks, 336
nscd.conf, 339
nsswitch.conf, 337
openldap, 640
passwd, 76
permissions, 672
powersave, 529
profile, 215, 219, 225
resolv.conf, 220, 335, 362
routes, 334
samba, 473
services, 473
slapd.conf, 429, 641
smb.conf, 473, 479
smppd.conf, 346
smpppd-c.conf, 347
sshd_config, 594, 638
ssh_config, 639
suseconfig, 194
sysconfig, 192-194
termcap, 222
wireless, 334
xorg.conf, 117
    Device, 121
    Monitor, 122
    Screen, 120
configuring, 192
    cable modem, 329
    DNS, 353
    DSL, 329
    FTP server, 521
    GRUB, 196, 203
    IPv6, 313
    IrDA, 561
    ISDN, 326
    modems, 324
    networks, 316
        manually, 332-345
    routing, 334
    Samba, 471-477
        clients, 477
    SSH, 589
    T-DSL, 331
consoles
    assigning, 222
    graphical, 210
    switching, 222
core files, 219
cp, 281
cpuspeed, 537
cron, 216

D

date, 289
deltarpm, 89
df, 288
DHCP, 375-391
    configuring with YaST, 376
    dhcpd, 388-389
    packages, 387
    server, 388-389
    static address assignment, 389
diff, 287

directories
- /, 273
- /bin, 273, 274
- /boot, 273, 274
- /dev, 273, 274
- /etc, 273, 274
- /home, 273, 274
- /lib, 273, 275
- /media, 273, 275
- /mnt, 273, 275
- /opt, 273, 275
- /root, 273, 275
- /sbin, 273, 275
- /srv, 273, 275
- /tmp, 274, 275
- /usr, 274, 276
- /var, 274, 276
- /windows, 274, 277
- changing, 282
- creating, 282
- deleting, 282
- structure, 273

disks
- boot, 209

DNS, 314
- BIND, 361-372
- configuring, 353
- domains, 335
- forwarding, 362
- logging, 366
- mail exchanger, 315
- name servers, 335
- NIC, 315
- options, 364
- reverse lookup, 371
- security and, 670
- starting, 362
- terminology, 353
- top level domain, 314
- troubleshooting, 362
- zones
  - files, 367

domain name system (see DNS)

DOS
- sharing files, 469

drives
- mounting, 287
- unmounting, 287

du, 288

E

editors
- Emacs, 221-222
- vi, 291

Emacs, 221-222
- .emacs, 221
- default.el, 221

encoding
- ISO-8859-1, 225

encrypting, 643-647
- creating partitions, 645
- files, 646-649
- files with vi, 649
- partitions, 644-646
- removable media, 647
- YaST, with, 644

error messages
- bad interpreter, 46
- permission denied, 46

F

file, 286

file systems, 241-250
- ACLs, 251-262
- changing, 44
- cryptofs, 643
- encrypting, 643
- Ext2, 243-244
- Ext3, 244-245

- LFS, 248
- limitations, 248
- ReiserFS, 242-243
- selecting, 242
- supported, 247-248
- terms, 241
- XFS, 246

files
- archiving, 284
- comparing, 287
- compressing, 284
- copying, 281
- deleting, 282
- encrypting, 646
- finding, 218
- moving, 282
- searching contents, 286
- searching for, 285
- viewing, 286

find, 285

Firefox
- URL open command, 81

firewalls, 577
- packet filters, 577, 581
- SuSEfirewall2, 577, 582

fonts, 124
- TrueType, 123
- X11 core, 124
- Xft, 125

free, 288

FTP server
- configuring, 521

G

GNOME
- shell, 272

graphics
- cards
  - drivers, 122

grep, 286

GRUB, 195
- boot menu, 198
- boot password, 204
- boot sectors, 196
- booting, 196
- commands, 196
- device names, 199
- device.map, 197, 202
- GRUB Geom Error, 211
- grub.conf, 197, 203
- limitations, 196
- Master Boot Record (MBR), 195
- menu editor, 201
- menu.lst, 197, 198
- partition names, 199
- troubleshooting, 211
- uninstalling, 209

gzip, 284

H

halt, 291

hardware
- ISDN, 326

hciconfig, 557

hcitool, 556

help
- info pages, 220
- man pages, 220, 280
- Novell/SUSE manuals, xiv
- X, 123

I

I18N, 223

info pages, 220

init, 183
- adding scripts, 188
- inittab, 183
- scripts, 186-190

installing
  GRUB, 196
  packages, 86
internationalization, 223
Internet
  cinternet, 347
  dial-up, 345-347
  DSL, 329
  ISDN, 326
  KInternet, 347
  qinternet, 347
  smpppd, 345-347
  TDSL, 331
IP addresses, 302
  classes, 303
  dynamic assignment, 375
  IPv6, 305
    configuring, 313
  masquerading, 580
  private, 305
IrDA, 560-563
  configuring, 561
  starting, 561
  stopping, 561
  troubleshooting, 562

K

KDE
  shell, 272
Kerberos, 613-619
  administering, 621-642
  authenticators, 614
  clients
    configuring, 631-633
  clock skew, 633
  clock synchronization, 625
  configuring
    clients, 631-633
  credentials, 614
  installing, 621-642
  KDC, 624-625, 626-628
    administering, 634
    nsswitch.conf, 625
    starting, 628
  keytab, 637
  LDAP and, 639-642
  master key, 627
  PAM support, 637-638
  principals, 614
    creating, 628
    host, 636
    service, 636
  realms, 623
    creating, 627
  session key, 614
  SSH configuration, 638
  stash file, 627
  ticket-granting service, 617
  tickets, 614, 617
kernel
  standard kernel, 84
kernels
  caches, 220
  limits, 249
keyboard
  Asian characters, 223
  layout, 222
  mapping, 222
    compose, 222
    multikey, 222
  X Keyboard Extension, 223
  XKB, 223
kill, 289
killall, 289

L

L10N, 223
laptops

IrDA, 560-563
power management, 527-537
LDAP, 409-440
access control, 433
ACLs, 431
adding data, 435
administering groups, 426
administering users, 426
configuring
YaST, 414
deleting data, 439
directory tree, 411
Kerberos and, 639-642
ldapadd, 435
ldapdelete, 439
ldapmodify, 437
ldapsearch, 438
modifying data, 437
searching data, 438
server configuration
manual, 429
YaST, 414
YaST
client, 419
modules, 419
templates, 419
less, 286
LFS, 248
Lightweight Directory Access Protocol (see LDAP)
Linux
networks and, 299
sharing files with another OS, 469
uninstalling, 209
ln, 282
localization, 223
locate, 218, 285
log files, 217
boot.msg, 529
messages, 362, 586
Logical Volume Manager (see LVM)
logrotate, 217
ls, 281
LSB
installing packages, 85
LVM
YaST, 49

M

man pages, 220, 280
masquerading, 580
configuring with SuSEfirewall2, 582
Master Boot Record (see MBR)
MBR, 195, 196
memory
RAM, 220
mkdir, 282
modems
cable, 329
YaST, 324
mount, 287
mv, 282

N

name servers (see DNS)
NAT (see masquerading)
NetBIOS, 469
Network File System (see NFS)
Network Information Service (see NIS)
networks, 299
authentication
Kerberos, 613-619
base network address, 304
broadcast address, 304
configuration files, 333-340
configuring, 315-331, 332-345
IPv6, 313
DHCP, 375
DNS, 314

localhost, 305
netmasks, 303
routing, 302, 303
SLP, 349
TCP/IP, 299
YaST, 316
alias, 319
gateway, 320
hostname, 319
IP address, 318
starting, 321
NFS, 455
clients, 456
exporting, 463
importing, 457
mounting, 457
servers, 459
NIS, 401-408
clients, 407
masters, 401-407
slaves, 401-407
Novell/SUSE manuals, xiv
nslookup, 290
NSS, 337
databases, 338

O

OpenLDAP (see LDAP)
OpenSSH (see SSH)
OS/2
sharing files, 469

P

packages
compiling, 93
compiling with build, 95
installing, 86
LSB, 85
package manager, 85
RPMs, 85
uninstalling, 86
verifying, 86
packet filters (see firewalls)
PAM, 263-270
pand, 558
partitions
creating, 41, 43
encrypting, 645
fstab, 46
LVM, 44
parameters, 44
partition table, 195
RAID, 44
reformatting, 44
swap, 44
types, 42
passwd, 291
passwords
changing, 291
PCMCIA
IrDA, 560-563
permissions, 277
ACLs, 251-262
changing, 279, 283
directories, 278
file permissions, 218
file systems, 277
files, 277
viewing, 279
ping, 290, 341
Pluggable Authentication Modules (see PAM)
ports
53, 364
PostgreSQL
updating, 76
power management, 527-542
ACPI, 527, 528-535
battery monitor, 528

cpufrequency, 537
cpuspeed, 537
hibernation, 528
powersave, 537
standby, 527
suspend, 527
powersave, 537
configuring, 537
printing, 97
command line, 104
CUPS, 103
GDI printers, 108
IrDA, 561
kprinter, 103
network, 110
Samba, 470
troubleshooting
network, 110
xpp, 103
private branch exchange, 328
processes, 289
killing, 289
overview, 289
protocols
CIFS, 469
IPv6, 305
LDAP, 409
SLP, 349
SMB, 469
ps, 289

R

RAID
YaST, 55
reboot, 291
RFCs, 299
rm, 282
rmdir, 282
routing, 302, 334
masquerading, 580
netmasks, 303
routes, 334
static, 334
RPM, 85-96
database
rebuilding, 87, 93
deltarpm, 89
dependencies, 86
patches, 87
queries, 90
rpmnew, 86
rpmorig, 86
rpmsave, 86
security, 672
SRPMS, 93
tools, 96
uninstalling, 87
updating, 86
verify, 92
verifying, 86
rpmbuild, 85
runlevels, 183-186
changing, 185-186
editing in YaST, 190

S

Samba, 469-480
CIFS, 469
clients, 470, 477-478
configuring, 471-477
installing, 471
login, 478
names, 469
permissions, 477
printers, 470
printing, 478
security, 477
server, 470

servers, 471-477
shares, 470, 475
SMB, 469
starting, 471
stopping, 471
swat, 473
TCP/IP and, 469
screen
resolution, 121
scripts
init.d, 183, 186-190, 344
boot, 187
boot.local, 188
boot.setup, 188
halt, 188
network, 344
nfsserver, 345
portmap, 345
postfix, 345
rc, 185, 186, 188
xinetd, 345
ypbind, 345
ypserv, 345
irda, 561
mkinitrd, 181
modify_resolvconf, 220, 335
SuSEconfig, 192-194
disabling, 194
sdptool, 557
security, 661-673
attacks, 669-670
booting, 662, 664
bugs and, 665, 668
DNS, 670
engineering, 662
firewalls, 577
local, 663-667
network, 667-670
passwords, 663-664
permissions, 664-665
reporting problems, 673
RPM signatures, 672
Samba, 477
serial terminals, 662, 663
SSH, 589-595
tcpd, 673
telnet, 589
tips and tricks, 671
viruses, 666
worms, 670
X and, 667
Service Location Protocol (see SLP)
shells, 271-295
Bash, 271
commands, 280-291
SLP, 349
Konqueror, 350
providing services, 351
registering services, 351
slptool, 350
SMB (see Samba)
smbd, 469
soft RAID (see RAID)
software
compiling, 93
source
compiling, 93
spm, 93
SSH, 589-595
authentication mechanisms, 592
daemon, 591
key pairs, 591, 593
scp, 590
sftp, 591
ssh, 590
ssh-agent, 593, 594
ssh-keygen, 593
sshd, 591
X and, 593
su, 291

system
  limiting resource use, 219
  localizing, 223
  rebooting, 291
  shutdown, 291
  updating, 75-78

T

Tablet PCs, 565-573
  configuring, 567
  Dasher, 570
  installing, 566
  Jarnal, 569
  KRandRTray, 568
  Xournal, 569
  xstroke, 568
  xvkbd, 567
tar, 284
TCP/IP, 299
  ICMP, 300
  IGMP, 300
  layer model, 300
  packets, 301, 302
  TCP, 300
  UDP, 300
TEI XSL stylesheets
  new location, 80
telnet, 290
top, 289

U

ulimit, 219
  options, 219
umount, 287
uninstalling
  GRUB, 209
  Linux, 209
updatedb, 285
updating, 75-78
  online, 63
    command line, 66
  passwd and group, 76
  problems, 76
  YaST, 76
users
  /etc/passwd, 266

V

variables
  environment, 223
virtual memory, 44

W

whois, 315
wild cards, 285
Windows
  sharing files, 469
wireless connections
  Bluetooth, 553

X

X
  character sets, 123
  configuring, 117-123
  drivers, 122
  font systems, 124
  fonts, 123
  help, 123
  SaX2, 118
  security, 667
  SSH and, 593
  TrueType fonts, 123
  virtual screen, 121
  X11 core fonts, 124
  xft, 123
  Xft, 125
  xorg.config, 118

X Keyboard Extension (see keyboard, XKB)
X Window System (see X)
X.509 certification
- certificates, 599
- principles, 597
- repository, 601
- revocation list, 600
- YaST, 597

X.Org, 117
Xft, 125
XKB (see keyboard, XKB)
xorg.conf
- color depth, 120
- Depth, 120
- Device, 121
- Display, 120
- Files, 118
- InputDevice, 118
- Modeline, 121
- modelines, 119
- Modes, 119, 121
- modules, 118
- Monitor, 119, 120
- ServerFlags, 118

Y

YaST
- boot configuration, 205
  - default system, 208
  - security, 208
  - time-out, 208
- boot loader
  - location, 207
  - password, 208
  - type, 206
- CA management, 602
- cable modem, 329
- command line, 72
- DHCP, 376
- DSL, 329
- GRUB, 206
- ISDN, 326
- LDAP
  - clients, 419
  - servers, 414
- LILO, 206
- LVM, 49
- modems, 324
- ncurses, 69
- network card, 316
- NIS clients, 407
- online update, 63-65
- partitioning, 41
- RAID, 55
- runlevels, 190
- Samba
  - clients, 477
- sysconfig editor, 192
- T-DSL, 331
- text mode, 69-73
  - modules, 72
- updating, 76
- X.509 certification, 597
  - certificates, 605
  - changing default values, 607
  - creating CRLs, 609
  - exporting CA objects as a file, 611
  - exporting CA objects to LDAP, 609
  - importing general server certificates, 611
  - root CA, 602
  - sub-CA, 604

YP (see NIS)

Z

zypper, 66-68