SUSE Linux Enterprise Server

10 SP3 www.novell.com

October 27, 2009 | Installation and Administration



Installation and Administration

All content is copyright © Novell, Inc.

Legal Notice

This manual is protected under Novell intellectual property rights. By reproducing, duplicating or distributing this manual you explicitly agree to conform to the terms and conditions of this license agreement.

This manual may be freely reproduced, duplicated and distributed either as such or as part of a bundled package in electronic and/or printed format, provided however that the following conditions are fulfilled:

That this copyright notice and the names of authors and contributors appear clearly and distinctively on all reproduced, duplicated and distributed copies. That this manual, specifically for the printed format, is reproduced and/or distributed for noncommercial use only. The express authorization of Novell, Inc must be obtained prior to any other use of any manual or part thereof.

For Novell trademarks, see the Novell Trademark and Service Mark list http://www.novell.com/company/legal/trademarks/tmlist.html. * Linux is a registered trademark of Linus Torvalds. All other third party trademarks are the property of their respective owners. A trademark symbol (®, ™ etc.) denotes a Novell trademark; an asterisk (*) denotes a third party trademark.

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 xv

Part I Deployment 1

1 Planning for SUSE Linux Enterprise 3

1.1 Considerations for Deployment of a SUSE Linux Enterprise 5
1.2 Deployment of SUSE Linux Enterprise 5
1.3 Running SUSE Linux Enterprise 6

2 Deployment Strategies 7

2.1 Deploying up to 10 Workstations 7
2.2 Deploying up to 100 Workstations 9
2.3 Deploying More than 100 Workstations 16

3 Installation with YaST 17

3.1 IBM POWER: System Start-Up for Network Installation 17
3.2 IBM System z: System Start-Up for Installation 18
3.3 System Start-Up for Installation 18
3.4 The Installation Workflow 20
3.5 The Boot Screen 20
3.6 Language 24
3.7 IBM System z: Hard Disk Configuration 24
3.8 Media Check 27
3.9 License Agreement 27
3.10 Installation Mode 28
3.11 Clock and Time Zone 29
3.12 Installation Settings 29

3.13 Performing the Installation 34
3.14 Configuration of the Installed System 36
3.15 Graphical Login 45

4 Remote Installation 47

4.1 Installation Scenarios for Remote Installation 47
4.2 Setting Up the Server Holding the Installation Sources 56
4.3 Preparing the Boot of the Target System 66
4.4 Booting the Target System for Installation 76
4.5 Monitoring the Installation Process 81

5 Automated Installation 85

5.1 Simple Mass Installation 85
5.2 Rule-Based Autoinstallation 97
5.3 For More Information 102

6 Deploying Customized Preinstallations 103

6.1 Preparing the Master Machine 104
6.2 Customizing the Firstboot Installation 104
6.3 Cloning the Master Installation 113
6.4 Personalizing the Installation 113

7 Advanced Disk Setup 115

7.1 LVM Configuration 115
7.2 Soft RAID Configuration 123

8 System Configuration with YaST 129

8.1 YaST Language 130
8.2 The YaST Control Center 130
8.3 Software 132
8.4 Hardware 146
8.5 System 153
8.6 Network Devices 164
8.7 Network Services 165
8.8 AppArmor 172
8.9 Security and Users 172
8.10 Virtualization 181
8.11 Miscellaneous 182
8.12 YaST in Text Mode 185
8.13 Managing YaST from the Command Line 188

8.14 SaX2 . . . 191
8.15 Troubleshooting . . . 197
8.16 For More Information . . . 197

9 Managing Software with ZENworks 199

9.1 Update from the Command Line with rug . . . 200
9.2 Managing Packages with the ZEN Tools . . . 204
9.3 For More Information . . . 209

10 Updating SUSE Linux Enterprise 211

10.1 Updating SUSE Linux Enterprise . . . 211
10.2 Installing Service Packs . . . 214
10.3 Software Changes from Version 9 to Version 10 . . . 224

Part II Administration 239

11 OpenWBEM 241

11.1 Setting Up OpenWBEM . . . 243
11.2 Changing the OpenWBEM CIMOM Configuration . . . 248
11.3 For More Information . . . 269

12 Mass Storage over IP Networks—iSCSI 271

12.1 Setting Up an iSCSI Target . . . 271
12.2 Configuring iSCSI Initiator . . . 277

13 iSNS for Linux Overview 283

13.1 How iSNS Works . . . 283
13.2 iSNS for Linux Installation and Setup . . . 285
13.3 Setting Up iSNS . . . 285
13.4 For More Information . . . 288

14 Oracle Cluster File System 2 289

14.1 O2CB Cluster Service . . . 291
14.2 Disk Heartbeat . . . 291
14.3 In-Memory File Systems . . . 292
14.4 Management Utilities and Commands . . . 293
14.5 OCFS2 Packages . . . 295
14.6 Creating an OCFS2 Volume . . . 295

14.7 Mounting an OCFS2 Volume . . . 300
14.8 Additional Information . . . 301

15 Access Control Lists in Linux 303

15.1 Traditional File Permissions . . . 303
15.2 Advantages of ACLs . . . 305
15.3 Definitions . . . 305
15.4 Handling ACLs . . . 306
15.5 ACL Support in Applications . . . 314
15.6 For More Information . . . 314

16 RPM—the Package Manager 315

16.1 Verifying Package Authenticity . . . 316
16.2 Managing Packages: Install, Update, and Uninstall . . . 316
16.3 RPM and Patches . . . 317
16.4 Delta RPM Packages . . . 319
16.5 RPM Queries . . . 320
16.6 Installing and Compiling Source Packages . . . 323
16.7 Compiling RPM Packages with build . . . 325
16.8 Tools for RPM Archives and the RPM Database . . . 326

17 System Monitoring Utilities 327

17.1 Debugging . . . 328
17.2 Files and File Systems . . . 330
17.3 Hardware Information . . . 332
17.4 Networking . . . 335
17.5 The `/proc` File System . . . 336
17.6 Processes . . . 339
17.7 System Information . . . 343
17.8 User Information . . . 347
17.9 Time and Date . . . 347

18 Working with the Shell 349

18.1 Getting Started with the Bash Shell . . . 350
18.2 Users and Access Permissions . . . 361
18.3 Important Linux Commands . . . 365
18.4 The vi Editor . . . 376

Part III System **381**

19 32-Bit and 64-Bit Applications in a 64-Bit System Environment **383**

19.1 Runtime Support 384
19.2 Software Development 385
19.3 Software Compilation on Biarch Platforms 386
19.4 Kernel Specifications 388

20 Booting and Configuring a Linux System **389**

20.1 The Linux Boot Process 389
20.2 The init Process 393
20.3 System Configuration via /etc/sysconfig 402

21 The Boot Loader **405**

21.1 Selecting a Boot Loader 406
21.2 Booting with GRUB 406
21.3 Configuring the Boot Loader with YaST 416
21.4 Uninstalling the Linux Boot Loader 420
21.5 Creating Boot CDs 420
21.6 The Graphical SUSE Screen 422
21.7 Troubleshooting 422
21.8 For More Information 424

22 Special System Features **425**

22.1 Information about Special Software Packages 425
22.2 Virtual Consoles 432
22.3 Keyboard Mapping 432
22.4 Language and Country-Specific Settings 433

23 Printer Operation **439**

23.1 The Workflow of the Printing System 441
23.2 Methods and Protocols for Connecting Printers 441
23.3 Installing the Software 442
23.4 Setting Up a Printer 443
23.5 Network Printers 447
23.6 Graphical Printing Interfaces 450
23.7 Printing from the Command Line 450
23.8 Special Features in SUSE Linux Enterprise 451
23.9 Troubleshooting 455

24 Dynamic Kernel Device Management with udev 463

24.1 The /dev Directory . . . 463
24.2 Kernel uevents and udev . . . 464
24.3 Drivers, Kernel Modules, and Devices . . . 464
24.4 Booting and Initial Device Setup . . . 465
24.5 Debugging udev Events . . . 465
24.6 Influencing Kernel Device Event Handling with udev Rules . . . 467
24.7 Persistent Device Naming . . . 467
24.8 The Replaced hotplug Package . . . 468
24.9 For More Information . . . 469

25 File Systems in Linux 471

25.1 Terminology . . . 471
25.2 Major File Systems in Linux . . . 472
25.3 Some Other Supported File Systems . . . 478
25.4 Large File Support in Linux . . . 479
25.5 For More Information . . . 480

26 The X Window System 483

26.1 Manually Configuring the X Window System . . . 483
26.2 Installing and Configuring Fonts . . . 490
26.3 For More Information . . . 496

27 Authentication with PAM 497

27.1 Structure of a PAM Configuration File . . . 498
27.2 The PAM Configuration of sshd . . . 499
27.3 Configuration of PAM Modules . . . 502
27.4 For More Information . . . 504

28 Power Management 507

28.1 Power Saving Functions . . . 508
28.2 APM . . . 509
28.3 ACPI . . . 511
28.4 Rest for the Hard Disk . . . 518
28.5 The powersave Package . . . 519
28.6 The YaST Power Management Module . . . 528

29 Wireless Communication 533

29.1 Wireless LAN . . . 533

Part IV Services 545

30 Basic Networking 547

30.1 IP Addresses and Routing . . . 551
30.2 IPv6—The Next Generation Internet . . . 554
30.3 Name Resolution . . . 563
30.4 Configuring a Network Connection with YaST . . . 565
30.5 Configuring VLAN Interfaces on SUSE Linux . . . 583
30.6 Managing Network Connections with NetworkManager . . . 584
30.7 Configuring a Network Connection Manually . . . 586
30.8 smpppd as Dial-up Assistant . . . 602

31 SLP Services in the Network 605

31.1 Activating SLP . . . 605
31.2 SLP Front-Ends in SUSE Linux Enterprise . . . 606
31.3 Installation over SLP . . . 606
31.4 Providing Services with SLP . . . 607
31.5 For More Information . . . 608

32 Time Synchronization with NTP 609

32.1 Configuring an NTP Client with YaST . . . 609
32.2 Configuring xntp in the Network . . . 613
32.3 Setting Up a Local Reference Clock . . . 613

33 The Domain Name System 615

33.1 DNS Terminology . . . 615
33.2 Configuration with YaST . . . 616
33.3 Starting the Name Server BIND . . . 626
33.4 The Configuration File /etc/named.conf . . . 628
33.5 Zone Files . . . 633
33.6 Dynamic Update of Zone Data . . . 637
33.7 Secure Transactions . . . 637
33.8 DNS Security . . . 639
33.9 For More Information . . . 639

34 DHCP 641

34.1 Configuring a DHCP Server with YaST . . . 642
34.2 DHCP Software Packages . . . 653
34.3 The DHCP Server dhcpd . . . 654
34.4 For More Information . . . 657

35 Using NIS **659**

35.1 Configuring NIS Servers 659
35.2 Configuring NIS Clients 665

36 LDAP—A Directory Service **667**

36.1 LDAP versus NIS 668
36.2 Structure of an LDAP Directory Tree 669
36.3 Server Configuration with slapd.conf 673
36.4 Data Handling in the LDAP Directory 679
36.5 Configuring an LDAP Server with YaST 683
36.6 Configuring an LDAP Client with YaST 688
36.7 Configuring LDAP Users and Groups in YaST 697
36.8 Browsing the LDAP Directory Tree 699
36.9 For More Information 700

37 Samba **703**

37.1 Terminology 703
37.2 Starting and Stopping Samba 705
37.3 Configuring a Samba Server 705
37.4 Configuring Clients 711
37.5 Samba as Login Server 712
37.6 Samba Server in the Network with Active Directory 713
37.7 Migrating a Windows NT Server to Samba 715
37.8 For More Information 717

38 Sharing File Systems with NFS **719**

38.1 Installing the Required Software 719
38.2 Importing File Systems with YaST 720
38.3 Importing File Systems Manually 721
38.4 Exporting File Systems with YaST 723
38.5 Exporting File Systems Manually 729
38.6 NFS with Kerberos 732
38.7 For More Information 732

39 File Synchronization **733**

39.1 Available Data Synchronization Software 733
39.2 Determining Factors for Selecting a Program 735
39.3 Introduction to CVS 738
39.4 Introduction to rsync 741

40 The Apache HTTP Server **745**

40.1 Quick Start . . . 745
40.2 Configuring Apache . . . 747
40.3 Starting and Stopping Apache . . . 761
40.4 Installing, Activating, and Configuring Modules . . . 763
40.5 Getting CGI Scripts to Work . . . 771
40.6 Setting Up a Secure Web Server with SSL . . . 773
40.7 Avoiding Security Problems . . . 779
40.8 Troubleshooting . . . 781
40.9 For More Information . . . 782

41 The Proxy Server Squid **785**

41.1 Some Facts about Proxy Caches . . . 786
41.2 System Requirements . . . 787
41.3 Starting Squid . . . 789
41.4 The Configuration File /etc/squid/squid.conf . . . 791
41.5 Configuring a Transparent Proxy . . . 797
41.6 cachemgr.cgi . . . 800
41.7 squidGuard . . . 802
41.8 Cache Report Generation with Calamaris . . . 803
41.9 For More Information . . . 804

Part V Security **805**

42 Managing X.509 Certification **807**

42.1 The Principles of Digital Certification . . . 807
42.2 YaST Modules for CA Management . . . 812

43 Masquerading and Firewalls **823**

43.1 Packet Filtering with iptables . . . 823
43.2 Masquerading Basics . . . 826
43.3 Firewalling Basics . . . 828
43.4 SuSEfirewall2 . . . 828
43.5 For More Information . . . 833

44 SSH: Secure Network Operations **835**

44.1 The OpenSSH Package . . . 835
44.2 The ssh Program . . . 836
44.3 scp—Secure Copy . . . 836
44.4 sftp—Secure File Transfer . . . 837

44.5 The SSH Daemon (sshd)—Server-Side 837
44.6 SSH Authentication Mechanisms 839
44.7 X, Authentication, and Forwarding Mechanisms 840

45 Network Authentication—Kerberos 841

45.1 Kerberos Terminology 841
45.2 How Kerberos Works 843
45.3 Users' View of Kerberos 846
45.4 For More Information 847

46 Installing and Administering Kerberos 849

46.1 Choosing the Kerberos Realms 849
46.2 Setting Up the KDC Hardware 850
46.3 Clock Synchronization 851
46.4 Configuring the KDC 851
46.5 Manually Configuring Kerberos Clients 854
46.6 Configuring a Kerberos Client with YaST 857
46.7 Remote Kerberos Administration 859
46.8 Creating Kerberos Host Principals 861
46.9 Enabling PAM Support for Kerberos 863
46.10 Configuring SSH for Kerberos Authentication 864
46.11 Using LDAP and Kerberos 865

47 Encrypting Partitions and Files 869

47.1 Setting Up an Encrypted File System with YaST 870
47.2 Using Encrypted Home Directories 873
47.3 Using vi to Encrypt Single ASCII Text Files 874

48 Confining Privileges with AppArmor 875

48.1 Installing Novell AppArmor 876
48.2 Enabling and Disabling Novell AppArmor 876
48.3 Getting Started with Profiling Applications 878

49 Security and Confidentiality 885

49.1 Local Security and Network Security 886
49.2 Some General Security Tips and Tricks 895
49.3 Using the Central Security Reporting Address 897

Part VI Troubleshooting 899

50 Help and Documentation 901

50.1 Using the SUSE Help Center . . . 901
50.2 Man Pages . . . 905
50.3 Info Pages . . . 906
50.4 The Linux Documentation Project . . . 906
50.5 Wikipedia: The Free Online Encyclopedia . . . 907
50.6 Guides and Books . . . 907
50.7 Package Documentation . . . 908
50.8 Usenet . . . 909
50.9 Standards and Specifications . . . 909

51 Common Problems and Their Solutions 913

51.1 Finding and Gathering Information . . . 913
51.2 Installation Problems . . . 916
51.3 Boot Problems . . . 924
51.4 Login Problems . . . 927
51.5 Network Problems . . . 933
51.6 Data Problems . . . 938
51.7 IBM System z: Using initrd as a Rescue System . . . 950

Index 955

About This Guide

This guide is intended for use by professional network and system administrators during the actual planning, deployment, configuration, and operation of SUSE Linux Enterprise®. As such, it is solely concerned with ensuring that SUSE Linux Enterprise is properly configured and that the required services on the network are available to allow it to function properly as initially installed. This guide does not cover the process of ensuring that SUSE Linux Enterprise offers proper compatibility with your enterprise's application software or that its core functionality meets those requirements. It assumes that a full requirements audit has been done and the installation has been requested or that a test installation, for the purpose of such an audit, has been requested.

This guide contains the following:

Deployment
: Before you install SUSE Linux Enterprise, choose the deployment strategy and disk setup that is best suited for your scenario. Learn how to install your system manually, how to use network installation setups, and how to perform an autoinstallation. Configure the installed system with YaST to adapt it to your requirements.

Administration
: SUSE Linux Enterprise offers a wide range of tools to customize various aspects of the system. This part introduces a few of them.

System
: Learn more about the underlying operating system by studying this part. SUSE Linux Enterprise supports a number of hardware architectures and you can use this to adapt your own applications to run on SUSE Linux Enterprise. The boot loader and boot procedure information assists you in understanding how your Linux system works and how your own custom scripts and applications may blend in with it.

Services
: SUSE Linux Enterprise is designed to be a network operating system. It offers a wide range of network services, such as DNS, DHCP, Web, proxy, and authentication services, and integrates well into heterogeneous environments including MS Windows clients and servers.

Security
: This edition of SUSE Linux Enterprise includes several security-related features. It ships with Novell® AppArmor, which enables you to protect your applications by restricting privileges. Secure login, firewalling, and file system encryption are covered as well.

Troubleshooting
: SUSE Linux Enterprise includes a wealth of applications, tools, and documentation should you need them in case of trouble. Some of the most common problems that can occur with SUSE Linux Enterprise and their solutions are discussed in detail.

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 Documentation Updates

For the latest version of this documentation, see the SUSE Linux Enterprise Server Web site [http://www.novell.com/documentation/sles10/index.html].

3 Additional Documentation

For additional documentation on this product, refer to http://www.novell.com/documentation/sles10/index.html:

Start-Up Guide
: Basic information about installation types and work flows.

Architecture-Specific Information
: Architecture-specific information needed to prepare a SUSE Linux Enterprise Server target for installation.

Novell AppArmor Administration Guide
: An in-depth administration guide to Novell AppArmor that introduces application confinement for heightened security in your environment.

Storage Administration Guide
: An introduction to managing various types of storage devices on SUSE Linux Enterprise.

Heartbeat Guide
: An in-depth administration guide to setting up high availability scenarios with Heartbeat.

Novell Virtualization Technology User Guide
: An introduction to virtualization solutions based on SUSE Linux Enterprise and the Xen* virtualization technology.

For a documentation overview on the SUSE® Linux Enterprise Desktop product, refer to `http://www.novell.com/documentation/sled10/index.html`. The following manuals are exclusively available for SUSE Linux Enterprise Desktop:

GNOME User Guide
: A comprehensive guide to the GNOME desktop and its most important applications.

KDE User Guide
: A comprehensive guide to the KDE desktop and its most important applications.

Deployment Guide
: An in-depth guide for administrators facing the deployment and management of SUSE Linux Enterprise Desktop.

Novell AppArmor Administration Guide
: An in-depth administration guide to Novell AppArmor that introduces application confinement for heightened security in your environment.

Many chapters in this manual contain links to additional documentation resources. This includes additional documentation that is available on the system as well as documentation available on the Internet.

4 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
- ► **amd64 ipf:** This paragraph is only relevant for the specified architectures. The arrows mark the beginning and the end of the text block. ◄

  ► **ipseries s390 zseries:** This paragraph is only relevant for the specified architectures. The arrows mark the beginning and the end of the text block. ◄
- *Dancing Penguins* (Chapter *Penguins*, ↑Another Manual): This is a reference to a chapter in another manual.

Part I. Deployment

Planning for SUSE Linux Enterprise

1

The implementation of an operating system either in an existing IT environment or as a completely new rollout must be carefully prepared. With SUSE Linux Enterprise 10, get a variety of new features. It is impossible to describe all the new features here. The following is just a list of major enhancements that might be of interest.

Xen 3.0 Virtualization
: Runs many virtual machines on a single server, each with its own instance of an operating system. For more information about this technology, see the virtualization manual on http://www.novell.com/documentation/sles10/index.html.

YaST
: Several new configuration options have been developed for YaST. These are normally described in the chapters about the technology involved.

CIM Management with openWBEM
: The Common Information Model Object Manager (CIMON) is a Web-based enterprise management utility. It provides a mature management framework. See also Chapter 11, *OpenWBEM* (page 241).

SPident
: The management utility SPident gives an overview of the installed software base and clarifies the current service pack level of the system.

Directory Services
: Several LDAP-compliant directory services are available:

- Microsoft Active Directory
- OpenLDAP

Novell AppArmor
: Harden your System with the Novell AppArmor technology. This service is described in depth in *Novell AppArmor Administration Guide* (↑*Novell AppArmor Administration Guide*).

iSCSI
: iSCSI provides an easy and reasonably inexpensive solution for connecting Linux computers to central storage systems. Find more information about iSCSI in Chapter 12, *Mass Storage over IP Networks—iSCSI* (page 271).

Network File System v4
: Starting with version 10, SUSE Linux Enterprise supports NFS also in version 4. This gives you performance improvements, strong security, and a "stateful" protocol. See also Chapter 38, *Sharing File Systems with NFS* (page 719).

Oracle Cluster File System 2
: OCFS2 is a general-purpose journaling file system that is fully integrated in the Linux 2.6 kernel and later. Find an overview of OCFS2 in Chapter 14, *Oracle Cluster File System 2* (page 289).

Heartbeat 2
: Heartbeat 2 provides a cluster membership and messaging infrastructure. The setup of such a cluster is described in the *Heartbeat Guide*.

Multipath I/O
: Device mapping multipath IO features automatic configuration of the subsystem for a large variety of setups. For details, see the chapter about multipath I/O in *Storage Administration Guide*.

Linux Kernel Crash Dump
: Debugging kernel-related problems is now much more comfortable when using Kexec and Kdump. This technology is available on x86, AMD64, Intel 64, and POWER platforms.

1.1 Considerations for Deployment of a SUSE Linux Enterprise

At the beginning of the planning process, you should try to define the project goals and needed features. This must be done individually for each project, but the questions to answer should include the following:

- How many installations should be done? Depending on this, the best deployment method differs. See also Chapter 2, *Deployment Strategies* (page 7).
- Will the system be in a hostile environment? Have a look at Chapter 49, *Security and Confidentiality* (page 885) to get an overview of consequences.
- How will you get regular updates? All patches are provided online for registered users. Find the registration and patch support database at `http://support.novell.com/patches.html`.
- Do you need help for your local installation? Novell provides training, support, and consulting for all topics around SUSE Linux Enterprise. Find more information about this at `http://www.novell.com/products/server/`.
- Do you need third-party products? Make sure that the required product is also supported on the desired platform. Novell can also provide help to port software to different platforms when needed.

1.2 Deployment of SUSE Linux Enterprise

To make sure that your system will run flawlessly, always try to use certified hardware. The hardware certification process is an ongoing process and the database of certified hardware is updated regularly. Find the search form for certified hardware at `http://developer.novell.com/yessearch/Search.jsp`.

Depending on the number of desired installations, it is beneficial to use installation servers or even completely automatic installations. Have a look at Chapter 2, *Deployment Strategies* (page 7) for more information. When using the Xen virtualization technolo-

gies, network root file systems or network storage solutions like iSCSI should be considered. See also Chapter 12, *Mass Storage over IP Networks—iSCSI* (page 271).

SUSE Linux Enterprise provides you with a broad variety of services. Find an overview of the documentation in this book in About This Guide (page xv). Most of the needed configurations can be made with YaST, the SUSE configuration utility. In addition to that, many manual configurations are described in the corresponding chapters.

In addition to the plain software installation, you should consider training the end users of the systems as well as help desk staff.

1.3 Running SUSE Linux Enterprise

The SUSE Linux Enterprise operating system is a well-tested and stable system. Unfortunately, this does not prevent hardware failures or other causes for downtime or data loss. For any serious computing task where data loss could occur, a regular backup should be done.

For optimal security and safe work, you should make regular updates of all the operated machines. If you have a mission critical server, you should probably run a second identical machine where you can apply all changes for testing purposes before doing so on the real system. This also gives you the possibility to switch machines in case of hardware failure.

2 Deployment Strategies

There are several different ways to deploy SUSE® Linux Enterprise. Choose from various approaches ranging from a local installation using physical media or a network installation server to a mass deployment using a remote-controlled, highly-customized, and automated installation technique. Select the method that best matches your requirements.

2.1 Deploying up to 10 Workstations

If your deployment of SUSE Linux Enterprise only involves 1 to 10 workstations, the easiest and least complex way of deploying SUSE Linux Enterprise is a plain manual installation as featured in Chapter 3, *Installation with YaST* (page 17). Manual installation can be done in several different ways depending on your requirements:

Installing from the SUSE Linux Enterprise Media (page 8)
: Consider this approach if you want to install a single, disconnected workstation.

Installing from a Network Server Using SLP (page 8)
: Consider this approach if you have a single workstation or a small number of workstations and if a network installation server announced via SLP is available.

Installing from a Network Server (page 9)
: Consider this approach if you have a single workstation or a small number of workstations and if a network installation server is available.

Table 2.1 *Installing from the SUSE Linux Enterprise Media*

Installation Source	SUSE Linux Enterprise media kit
Tasks Requiring Manual Interaction	• Inserting the installation media • Booting the installation target • Changing media • Determining the YaST installation scope • Configuring the system with YaST system
Remotely Controlled Tasks	None
Details	Section 3.3.2, “Installing from the SUSE Linux Enterprise Media” (page 19)

Table 2.2 *Installing from a Network Server Using SLP*

Installation Source	Network installation server holding the SUSE Linux Enterprise installation media
Tasks Requiring Manual Interaction	• Inserting the boot disk • Booting installation target • Determining the YaST installation scope • Configuring the system with YaST
Remotely Controlled Tasks	None, but this method can be combined with VNC
Details	Section 3.3.3, “Installing from a Network Server Using SLP” (page 19)

Table 2.3 *Installing from a Network Server*

Installation Source	Network installation server holding the SUSE Linux Enterprise installation media
Tasks Requiring Manual Interaction	• Inserting the boot disk • Providing boot options • Booting the installation target • Determining the YaST installation scope • Configuring the system with YaST
Remotely Controlled Tasks	None, but method can be combined with VNC
Details	Section 3.3.4, “Installing from a Network Source without SLP” (page 20)

2.2 Deploying up to 100 Workstations

With a growing numbers of workstations to install, you certainly do not want to install and configure each one of them manually. There are many automated or semiautomated approaches as well as several options to perform an installation with minimal to no physical user interaction.

Before considering a fully-automated approach, take into account that the more complex the scenario gets the longer it takes to set up. If a time limit is associated with your deployment, it might be a good idea to select a less complex approach that can be carried out much more quickly. Automation makes sense for huge deployments and those that need to be carried out remotely.

Choose from the following options:

Simple Remote Installation via VNC—Static Network Configuration (page 11)
: Consider this approach in a small to medium scenario with a static network setup. A network, network installation server, and VNC viewer application are required.

Simple Remote Installation via VNC—Dynamic Network Configuration (page 11)
Consider this approach in a small to medium scenario with dynamic network setup through DHCP. A network, network installation server, and VNC viewer application are required.

Remote Installation via VNC—PXE Boot and Wake on LAN (page 12)
Consider this approach in a small to medium scenario that should be installed via network and without physical interaction with the installation targets. A network, a network installation server, network boot images, network bootable target hardware, and a VNC viewer application are required.

Simple Remote Installation via SSH—Static Network Configuration (page 12)
Consider this approach in a small to medium scenario with static network setup. A network, network installation server, and SSH client application are required.

Remote Installation via SSH—Dynamic Network Configuration (page 13)
Consider this approach in a small to medium scenario with dynamic network setup through DHCP. A network, network installation server, and SSH client application are required.

Remote Installation via SSH—PXE Boot and Wake on LAN (page 13)
Consider this approach in a small to medium scenario that should be installed via network and without physical interaction with the installation targets. A network, a network installation server, network boot images, network bootable target hardware, and an SSH client application are required.

Simple Mass Installation (page 14)
Consider this approach for large deployments to identical machines. If configured to use network booting, physical interaction with the target systems is not needed at all. A network, a network installation server, a remote controlling application such as a VNC viewer or an SSH client, and an AutoYaST configuration profile are required. If using network boot, a network boot image and network bootable hardware are required as well.

Rule-Based Autoinstallation (page 15)
Consider this approach for large deployments to various types of hardware. If configured to use network booting, physical interaction with the target systems is not needed at all. A network, a network installation server, a remote controlling application such as a VNC viewer or an SSH client, and several AutoYaST configuration profiles as well as a rule setup for AutoYaST are required. If using network boot, a network boot image and network bootable hardware are required as well.

Table 2.4 *Simple Remote Installation via VNC—Static Network Configuration*

Installation Source	Network
Preparations	• Setting up an installation source • Booting from the installation media
Control and Monitoring	Remote: VNC
Best Suited For	small to medium scenarios with varying hardware
Drawbacks	• Each machine must be set up individually • Physical access is needed for booting
Details	Section 4.1.1, “Simple Remote Installation via VNC—Static Network Configuration” (page 48)

Table 2.5 *Simple Remote Installation via VNC—Dynamic Network Configuration*

Installation Source	Network
Preparations	• Setting up the installation source • Booting from the installation media
Control and Monitoring	Remote: VNC
Best Suited For	Small to medium scenarios with varying hardware
Drawbacks	• Each machine must be set up individually • Physical access is needed for booting

Details	Section 4.1.2, “Simple Remote Installation via VNC—Dynamic Network Configuration” (page 49)

Table 2.6 *Remote Installation via VNC—PXE Boot and Wake on LAN*

Installation Source	Network
Preparations	• Setting up the installation source • Configuring DHCP, TFTP, PXE boot, and WOL • Booting from the network
Control and Monitoring	Remote: VNC
Best Suited For	• Small to medium scenarios with varying hardware • Completely remote installs; cross-site deployment
Drawbacks	Each machine must be set up manually
Details	Section 4.1.3, “Remote Installation via VNC—PXE Boot and Wake on LAN” (page 51)

Table 2.7 *Simple Remote Installation via SSH—Static Network Configuration*

Installation Source	Network
Preparations	• Setting up the installation source • Booting from the installation media
Control and Monitoring	Remote: SSH
Best Suited For	• Small to medium scenarios with varying hardware

	• Low bandwidth connections to target
Drawbacks	• Each machine must be set up individually • Physical access is needed for booting
Details	Section 4.1.4, “Simple Remote Installation via SSH—Static Network Configuration” (page 52)

Table 2.8 *Remote Installation via SSH—Dynamic Network Configuration*

Installation Source	Network
Preparations	• Setting up the installation source • Booting from installation media
Control and Monitoring	Remote: SSH
Best Suited For	• Small to medium scenarios with varying hardware • Low bandwidth connections to target
Drawbacks	• Each machine must be set up individually • Physical access is needed for booting
Details	Section 4.1.5, “Simple Remote Installation via SSH—Dynamic Network Configuration” (page 53)

Table 2.9 *Remote Installation via SSH—PXE Boot and Wake on LAN*

Installation Source	Network
Preparations	• Setting up the installation source

	• Configuring DHCP, TFTP, PXE boot, and WOL • Booting from the network
Control and Monitoring	Remote: SSH
Best Suited For	• Small to medium scenarios with varying hardware • Completely remote installs; cross-site deployment • Low bandwidth connections to target
Drawbacks	Each machine must be set up individually
Details	Section 4.1.6, “Remote Installation via SSH—PXE Boot and Wake on LAN” (page 55)

Table 2.10 *Simple Mass Installation*

Installation Source	Preferably network
Preparations	• Gathering hardware information • Creating AutoYaST profile • Setting up the installation server • Distributing the profile • Setting up network boot (DHCP, TFTP, PXE, WOL) *or* Booting the target from installation media
Control and Monitoring	Local or remote through VNC or SSH

Best Suited For	• Large scenarios • Identical hardware • No access to system (network boot)
Drawbacks	Applies only to machines with identical hardware
Details	Section 5.1, “Simple Mass Installation” (page 85)

Table 2.11 *Rule-Based Autoinstallation*

Installation Source	Preferably network
Preparations	• Gathering hardware information • Creating AutoYaST profiles • Creating AutoYaST rules • Setting up the installation server • Distributing the profile • Setting up network boot (DHCP, TFTP, PXE, WOL) *or* Booting the target from installation media
Control and Monitoring	Local or remote through SSH or VNC
Best Suited For	• Varying hardware • Cross-site deployments
Drawbacks	Complex rule setup

2.3 Deploying More than 100 Workstations

Most of the considerations brought up for medium installation scenarios in Section 2.1, “Deploying up to 10 Workstations” (page 7) still hold true for large scale deployments. However, with a growing number of installation targets, the benefits of a fully automated installation method outweigh its disadvantages.

It pays off to invest a considerable amount of time to create a sophisticated rule and class framework in AutoYaST to match the requirements of a huge deployment site. Not having to touch each target separately can save you a tremendous amount of time depending on the scope of your installation project.

3 Installation with YaST

After your hardware has been prepared for the installation of SUSE Linux Enterprise® as described in the *Architecture-Specific Information* manual and after the connection with the installation system has been established, you are presented with the interface of SUSE Linux Enterprise's system assistant YaST. YaST guides you through the entire installation and configuration procedure.

3.1 IBM POWER: System Start-Up for Network Installation

For IBM POWER platforms, the system is initialized (IPL) as described in the *Architecture-Specific Information* manual. For a network installation, SUSE Linux Enterprise Server does not show a splash screen or boot loader command line on these systems. During the installation, load the kernel manually. YaST starts with its installation screen as soon as a connection has been established to the installation system via VNC, X, or SSH. Because there is no splash screen or boot loader command line, kernel or boot parameters cannot be entered on screen, but must be included in the kernel image using the `mkzimage_cmdline` utility. See the Preparation chapter in the *Architecture-Specific Information* manual for a description.

TIP: IBM POWER: The Next Steps

To install, follow the description of the installation procedure with YaST starting from Section 3.6, “Language” (page 24).

3.2 IBM System z: System Start-Up for Installation

For IBM System z platforms, the system is initialized (IPL) as described in the *Architecture-Specific Information* manual. SUSE Linux Enterprise does not show a splash screen on these systems. During the installation, load the kernel, initrd, and parmfile manually. YaST starts with its installation screen as soon as a connection has been established to the installation system via VNC, X, or SSH. Because there is no splash screen, kernel or boot parameters cannot be entered on screen, but must be specified in a parmfile (see the parmfile information in Appendix A, *Appendix* (↑*Architecture-Specific Information*)).

TIP: IBM System z: The Next Steps

To install, follow the description of the installation procedure with YaST starting from Section 3.6, “Language” (page 24).

3.3 System Start-Up for Installation

You can install SUSE Linux Enterprise from local installation sources, such as the SUSE Linux Enterprise CDs or DVD, or from the network source of an FTP, HTTP, SLP, or NFS server. Any of these approaches requires physical access to the system to install and user interaction during the installation. The installation procedure is basically the same regardless of the installation source.

3.3.1 Boot Options

Boot options other than CD or DVD exist and can be used if problems arise booting from CD or DVD. These options are described in Table 3.1, “Boot Options” (page 19).

Table 3.1 *Boot Options*

Boot Option	Description
DVD/CD-ROM	This is the easiest boot option. This option can be used if the system has a local CD/DVD-ROM drive that is supported by Linux.
Floppy	The images for generating boot floppies are located on CD/DVD 1 in the `/boot` directory. A `README` is available in the same directory.
PXE or BOOTP	This must be supported by the system's BIOS or firmware and a boot server must be available in the network. This task can also be handled by another SUSE Linux Enterprise system.
Hard Disk	SUSE Linux Enterprise can also be booted from the hard disk. To do this, copy the kernel (`linux`) and the installation system (`initrd`) from the directory `/boot/loader` on CD/DVD 1 to the hard disk and add the appropriate entry to the boot loader.

3.3.2 Installing from the SUSE Linux Enterprise Media

To install from the media, insert the first CD or DVD into the appropriate drive of the system to install. Reboot the system to boot from the media and open the boot screen.

3.3.3 Installing from a Network Server Using SLP

If your network setup supports OpenSLP and your network installation source has been configured to announce itself via OpenSLP (described in Section 4.2, “Setting Up the Server Holding the Installation Sources” (page 56)), boot the system from the media or with another boot option. In the boot screen, select the desired installation option. Press F4 then select *SLP*.

The installation program retrieves the location of the network installation source using OpenSLP and configures the network connection with DHCP. If the DHCP network configuration fails, you are prompted to enter the appropriate parameters manually. The installation then proceeds as described below.

3.3.4 Installing from a Network Source without SLP

If your network setup does not support OpenSLP for the retrieval of network installation sources, boot the system from the media or with another boot option. In the boot screen, select the desired installation option. Press F4 then select the desired network protocol (NFS, HTTP, FTP, or SMB). Provide the server's address and the path to the installation media.

The installation program retrieves the location of the network installation source using OpenSLP and configures the network connection with DHCP. If the DHCP network configuration fails, you are prompted to enter the appropriate parameters manually. The installation then proceeds as described below.

3.4 The Installation Workflow

The SUSE Linux Enterprise installation is split into three main parts: preparation, installation, configuration. During the preparation phase you configure some basic parameters such as language, time, and desktop type. In the installation phase you decide which software to install, where to install it and how to boot the installed system. Upon finishing the installation the machine reboots into the newly installed system and starts the configuration. In this stage you set up users and passwords, and configure network and Internet access as well as hardware components such as printers.

3.5 The Boot Screen

The boot screen displays a number of options for the installation procedure. *Boot from Hard Disk* boots the installed system and is selected default, because the CD/DVD is often left in the drive. To install the system, select one of the installation options with the arrow keys. The relevant options are:

Installation
: The normal installation mode. All modern hardware functions are enabled. All modern hardware functions are enabled.

Installation—ACPI Disabled
: If the normal installation fails, this might be due to the system hardware not supporting ACPI (advanced configuration and power interface). If this seems to be the case, use this option to install without ACPI support.

Installation—Local APIC Disabled
: If the normal installation fails, this might be due to the system hardware not supporting local APIC (Advanced Programmable Interrupt Controllers). If this seems to be the case, use this option to install without local APIC support.

  If you are not sure, try one of the following options first: *Installation—ACPI Disabled* or *Installation—Safe Settings*.

Installation—Safe Settings
: Boots the system with the DMA mode (for CD-ROM drives) and power management functions disabled.

Rescue System
: Starts a minimal Linux system without a graphical user interface. For more information, see Section “Using the Rescue System” (page 946).

Memory Test
: Tests your system RAM using repeated read and write cycles. Terminate the test by rebooting. For more information, see Section 51.2.5, “Fails to Boot” (page 920).

Installation options from the menu disable only the most problematic functions. If you need to disable or set other functions, use the *Boot Options* prompt. Find detailed information about kernel parameters at `http://en.opensuse.org/Linuxrc`.

Use the function keys indicated in the bar at the bottom of the screen to change the language, resolution of the monitor, or installation source or to add an additional driver from your hardware vendor:

F1 *Help*
: Get context-sensitive help for the active element of the boot screen.

F2 *Language*
: Select the display language for the installation. The default language is English.

F3 *Video Mode*
: Select various graphical display modes for the installation. Select *Text Mode* if the graphical installation causes problems.

F4 *Source*
: Normally, the installation is performed from the inserted installation medium. Here, select other sources, like FTP or NFS servers. If the installation is carried out in a network with an SLP server, select one of the installation sources available on the server with this option. Find information about SLP in Chapter 31, *SLP Services in the Network* (page 605).

F5 *Driver*
: Press this key to tell the system that you have an optional disk with a driver update for SUSE Linux Enterprise. With *File*, load drivers directly from CD before the installation starts. If you select *Yes*, you are prompted to insert the update disk at the appropriate point in the installation process. The default option is *No*—not to load a driver update.

After starting the installation, SUSE Linux Enterprise loads and configures a minimal Linux system to run the installation procedure. To view the boot messages and copyright notices during this process, press Esc. On completion of this process, the YaST installation program starts and displays the graphical installer.

TIP: Installation without a Mouse

If the installer does not detect your mouse correctly, use Tab for navigation, arrow keys to scroll, and Enter to confirm a selection.

3.5.1 Providing Data to Access a SMT Server

If your network provides a SMT server to provide a local update source, you need to equip the client with the server's URL. Client and server communicate solely via HTTPS protocol, therefore you also need to enter a path to the server's certificate if the certificate was not issued by a certificate authority. This information has to be entered at the boot prompt.

smturl
: URL of the SMT server. The URL has a fixed format `https://FQN/center/regsvc/` *FQN* has to be full qualified hostname of the SMT server. Example:

```
smturl=https://smt.example.com/center/regsvc/
```

smtcert
: Location of the SMT server's certificate. Specify one of the following locations:

  URL
  : Remote location (http, https or ftp) from which the certificate can be downloaded. Example:

    ```
    smtcert=http://smt.example.com/smt-ca.crt
    ```

  Floppy
  : Specifies a location on a floppy. The floppy has to be inserted at boot time, you will not be prompted to insert it if it is missing. The value has to start with the string `floppy` followed by the path to the certificate. Example:

    ```
    smtcert=floppy/smt/smt-ca.crt
    ```

  local path
  : Absolute path to the certificate on the local machine. Example:

    ```
    smtcert=/data/inst/smt/smt-ca.cert
    ```

  Interactive
  : Use `ask` to open a pop-up menu during the installation where you can specify the path to the certificate. Do not use this option with AutoYaST. Example

    ```
    smtcert=ask
    ```

  Deactivate certificate installation
  : Use `done` if either the certificate will be installed by an add-on product, or if you are using a certificate issued by an official certificate authority. Example:

    ```
    smtcert=done
    ```

WARNING: Beware of typing errors

Make sure the values you enter are correct. If `smturl` has not been specified correctly, the registration of the update source will fail. If a wrong value for

smtcert has been entered, you will be prompted for a local path to the certificate.

In case smtcert is not specified, it will default to `http://FQN/smt.crt` with `FQN` being the name of the SMT server.

3.6 Language

YaST and SUSE Linux Enterprise in general can be configured to use different languages according to your needs. The language selected here is also used for the keyboard layout. In addition, YaST uses the language setting to guess a time zone for the system clock. These settings can be modified later along with the selection of secondary languages to install on your system.

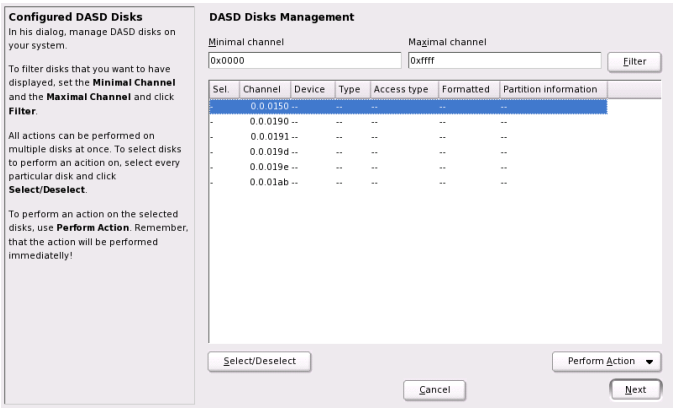
You can change the language later during installation as described in Section 3.12, "Installation Settings" (page 29). For information about language settings in the installed system, see Section 8.1, "YaST Language" (page 130).

3.7 IBM System z: Hard Disk Configuration

When installing on IBM System z platforms, the language selection dialog is followed by a dialog to configure the attached hard disks. Select DASD, Fibre Channel Attached SCSI Disks (zFCP), or iSCSI for installation of SUSE Linux Enterprise.

After selecting *Configure DASD Disks*, an overview lists all available DASDs. To get a clearer picture of the available devices, use the entry field located above the list to specify a range of channels to display. To filter the list according to such a range, select *Filter*. See Figure 3.1, "IBM System z: Selecting a DASD" (page 25).

Figure 3.1 *IBM System z: Selecting a DASD*



Now specify the DASDs to use for the installation by selecting the corresponding entries in the list then clicking *Select or Deselect*. After that, activate and make the DASDs available for the installation by selecting *Perform Action* > *Activate*. See Figure 3.2, “IBM System z: Activating a DASD” (page 25). To format the DASDs, select *Perform Action* > *Format* right away or use the YaST partitioner later as described in Section 8.5.7, “Using the YaST Partitioner” (page 155).

Figure 3.2 *IBM System z: Activating a DASD*

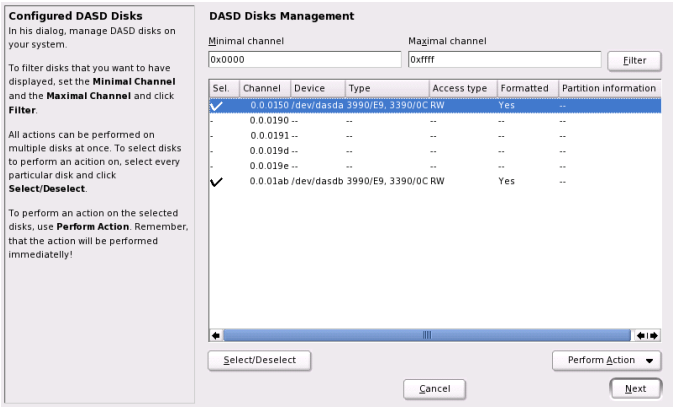
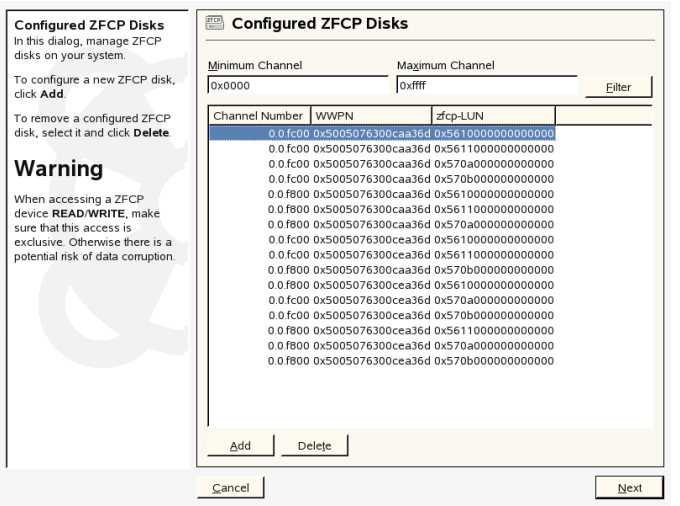


Figure 3.3 *IBM System z: Overview of Available zFCP Disks*



To use zFCP disks for the SUSE Linux Enterprise installation, select *Configure zFCP Disks* in the selection dialog. This opens a dialog with a list of the zFCP disks available on the system. In this dialog, select *Add* to open another dialog in which to enter zFCP parameters. See Figure 3.3, “IBM System z: Overview of Available zFCP Disks” (page 26).

To make a zFCP disk available for the SUSE Linux Enterprise installation, choose an available *Channel Number* from the drop-down list. *Get WWPNs* (World Wide Port Number) and *Get LUNs* (Logical Unit Number) return lists with available WWPNs and FCP-LUNs, respectively, to choose from. When completed, exit the zFCP dialog with *Next* and the general hard disk configuration dialog with *Finish* to continue with the rest of the configuration.

TIP: Adding DASD or zFCP Disks at a Later Stage

Adding DASD or zFCP disks is not only possible during the installation workflow, but also when the installation proposal is shown. To add disks at that stage, click *Expert* and scroll down. The DASD and zFCP entries are shown at the very bottom.

After adding the disks, reread the partition table. Return to the installation proposal screen and choose *Partitioning* then select *Reread Partition Table*. This reads the new partition table and resets any previously entered information.

3.8 Media Check

The media check dialog appears only if you install from media created from downloaded ISOs. If you install from the original media set, the dialog is skipped.

The media check examines the integrity of a medium. To start the media check, select the drive in that contains the installation medium and click *Start Check*. The check can take some time.

To test multiple media, wait until a result message appears in the dialog before changing the medium. If the last medium checked is not the one with which you started the installation, YaST prompts for the appropriate medium before continuing with the installation.

WARNING: Failure of Media Check

If the media check fails, your medium is damaged. Do not continue the installation because installation may fail or you may lose your data. Replace the broken medium and restart the installation process.

If the result of the media check is positive, click *Next* to continue the installation.

3.9 License Agreement

Read the license agreement that is displayed on screen thoroughly. If you agree to the terms, choose *Yes, I Agree to the License Agreement* and click *Next* to confirm your selection. If you do not agree to the license agreement, you cannot install SUSE Linux Enterprise and the installation terminates.

3.10 Installation Mode

After a system analysis where YaST tries to find other installed systems or an already existing SUSE Linux Enterprise system on your machine, YaST displays the installation modes available:

New installation
: Select this option to start a new installation from scratch.

Update an existing system
: Select this option to update to a newer version. For more information about system update, see Chapter 10, *Updating SUSE Linux Enterprise* (page 211).

Other Options
: This option provides an opportunity to abort installation and boot or repair an installed system instead. To boot an already installed SUSE Linux Enterprise, select *Boot Installed System*. If you have problems booting an already installed SUSE Linux Enterprise, see Section 51.3, “Boot Problems” (page 924).

  To repair an installed system that fails to boot, select *Repair Installed System*. Find a description of the system repair options in Section “Using YaST System Repair” (page 941).

NOTE: Updating an Installed System

Updating is only possible if an older SUSE Linux Enterprise system is already installed. If no SUSE Linux Enterprise system is installed, you can only perform a new installation.

You can choose to install add-on products together with your SUSE Linux Enterprise system during the initial installation process or at any time later as described inSection 8.3.2, “Installing Add-On Products” (page 139). Add-on products are extensions for your SUSE Linux Enterprise. An add-on product can include proprietary third-party products or additional software for your system.

To include add-on products during the installation of SUSE Linux Enterprise, select *Include Add-On Products from Separate Media* and click *Next*. In the next dialog, click *Add* to select the source from which to install the add-on products. Many source types

are available, such as CD, FTP, or a local directory. After adding the add-on media, you may need to agree to additional license agreements for third-party products.

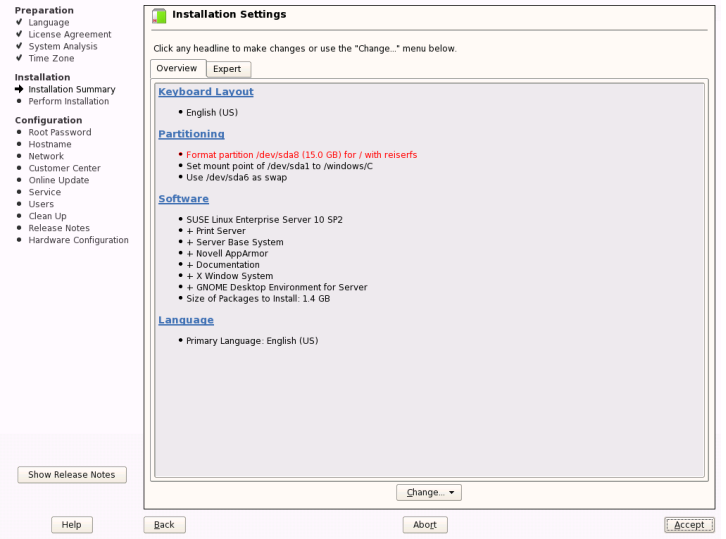
3.11 Clock and Time Zone

In this dialog, select your region and time zone from the lists. During installation, both are preselected according to the selected installation language. Choose between *Local Time* and *UTC* (GMT) for *Hardware Clock Set To*. The selection depends on how the BIOS hardware clock is set on your machine. If it is set to GMT, which corresponds to UTC, your system can rely on SUSE Linux Enterprise to switch from standard time to daylight saving time and back automatically. Click *Change* to set the current date and time. When finished, click *Next* to continue the installation.

3.12 Installation Settings

After a thorough system analysis, YaST presents reasonable suggestions for all installation settings. Basic settings can be changed in the *Overview* tab, advanced options are available on the *Experts* tab. To modify the suggestions, either click *Change* and select the category to change or click on one of the headlines. After configuring any of the items presented in these dialogs, you are always returned to the summary window, which is updated accordingly.

Figure 3.4 *Installation Settings*



TIP: Resetting the changes to default values

You can reset all changes to the defaults by clicking *Change* > *Reset to Defaults*. YaST then shows the original proposal again.

3.12.1 Overview

The options that sometimes need manual intervention in common installation situations are presented in the *Overview* tab. Modify Partitioning, Software selection and Locale settings here.

Keyboard Layout

To change the keyboard layout, select *Keyboard Layout*. By default, the layout corresponds to the language chosen for installation. Select a keyboard layout from the list. Use the *Test* field at the bottom of the dialog to check if you can enter special characters of that layout correctly. Find more information about changing the keyboard layout in Section 8.4.10, “Keyboard Layout” (page 150). When finished, click *Accept* to return to the installation summary.

► **zseries:** On the IBM System z platforms, the installation is performed from a remote terminal. The host as such has no keyboard or mouse locally connected to it. ◄

Partitioning

In most cases, YaST proposes a reasonable partitioning scheme that can be accepted without change. YaST can also be used to customize the partitioning, but only experienced users should change partitioning.

When you select the *Partitioning* for the first time, the YaST partitioning dialog displays the proposed partition settings. To accept these settings, click *Accept Proposal*.

To make small changes in the proposal, select *Base Partition Setup on This Proposal* and adjust partitioning in the next dialog. For a completely different partitioning, select *Create Custom Partition Setup*. In the next dialog, choose a specific disk to partition or *Custom Partitioning* if you want to have access to all disks. For more information about custom partitioning, refer to Section 8.5.7, “Using the YaST Partitioner” (page 155)the SUSE Linux Enterprise Server documentation. The YaST partitioner also provides a tool for LVM creation. To create an LVM proposal, select *Create LVM Based Proposal*. See Section 7.1, “LVM Configuration” (page 115) for more information on LVM.

NOTE: Using Minidisks in z/VM

If SUSE Linux Enterprise Server is installed on minidisks in z/VM, which reside on the same physical disk, the access path of the minidisks (/dev/disk/by-id/) is not unique but rather the ID of the physical disk. So if two or more minidisks are on the same physical disk, they all have the same ID.

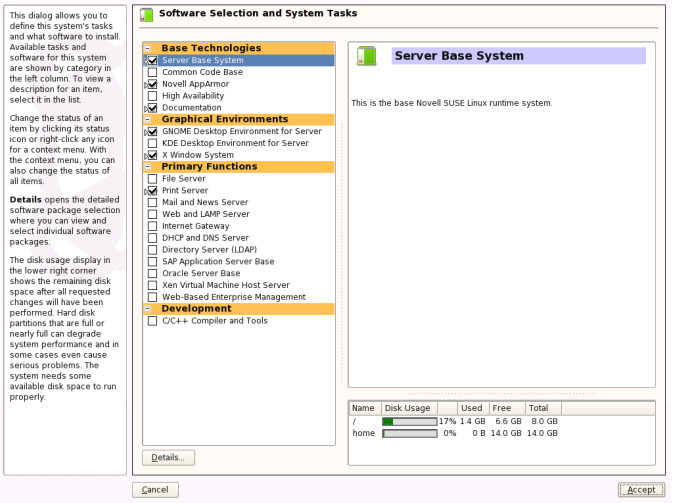
To avoid problems when mounting the minidisks, always mount them either "by path" or "by UUID".

Software

SUSE Linux Enterprise contains a number of software packages for various application purposes. Click *Software* in the suggestion window to start the software selection and modify the installation scope according to your needs. Select your pattern from the list in the middle and see the description in the right part of the window. Each pattern contains a number of software packages needed for specific functions (e.g. Multimedia

or Office software). For a more detailed selection based on software packages to install, select *Details* to switch to the YaST Software Manager. See Figure 3.5, “Installing and Removing Software with the YaST Software Manager” (page 32).

Figure 3.5 *Installing and Removing Software with the YaST Software Manager*



You can also install additional software packages or remove software packages from your system at any time later. For more information, refer to Section 8.3.1, “Installing and Removing Software” (page 132).

NOTE: Default Desktop

The default desktop of SUSE Linux Enterprise is GNOME. To install KDE, click *Software* and select *KDE Desktop Environment* from *Graphical Environments*.

Language

To change the system language or to configure support for secondary languages, select *Language*. Select a language from the list. The primary language is used as the system language. Choose a secondary languages to be able to switch to one of these languages at any time without having to install additional packages. For more information, see Section 8.5.15, “Language Selection” (page 163).

3.12.2 Expert

If you are an advanced user and want to configure booting or change the time zone or default runlevel, select the *Expert* tab. It shows the following additional entries not contained on the *Overview* tab:

System
: This dialog presents all the hardware information YaST could obtain about your computer. Select any item in the list and click *Details* to see detailed information about the selected item. Advanced users can also change the PCI ID setup and Kernel Settings by choosing *System Settings*.

Add-On Products
: The added source for add-on media appears in the overview. Before you start the installation of the SUSE Linux Enterprise, add, remove, or modify add-on products here if needed.

Booting

: ► **zseries:** This module cannot be used to configure the boot loader (zipl) on the IBM System z platforms. ◄

: YaST proposes a boot configuration for your system. Normally, you can leave these settings unchanged. However, if you need a custom setup, modify the proposal for your system. For information, see Section 21.3, “Configuring the Boot Loader with YaST” (page 416).

Time Zone
: This is the same as the configuration shown earlier in Section 3.11, “Clock and Time Zone” (page 29).

Default Runlevel
: SUSE Linux Enterprise can boot to different runlevels. Normally there should be no need to change anything here, but if necessary, set the default runlevel with this dialog. Refer to Section 20.2.3, “Configuring System Services (Runlevel) with YaST” (page 400) for information about runlevel configuration.

3.13 Performing the Installation

After making all installation settings, click *Accept* in the suggestion window to begin the installation. Confirm with *Install*. Some software may require a license confirmation. If your software selection includes such software, license confirmation dialogs are displayed. Click *Accept* to install the software. When not agreeing to the license, click *I Disagree* and the software will not be installed.

The installation usually takes between 15 and 30 minutes, depending on the system performance and the software selected. During this procedure a slide show introduces the features of SUSE Linux Enterprise. Choose *Details* to switch to the installation log. As soon as all packages are installed, YaST boots into the new Linux system, after which you can configure the hardware and set up system services.

3.13.1 IBM System z: IPLing the Installed System

In most cases, YaST automatically reboots into the installed system on the IBM System z platform. Known exceptions to this are installations wherein the bootloader resides on an FCP device in environments with LPAR on a machine older than z9 or with z/VM older than release 5.3. The bootloader gets written to the device that holds the `/boot` directory. If `/boot` is not on a separate partition, it is on the same partition as the root file system `/`.

In cases where an automatic reboot is not possible, YaST will show a dialog box containing information about from which device to do an IPL. Accept the shutdown option and perform an IPL after the shutdown. The procedure varies according to the type of installation:

LPAR Installation
: In the IBM System z HMC, select *Load*, select *Clear*, then enter the loading address (the device address of the device holding the `/boot` directory with the bootloader). If using a ZFCP disk as the boot device, choose *Load from SCSI* and specify the load address of your FCP adapter as well as WWPN and LUN of the boot device. Now start the loading process.

z/VM Installation

Log in to the VM guest (see Example "Configuration of a z/VM Directory" (↑*Architecture-Specific Information*) for the configuration) as `LINUX1` and proceed to IPL the installed system:

```
IPL 151 CLEAR
```

`151` is an example address of the DASD boot device, replace this value with the correct address.

If using a ZFCP disk as the boot device, specify both the ZFCP WWPN and LUN of the boot device before initiating the IPL. The parameter length is limited to eight characters. Longer numbers must be separated by spaces:

```
SET LOADDEV PORT 50050763 00C590A9 LUN 50010000 00000000
```

Finally, initiate the IPL:

```
IPL FC00
```

`FC00` is an example address of the ZFCP adapter, replace this value with the correct address.

3.13.2 IBM System z: Connecting to the Installed System

After IPLing the installed system, establish a connection with it to complete the installation. The steps involved in this vary depending on the type of connection used at the outset.

Using VNC to Connect

A message in the 3270 terminal asks you to connect to the Linux system using a VNC client. This message is easily missed, however, because it is mixed with kernel messages and because the terminal process might quit before you become aware of the message. If nothing happens for five minutes, try to initiate a connection to the Linux system using a VNC viewer.

If connecting using a Java-capable browser, enter the complete URL, consisting of the IP address of the installed system along with the port number, in the following fashion:

```
http://<IP of installed system>:5801/
```

Using X to Connect

When IPLing the installed system, make sure that the X server used for the first phase of the installation is up and still available before booting from the DASD. YaST opens on this X server to finish the installation. Complications may arise if the system is booted up but unable to connect to the X server in a timely fashion.

Using SSH to Connect

IMPORTANT: IBM System z: Connecting from a Linux or UNIX System

Start SSH in an xterm. Other terminal emulators lack complete support for the text-based interface of YaST.

A message in the 3270 terminal asks you to connect to the Linux system with an SSH client. This message is easily missed, however, because it is mixed with kernel messages and because the terminal process might quit before you become aware of the message.

Once the message appears, use SSH to log in to the Linux system as `root`. If the connection is denied or times out, wait a few minutes then try again.

When the connection is established, execute the command `/usr/lib/YaST2/startup/YaST2.ssh`. `yast` does not suffice in this case.

YaST then starts to complete the installation of the remaining packages and create an initial system configuration.

3.14 Configuration of the Installed System

The system is installed now but not configured for use. No users, hardware, or services are configured, yet. If the configuration fails at one of the steps of this stage, it restarts and continues from the last successful step.

First, provide a password for the account of the system administrator (the `root` user). Configure your Internet access and network connection. With a working Internet connection, you can perform an update of the system as part of the installation. You can also connect to an authentication server for centralized user administration in a local network. Finally, configure the hardware devices connected to the machine.

3.14.1 Password for the System Administrator “root”

`root` is the name of the superuser, the administrator of the system. Unlike regular users, who may or may not have permission to do certain things on the system, `root` has unlimited power to do anything: change the system configuration, install programs, and set up new hardware. If users forget their passwords or have other problems with the system, `root` can help. The `root` account should only be used for system administration, maintenance, and repair. Logging in as `root` for daily work is rather risky: a single mistake could lead to irretrievable loss of system files.

For verification purposes, the password for `root` must be entered twice. Do not forget the `root` password. Once entered, this password cannot be retrieved.

When typing passwords, the characters are replaced by dots, so you do not see the string you are typing. If you are unsure whether you typed the correct string, use the *Test Keyboard Layout* field for testing purposes.

SUSE Linux Enterprise can use the DES, MD5, or Blowfish encryption algorithms for passwords. The default encryption type is Blowfish. To change the encryption type, click *Expert Options* > *Encryption Type* and select the new type.

The `root` can be changed any time later in the installed system. To do so run YaST and start *Security and Users* > *User Management*.

3.14.2 Hostname and Domain Name

The hostname is the computer's name in the network. The domain name is the name of the network. A hostname and domain are proposed by default. If your system is part of a network, the hostname has to be unique in this network whereas the domain name has to be common to all hosts on the network.

In many networks, the system receives its name over DHCP. In this case it is not necessary to modify the hostname and domain name. Select *Change Hostname via DHCP* instead. To be able to access your system using this hostname, even when it is not connected to the network, select *Write Hostname to /etc/hosts*. If you often change networks without restarting the desktop environment (e.g. when switching between different WLANs), do not enable this option, because the desktop system may get confused when the hostname in `/etc/hosts` changes.

To change hostname settings at any time after installation, use YaST *Network Devices* > *Network Card*. For more information, see Section 30.4.1, "Configuring the Network Card with YaST" (page 565).

3.14.3 Network Configuration

TIP: IBM System z: Network Configuration

For the IBM System z platforms, a working network connection is needed at installation time to connect to the target system, the installation source, and the YaST terminal controlling the process. The steps to set up the network are discussed in the network configuration chapter of the *Architecture-Specific Information* manual (Chapter 2, *Preparing for Installation* (↑*Architecture-Specific Information*)). The IBM System z platforms only support the types of network interfaces mentioned there (OSA Token Ring, OSA Ethernet, OSA Gigabit Ethernet, OSA Express Fast Ethernet, Escon, IUCV, and OSA Express High-Speed Token Ring). The YaST dialog simply displays the interface with its settings as already configured. Just confirm this dialog to continue.

By default, *Traditional Method without NetworkManager Applet* is enabled. If desired, you can also use NetworkManager to manage all your network devices. However, the traditional method is the preferred option for server solutions. Find detailed information about NetworkManager in Section 30.6, "Managing Network Connections with NetworkManager" (page 584).

This configuration step also lets you configure the network devices of your system and make security settings, for example, for a firewall or proxy. To configure your network connection later, select *Skip Configuration* and click *Next*. Network hardware can also be configured after the system installation has been completed. If you skip the network device configuration, your system is left offline and is unable to retrieve any available updates.

Apart from the device configuration, the following network settings can be configured in this step:

Network Mode
: Enable or disable the use of NetworkManager as described above.

Firewall
: By default SuSEfirewall2 is enabled on all configured network interfaces. To globally disable the firewall for this computer, click on *disable*. If the firewall is enabled, you may *open* the SSH port in order to allow remote connections via secure shell. To open the detailed firewall configuration dialog, click on *Firewall*. See Section 43.4.1, "Configuring the Firewall with YaST" (page 829) for detailed information.

IPv6
: By default, the IPv6 support is enabled. To disable it, click *Disable IPv6*. For more information about IPv6, see Section 30.2, "IPv6—The Next Generation Internet" (page 554).

VNC Remote Administration
: To administer your machine remotely by VNC, click *Change* > *VNC Remote Administration*, enable remote administration, and open the port in the firewall. If you have multiple network devices and want to select on which to open the port, click *Firewall Details* and select the network device. You can also use SSH, a more secure option, for remote administration.

Proxy
: If you have a proxy server controlling the Internet access in your network, configure the proxy URLs and authentication details in this dialog.

TIP: Resetting the Network Configuration to the Defaults

Reset the network settings to the original proposed values by clicking *Change* > *Reset to Defaults*. This discards any changes made.

Test Internet Connection

After having configured a network connection, you can test it. For this purpose, YaST establishes a connection to the SUSE Linux Enterprise server and downloads the latest

release notes. Read them at the end of the installation process. A successful test is also a prerequisite for registering and updating online.

If you have multiple network interfaces, verify that the desired card is used to connect to the Internet. If not, click *Change Device*.

To start the test, select *Yes, Test Connection to the Internet* and click *Next*. In the next dialog, view the progress of the test and the results. Detailed information about the test process is available via *View Logs*. If the test fails, click *Back* to return to the network configuration to correct your entries.

If you do not want to test the connection at this point, select *No, Skip This Test* then *Next*. This also skips downloading the release notes, configuring the customer center, and updating online. These steps can be performed any time after the system has been initially configured.

3.14.4 Novell Customer Center Configuration

To get technical support and product updates, first register and activate your product. *Novell Customer Center Configuration* provides assistance for doing so.

If you are offline or want to skip this step, select *Configure Later*. This also skips SUSE Linux Enterprise online update.

In *Include for Convenience*, select whether to send unsolicited additional information when registering. This simplifies the registration process. Click on *Details* to obtain in-depth information about data privacy and the data collected.

Apart from activating and registering your product, this module also adds the official update catalog to your configuration. This catalog provides fixes for known bugs or security issues which can be installed via an online update.

To keep your catalogs valid, select *Regularly Synchronize with Customer Center*. This option checks your catalogs and adds newly available catalogs or removes obsolete ones. It does not touch manually added catalogs.

TIP: Technical Support

Find more information about the technical support at http://www.novell.com/support/products/linuxenterpriseserver/.

3.14.5 Online Update

If the *Novell Customer Center Configuration* was successful, select whether to perform a YaST online update. If there are any patched packages available on the servers, download and install them now to fix known bugs or security issues. Directives on how to perform an online update in the installed system are available at Section 8.3.5, “YaST Online Update” (page 140)

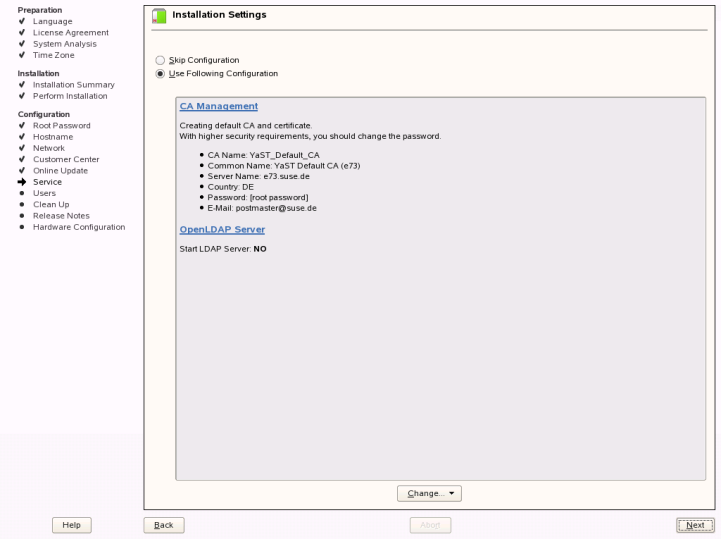
IMPORTANT: Downloading Software Updates

The download of updates might take quite some time, depending on the bandwidth of the Internet connection and the size of the update files. In case the patch system itself is updated, the online update will restart and download more patches after the restart. If the kernel was updated, the system will reboot before completing the configuration.

3.14.6 Network Services

Having configured the network, a dialog opens in which to enable and configure two important network services: a certificate authority and an OpenLDAP server. If preferred, you can skip this configuration proposal for now. After the installation is finished, configure and start the same services with the help of YaST.

Figure 3.6 *Proposed Setup for Network Services*



CA Management

The purpose of a CA (certificate authority) is to guarantee a trust relationship among all network services communicating with each other. Without a CA, you can secure server communications with SSL and TLS separately for each individual service. By default, a CA is created and enabled during the installation. Find details about the creation of a CA with YaST in Chapter 42, *Managing X.509 Certification* (page 807).

OpenLDAP Server

You can run an LDAP service on your host to have a central facility manage a range of configuration files. Typically, an LDAP server handles user account data, but with SUSE Linux Enterprise it can also be used for mail, DHCP, and DNS data. Find details about LDAP and its configuration with YaST in Chapter 36, *LDAP—A Directory Service* (page 667).

TIP: Resetting the Service Configuration to Defaults

Restore the defaults by clicking *Change* > *Reset to Defaults*. This discards any changes made.

3.14.7 Users

If network access was configured successfully during the previous steps of the installation, you can now choose from several user management options. If a network connection has not been configured, create local user accounts. For detailed information about user management, see Section 8.9.1, “User Management” (page 172)the SUSE Linux Enterprise Server documentation.

Local (/etc/passwd)
: Users are administered locally on the installed host. This is a suitable option for stand-alone workstations. User data is managed by the local file `/etc/passwd`. All users who are entered in this file can log in to the system even if no network is available.

  If YaST found a former version of SUSE Linux Enterprise or another system using `/etc/passwd`, it offers to import local users. To do so, check *Read User Data from a Previous Installation* and click *Choose*. In the next dialog, select the users to import and click *OK*.

LDAP
: Users are administered centrally on an LDAP server for all systems in the network. More information is available in Section 36.6, “Configuring an LDAP Client with YaST” (page 688).

NIS
: Users are administered centrally on a NIS server for all systems in the network. See Section 35.2, “Configuring NIS Clients” (page 665) for more information.

Windows Domain
: SMB authentication is often used in mixed Linux and Windows networks. Detailed information is available in Section 37.6, “Samba Server in the Network with Active Directory” (page 713).

NOTE: Content of the Authentication Menu

If you use the custom package selection and one or more authentication methods are missing from the menu, the required packages probably are not installed.

Along with the selected user administration method, you can use Kerberos authentication. This is essential for integrating your SUSE Linux Enterprise to an Active Directory domain, which is described in Section 37.6, “Samba Server in the Network with Active Directory” (page 713). To use Kerberos authentication, select *Set Up Kerberos Authentication*.

3.14.8 Release Notes

After completing the user authentication setup, YaST displays the release notes. Reading them is recommended, because they contain important up-to-date information which was not available when the manuals were printed. If you tested the Internet connection, read the most recent version of the release notes, as fetched from SUSE Linux Enterprise's servers. Use *Miscellaneous* > *Release Notes* to view the release notes after installation.

3.14.9 Hardware Configuration

At the end of the installation, YaST opens a dialog for the configuration of the graphics card and other hardware components connected to the system. Click the individual components to start the hardware configuration. For the most part, YaST detects and configures the devices automatically.

TIP: IBM System z: Hardware Configuration

On the IBM System z, there is no display that would be supported by XFree. Accordingly, you do not find a *Graphics Cards* entry on these systems.

You can skip any peripheral devices and configure them later, as described in Section 8.4, “Hardware” (page 146) . To skip the configuration, select *Skip Configuration* and click *Next*.

However, you should configure the graphics card right away. Although the display settings as configured by YaST should be generally acceptable, most users have very strong preferences as far as resolution, color depth, and other graphics features are concerned. To change these settings, select the respective item and set the values as desired. To test your new configuration, click *Test the Configuration*.

TIP: Resetting Hardware Configuration to Defaults

You can cancel changes by clicking *Change* > *Reset to Defaults*. YaST then shows the original proposal again.

3.14.10 Completing the Installation

After a successful installation, YaST shows the *Installation Completed* dialog. In this dialog, select whether to clone your newly installed system forAutoYaST. To do so, select *Clone This System for AutoYaST*. The profile of the current system is stored in `/root/autoyast.xml`. Cloning is selected by default.

AutoYaST is a system for installing one or more SUSE Linux Enterprise systems automatically without user intervention. AutoYaST installations are performed using a control file with installation and configuration data. For detailed information, refer to Chapter 5, *Automated Installation* (page 85). Finish the installation of SUSE Linux Enterprise with *Finish* in the final dialog.

3.15 Graphical Login

TIP: IBM System z: No Graphical Login

The graphical login is not available on IBM System z platforms.

SUSE Linux Enterprise is now installed and configured. Unless you enabled the automatic login function or customized the default runlevel, you should see the graphical login on your screen in which to enter a username and password to log in to the system. If automatic login is activated, the desktop starts automatically.

4 Remote Installation

SUSE Linux Enterprise® can be installed in several different ways. As well as the usual media installation covered in Chapter 3, *Installation with YaST* (page 17), you can choose from various network-based approaches or even take a completely hands-off approach to the installation of SUSE Linux Enterprise.

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 SUSE Linux Enterprise 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.

4.1 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.

4.1.1 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 3, *Installation with YaST* (page 17).

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 4.2, “Setting Up the Server Holding the Installation Sources” (page 56). Choose an NFS, HTTP, or FTP network server. For an SMB installation source, refer to Section 4.2.5, “Managing an SMB Installation Source” (page 64).

2 Boot the target system using the first CD or DVD of the SUSE Linux Enterprise 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 4.4, “Booting the Target System for Installation” (page 76).

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 4.5.1, “VNC Installation” (page 81).

5 Perform the installation as described in Chapter 3, *Installation with YaST* (page 17). Reconnect to the target system after it reboots for the final part of the installation.

6 Finish the installation.

4.1.2 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 4.2, "Setting Up the Server Holding the Installation Sources" (page 56). Choose an NFS, HTTP, or FTP network server. For an SMB installation source, refer to Section 4.2.5, "Managing an SMB Installation Source" (page 64).
2. Boot the target system using the first CD or DVD of the SUSE Linux Enterprise 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 4.4, "Booting the Target System for Installation" (page 76).

   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 4.5.1, "VNC Installation" (page 81).
5. Perform the installation as described in Chapter 3, *Installation with YaST* (page 17). Reconnect to the target system after it reboots for the final part of the installation.
6. Finish the installation.

4.1.3 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 4.2, “Setting Up the Server Holding the Installation Sources” (page 56). Choose an NFS, HTTP, or FTP network server or configure an SMB installation source as described in Section 4.2.5, “Managing an SMB Installation Source” (page 64).
2. Set up a TFTP server to hold a boot image that can be pulled by the target system. This is described in Section 4.3.2, “Setting Up a TFTP Server” (page 68).
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 4.3.1, “Setting Up a DHCP Server” (page 66).
4. Prepare the target system for PXE boot. This is described in further detail in Section 4.3.5, “Preparing the Target System for PXE Boot” (page 75).

5 Initiate the boot process of the target system using Wake on LAN. This is described in Section 4.3.7, “Wake on LAN” (page 75).

6 On the controlling workstation, open a VNC viewing application or Web browser and connect to the target system as described in Section 4.5.1, “VNC Installation” (page 81).

7 Perform the installation as described in Chapter 3, *Installation with YaST* (page 17). Reconnect to the target system after it reboots for the final part of the installation.

8 Finish the installation.

4.1.4 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 3, *Installation with YaST* (page 17).

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 4.2, “Setting Up the Server Holding the Installation Sources” (page 56). Choose an NFS, HTTP, or FTP network server. For an SMB installation source, refer to Section 4.2.5, “Managing an SMB Installation Source” (page 64).

2. Boot the target system using the first CD or DVD of the SUSE Linux Enterprise 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 4.4.3, “Using Custom Boot Options” (page 78).

   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 83).

5. Perform the installation as described in Chapter 3, *Installation with YaST* (page 17). Reconnect to the target system after it reboots for the final part of the installation.

6. Finish the installation.

4.1.5 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 4.2, “Setting Up the Server Holding the Installation Sources” (page 56). Choose an NFS, HTTP, or FTP network server. For an SMB installation source, refer to Section 4.2.5, “Managing an SMB Installation Source” (page 64).
2. Boot the target system using the first CD or DVD of the SUSE Linux Enterprise 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 4.4.3, “Using Custom Boot Options” (page 78) 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 83).
5. Perform the installation as described in Chapter 3, *Installation with YaST* (page 17). Reconnect to the target system after it reboots for the final part of the installation.
6. Finish the installation.

4.1.6 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 4.2, “Setting Up the Server Holding the Installation Sources” (page 56). Choose an NFS, HTTP, or FTP network server. For the configuration of an SMB installation source, refer to Section 4.2.5, “Managing an SMB Installation Source” (page 64).
2. Set up a TFTP server to hold a boot image that can be pulled by the target system. This is described in Section 4.3.2, “Setting Up a TFTP Server” (page 68).
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 4.3.1, “Setting Up a DHCP Server” (page 66).
4. Prepare the target system for PXE boot. This is described in further detail in Section 4.3.5, “Preparing the Target System for PXE Boot” (page 75).
5. Initiate the boot process of the target system using Wake on LAN. This is described in Section 4.3.7, “Wake on LAN” (page 75).

6 On the controlling workstation, start an SSH client and connect to the target system as described in Section 4.5.2, “SSH Installation” (page 83).

7 Perform the installation as described in Chapter 3, *Installation with YaST* (page 17). Reconnect to the target system after it reboots for the final part of the installation.

8 Finish the installation.

4.2 Setting Up the Server Holding the Installation Sources

Depending on the operating system running on the machine to use as network installation source for SUSE Linux Enterprise, there are several options for the server configuration. The easiest way to set up an installation server is to use YaST on SUSE Linux Enterprise Server 9 or 10 orSUSE Linux 9.3 and higher. On other versions of SUSE Linux Enterprise Server or SUSE Linux Enterprise, set up the installation source manually.

TIP

You can even use a Microsoft Windows machine as installation server for your Linux deployment. See Section 4.2.5, “Managing an SMB Installation Source” (page 64) for details.

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

3 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.

4 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`.

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.

5 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 4.4, “Booting the Target System for Installation” (page 76).

6 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.

4.2.2 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 SUSE Linux Enterprise Server, 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 SUSE Linux Enterprise 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 31, *SLP Services in the Network* (page 605).

4.2.3 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 4.2.2, “Setting Up an NFS Installation Source Manually” (page 58).

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/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`.

4.2.4 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 4.2.2, “Setting Up an NFS Installation Source Manually” (page 58).

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

2a Install the Web server Apache as described in Section 40.1.2, “Installation” (page 746).

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/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`.

4.2.5 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 SUSE Linux Enterprise 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.

4.2.6 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 4.2.2, “Setting Up an NFS Installation Source Manually” (page 58), Section 4.2.3, “Setting Up an FTP Installation Source Manually” (page 61), or Section 4.2.4, “Setting Up an HTTP Installation Source Manually” (page 62).

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 4.2.2, “Setting Up an NFS Installation Source Manually” (page 58), Section 4.2.3, “Setting Up an FTP Installation Source Manually” (page 61), or Section 4.2.4, “Setting Up an HTTP Installation Source Manually” (page 62).

4.3 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.

4.3.1 Setting Up a DHCP Server

There are two ways to set up a DHCP server. For SUSE Linux Enterprise Server 9 and higher, YaST provides a graphical interface to the process. Users of any other SUSE Linux-based products and non-SUSE Linux users should manually edit the configuration files or use the front-end provided by their operating system vendors.

Setting Up a DHCP Server with YaST

To announce the TFTP server's location to the network clients and specify the boot image file the installation target should use, add two declarations to your DHCP server configuration.

1. Log in as `root` to the machine hosting the DHCP server.
2. Start *YaST* > *Network Services* > *DHCP Server*.
3. Complete the setup wizard for basic DHCP server setup.
4. Select *Expert Settings* and select *Yes* when warned about leaving the start-up dialog.
5. In the *Configured Declarations* dialog, select the subnet in which the new system should be located and click *Edit*.
6. In the *Subnet Configuration* dialog select *Add* to add a new option to the subnet's configuration.
7. Select `filename` and enter `pxelinux.0` as the value.

8 Add another option (`next-server`) and set its value to the address of the TFTP server.

9 Select *OK* and *Finish* to complete the DHCP server configuration.

To configure DHCP to provide a static IP address to a specific host, enter the *Expert Settings* of the DHCP server configuration module (Step 4 (page 66)) and add a new declaration of the host type. Add the options `hardware` and `fixed-address` to this host declaration and provide the appropriate values.

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.

4.3.2 Setting Up a TFTP Server

Set up a TFTP server with YaST on SUSE Linux Enterprise Server and SUSE Linux Enterprise 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. Start *YaST* > *Network Services* > *TFTP Server* and install the requested package.
3. 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.
4. 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.

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

6 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 4.3.3, "Using PXE Boot" (page 70).

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`.

4.3.3 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 4.4, “Booting the Target System for Installation” (page 76).

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/`.

4.3.4 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.

4.3.5 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.

4.3.6 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.

4.3.7 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.

Users of SUSE Linux Enterprise Server 9 and higher can use a YaST module called WOL to easily configure Wake on LAN. Users of other versions of SUSE Linux-based operating systems can use a command line tool.

4.3.8 Wake on LAN with YaST

1. Log in as `root`.
2. Start *YaST* > *Network Services* > *WOL*.
3. Click *Add* and enter the hostname and MAC address of the target system.
4. To turn on this machine, select the appropriate entry and click *Wake up*.

4.3.9 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.

4.4 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 4.3.7, “Wake on LAN” (page 75) and Section 4.3.3, “Using PXE Boot” (page 70). 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.

4.4.1 Using the Default Boot Options

The boot options are described in detail in Chapter 3, *Installation with YaST* (page 17). 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 51.2, “Installation Problems” (page 916).

4.4.2 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 4.4.3, “Using Custom Boot Options” (page 78)).

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

Table 4.1 *F Keys During Installation*

Key	Purpose	Available Options	Default Value
F1	Provide help	None	None
F2	Select the installation language	All supported languages	English
F3	Change screen resolution for installation	• Text mode • VESA • resolution #1 • resolution #2 • ...	• Default value depends on your graphics hardware

Key	Purpose	Available Options	Default Value
F4	Select the installation source	• CD-ROM or DVD • SLP • FTP • HTTP • NFS • SMB • Hard Disk	CD-ROM or DVD
F5	Apply driver update disk	Driver	None

4.4.3 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 4.2 *Installation (Boot) Scenarios Used in This Chapter*

Installation Scenario	Parameters Needed for Booting	Boot Options
Chapter 3, *Installation with YaST* (page 17)	None: system boots automatically	None needed
Section 4.1.1, "Simple Remote Installation via VNC—Static Network Configuration" (page 48)	• 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 4.1.2, "Simple Remote Installation via VNC—Dynamic Network Configuration" (page 49)	• Location of the installation server • VNC enablement • VNC password	• `install=(nfs,http,ftp,smb)://`*`path_to_instmedia`* • `vnc=1` • `vncpassword=`*`some_password`*
Section 4.1.3, "Remote Installation via VNC—PXE Boot and Wake on LAN" (page 51)	• Location of the installation server • Location of the TFTP server • VNC enablement • VNC password	Not applicable; process managed through PXE and DHCP

Installation Scenario	Parameters Needed for Booting	Boot Options
Section 4.1.4, "Simple Remote Installation via SSH—Static Network Configuration" (page 52)	• 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 4.1.5, "Simple Remote Installation via SSH—Dynamic Network Configuration" (page 53)	• Location of the installation server • SSH enablement • SSH password	• `install=(nfs,http,ftp,smb)://`*`path_to_instmedia`* • `usessh=1` • `sshpassword=`*`some_password`*
Section 4.1.6, "Remote Installation via SSH—PXE Boot and Wake on LAN" (page 55)	• Location of the installation server • Location of the TFTP server • SSH enablement • SSH password	Not applicable; process managed through PXE and DHCP

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`.

4.5 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.

4.5.1 VNC Installation

Using any VNC viewer software, you can remotely control the installation of SUSE Linux Enterprise 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 4.4.3, “Using Custom Boot Options” (page 78)). 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.

4.5.2 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 4.4.3, “Using Custom Boot Options” (page 78) 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 3, *Installation with YaST* (page 17).

5 Automated Installation

AutoYaST allows you to install SUSE® Linux Enterprise on a large number of machines in parallel. The AutoYaST technology offers great flexibility to adjust deployments to heterogeneous hardware. This chapter tells you how to prepare a simple automated installation and lay out an advanced scenario involving different hardware types and installation purposes.

5.1 Simple Mass Installation

IMPORTANT: Identical Hardware

This scenario assumes you are rolling out SUSE Linux Enterprise to a set of machines with exactly the same hardware configuration.

To prepare for an AutoYaST mass installation, proceed as follows:

1. Create an AutoYaST profile that contains the installation details needed for your deployment as described in Section 5.1.1, “Creating an AutoYaST Profile” (page 86).

2. Determine the source of the AutoYaST profile and the parameter to pass to the installation routines as described in Section 5.1.2, “Distributing the Profile and Determining the autoyast Parameter” (page 88).

3. Determine the source of the SUSE Linux Enterprise installation data as described in Section 5.1.3, “Providing the Installation Data” (page 91).

4 Determine and set up the boot scenario for autoinstallation as described in Section 5.1.4, “Setting Up the Boot Scenario” (page 91).

5 Pass the command line to the installation routines by adding the parameters manually or by creating an `info` file as described in Section 5.1.5, “Creating the `info` File” (page 94).

6 Start the autoinstallation process as described in Section 5.1.6, “Initiating and Monitoring the Autoinstallation” (page 97).

5.1.1 Creating an AutoYaST Profile

An AutoYaST profile tells AutoYaST what to install and how to configure the installed system to get a completely ready-to-use system in the end. It can be created in several different ways:

- Clone a fresh installation from a reference machine to a set of identical machines
- Use the AutoYaST GUI to create and modify a profile to meet your requirements
- Use an XML editor and create a profile from scratch

To clone a fresh reference installation, proceed as follows:

1 Perform a normal installation.

2 After you complete the hardware configuration and read the release notes, check *Clone This Installation for AutoYaST*, if it is not yet checked by default. This creates a ready-to-use profile as `/root/autoinst.xml` that can be used to create clones of this particular installation.

To use the AutoYaST GUI to create a profile from an existing system configuration and modify it to your needs, proceed as follows:

1 As `root`, start YaST.

2 Select *Miscellaneous* > *Autoinstallation* to start the graphical AutoYaST front-end.

3 Select *Tools* > *Create Reference Control File* to prepare AutoYaST to mirror the current system configuration into an AutoYaST profile.

4 As well as the default resources, like boot loader, partitioning, and software selection, you can add various other aspects of your system to the profile by checking the items in the list in *Create a Reference Control File*.

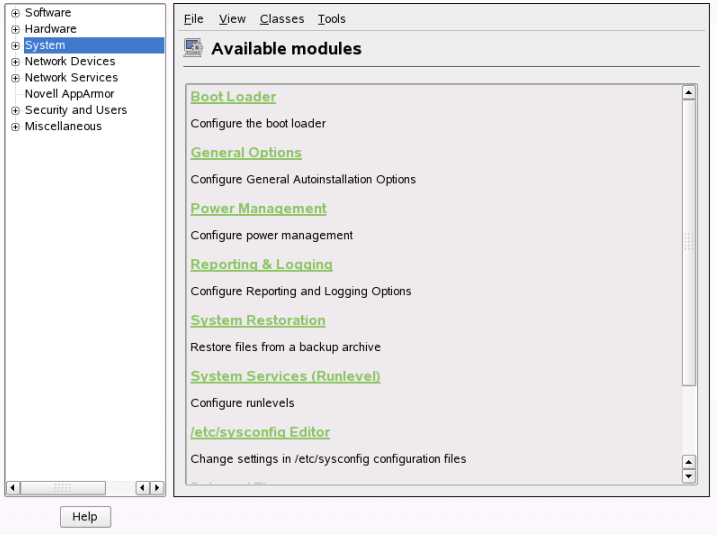
5 Click *Create* to have YaST gather all the system information and write it to a new profile.

6 To proceed, choose one of the following:

- If the profile is complete and matches your requirements, select *File* > *Save as* and enter a filename for the profile, such as `autoinst.xml`.
- Modify the reference profile by selecting the appropriate configuration aspects (such as "Hardware/Printer") from the tree view to the left and clicking *Configure*. The respective YaST module starts but your settings are written to the AutoYaST profile instead of applied to your system. When done, select *File* > *Save as* and enter a suitable name for the profile.

7 Leave the AutoYaST module with *File* > *Exit*.

Figure 5.1 *Editing an AutoYaST Profile with the AutoYaST Front-End*



5.1.2 Distributing the Profile and Determining the autoyast Parameter

The AutoYaST profile can be distributed in several different ways. Depending on the protocol used to distribute the profile data, different AutoYaST parameters are used to make the profile location known to the installation routines on the client. The location of the profile is passed to the installation routines by means of the boot prompt or an `info` file that is loaded upon boot. The following options are available:

Profile Location	Parameter	Description
File	`autoyast=file://`*`path`*	Makes the installation routines look for the control file in specified path (relative to source root directory—`file:///`

Profile Location	Parameter	Description
		`autoinst.xml` if in the top directory of a CD-ROM).
Device	`autoyast=device://`*`path`*	Makes the installation routines look for the control file on a storage device. Only the device name is needed—`/dev/sda1` is wrong, use `sda1` instead.
Floppy	`autoyast=floppy://`*`path`*	Makes the installation routines look for the control file on a floppy in the floppy drive. This option is especially useful, if you want to boot from CD-ROM. If a control file cannot be retrieved from the floppy disk, AutoYaST automatically scans any USB device attached to your machine.
USB (Flash) Disk	`autoyast=usb://`*`path`*	This option triggers a search for the control file on any USB attached device.
NFS	`autoyast=nfs://`*`server/path`*	Has the installation routines retrieve the control file from an NFS server.
HTTP	`autoyast=http://`*`server/path`*	Has the installation routines retrieve the control file from an HTTP server.
HTTPS	`autoyast=https://`*`server/path`*	Has the installation routines retrieve the control file from an HTTPS server.

Profile Location	Parameter	Description
TFTP	`autoyast=tftp://`*`server/path`*	Has the installation routines retrieve the control file from a TFTP server.
FTP	`autoyast=ftp://`*`server/path`*	Has the installation routines retrieve the control file from an FTP server.

Replace the *`server`* and *`path`* placeholders with values matching your actual setup.

AutoYaST includes a feature that allows binding certain profiles to the client's MAC address. Without having to alter the `autoyast=` parameter, you can have the same setup install several different instances using different profiles.

To use this, proceed as follows:

1. Create separate profiles with the MAC address of the client as the filename and put them on the HTTP server that holds your AutoYaST profiles.
2. Omit the exact path including the filename when creating the `autoyast=` parameter, for example:

   ```
   autoyast=http://192.0.2.91/
   ```

3. Start the autoinstallation.

YaST tries to determine the location of the profile in the following way:

1. YaST searches for the profile using its own IP address in uppercase hexadecimal, for example, `192.0.2.91` is `C000025B`.
2. If this file is not found, YaST removes one hex digit and tries again. This action is repeated eight times until the file with the correct name is found.
3. If that still fails, it tries looking for a file with the MAC address of the clients as the filename. The MAC address of the example client is `0080C8F6484C`.

4. If the MAC address–named file cannot be found, YaST searches for a file named `default` (in lowercase). An example sequence of addresses where YaST searches for the AutoYaST profile looks as follows:

```
C000025B
C000025
C00002
C0000
C000
C00
C0
C
0080C8F6484C
default
```

5.1.3 Providing the Installation Data

The installation data can be provided by means of the product CDs or DVDs or using a network installation source. If the product CDs are used as the installation source, physical access to the client to install is needed, because the boot process needs to be initiated manually and the CDs need to be changed.

To provide the installation sources over the network, set up a network installation server (HTTP, NFS, FTP) as described in Section 4.2.1, “Setting Up an Installation Server Using YaST” (page 56). Use an `info` file to pass the server's location to the installation routines.

5.1.4 Setting Up the Boot Scenario

The client can be booted in several different ways:

Network Boot
: As for a normal remote installation, autoinstallation can be initiated with Wake on LAN and PXE, the boot image and control file can be pulled in via TFTP, and the installation sources from any network installation server.

Bootable CD-ROM
: You can use the original SUSE Linux Enterprise media to boot the system for autoinstallation and pull in the control file from a network location or a floppy. Alter-

natively, create your own custom CD-ROM holding both the installation sources and the AutoYaST profile.

The following sections provide a basic outline of the procedures for network boot or boot from CD-ROM.

Preparing for Network Boot

Network booting with Wake on LAN, PXE, and TFTP is discussed in Section 4.1.3, "Remote Installation via VNC—PXE Boot and Wake on LAN" (page 51). To make the setup introduced there work for autoinstallation, modify the featured PXE Linux configuration file (`/srv/tftp/pxelinux.cfg/default`) to contain the `autoyast` parameter pointing to the location of the AutoYaST profile. An example entry for a standard installation looks like this:

```
default linux

# default label linux
    kernel linux append initrd=initrd ramdisk_size=65536 insmod=e100 \
    install=http://192.168.0.22/install/suse-enterprise/
```

The same example for autoinstallation looks like this:

```
default linux

# default label linux
    kernel linux append initrd=initrd ramdisk_size=65536 insmod=e100 \
    install=http://192.168.0.22/install/suse-enterprise/ \
    autoyast=nfs://192.168.0.23/profiles/autoinst.xml
```

Replace the example IP addresses and paths with the data used in your setup.

Preparing to Boot from CD-ROM

There are several ways in which booting from CD-ROM can come into play in AutoYaST installations. Choose from the following scenarios:

Boot from SUSE Linux Enterprise Media, Get the Profile over the Network
: Use this approach if a totally network-based scenario is not possible (for example, if your hardware does not support PXE) and you have physical access to system to install during most of the process.

  You need:

  - The SUSE Linux Enterprise media
  - A network server providing the profile data (see Section 5.1.2, “Distributing the Profile and Determining the autoyast Parameter” (page 88) for details)
  - A floppy containing the `info` file that tells the installation routines where to find the profile

    or

    Access to the boot prompt of the system to install where you manually enter the `autoyast=` parameter

Boot and Install from SUSE Linux Enterprise Media, Get the Profile from a Floppy
: Use this approach if an entirely network-based installation scenario would not work. It requires physical access to the system to install for turning on the target machine, or, in the second case, to enter the profile's location at the boot prompt. In both cases, you may also need to change media depending on the scope of installation.

  You need:

  - The SUSE Linux Enterprise media
  - A floppy holding both the profile and the `info` file

    or

    Access to the boot prompt of the target to enter the `autoyast=` parameter

Boot and Install from Custom Media, Get the Profile from the Media
: If you just need to install a limited number of software packages and the number of targets is relatively low, creating your own custom CD holding both the installa-

tion data and the profile itself might prove a good idea, especially if no network is available in your setup.

5.1.5 Creating the info File

The installation routines at the target need to be made aware of all the different components of the AutoYaST framework. This is done by creating a command line containing all the parameters needed to locate the AutoYaST components, installation sources, and the parameters needed to control the installation process.

Do this by manually passing these parameters at the boot prompt of the installation or by providing a file called `info` that is read by the installation routines (linuxrc). The former requires physical access to any client to install, which makes this approach unsuitable for large deployments. The latter enables you to provide the `info` file on some media that is prepared and inserted into the clients' drives prior to the autoinstallation. Alternatively, use PXE boot and include the linuxrc parameters in the `pxelinux.cfg/default` file as shown in Section “Preparing for Network Boot” (page 92).

The following parameters are commonly used for linuxrc. For more information, refer to the AutoYaST package documentation under `/usr/share/doc/packages/autoyast`.

IMPORTANT: Separating Parameters and Values

When passing parameters to linuxrc at the boot prompt, use `=` to separate parameter and value. When using an `info` file, separate parameter and value with `:`.

Keyword	Value
`netdevice`	The network device to use for network setup (for BOOTP/DHCP requests). Only needed if several network devices are available.
`hostip`	When empty, the client sends a BOOTP request. Otherwise the client is configured using the specified data.

Keyword	Value
netmask	Netmask.
gateway	Gateway.
nameserver	Name server.
autoyast	Location of the the control file to use for the automatic installation, such as autoyast=http://192.168.2.1/profiles/.
install	Location of the installation source, such as install=nfs://192.168.2.1/CDs/.
vnc	If set to 1, enables VNC remote controlled installation.
vncpassword	The password for VNC.
usessh	If set to 1, enables SSH remote controlled installation.

If your autoinstallation scenario involves client configuration via DCHP and a network installation source and you want to monitor the installation process using VNC, your `info` would look like this:

```
autoyast:profile_source install:install_source vnc:1 vncpassword:some_password
```

If you prefer a static network setup at installation time, your `info` file would look like the following:

```
autoyast:profile_source \
install:install_source \
hostip:some_ip \
netmask:some_netmask \
gateway:some_gateway
```

The \ indicate that the line breaks have only been added for the sake of readability. All options must be entered as one continuous string.

The `info` data can be made available to linuxrc in various different ways:

- As a file in the root directory of a floppy that is in the client's floppy drive at installation time.
- As a file in the root directory of the initial RAM disk used for booting the system provided either from custom installation media or via PXE boot.
- As part of the AutoYaST profile. In this case, the AutoYaST file needs to be called `info` to enable linuxrc to parse it. An example for this approach is given below.

linuxrc looks for a string (`start_linuxrc_conf`) in the profile that represents the beginning of the file. If it is found, it parses the content starting from that string and finishes when the string `end_linuxrc_conf` is found. The options are stored in the profile as follows:

```
....
  <install>
....
    <init>
      <info_file>
<![CDATA[
#
# Don't remove the following line:
# start_linuxrc_conf
#
install: nfs:server/path
vnc: 1
vncpassword: test
autoyast: file:///info

# end_linuxrc_conf
# Do not remove the above comment
#
]]>

      </info_file>
    </init>
......
  </install>
....
```

linuxrc loads the profile containing the boot parameters instead of the traditional `info` file. The `install:` parameter points to the location of the installation sources. `vnc` and `vncpassword` indicate the use of VNC for installation monitoring. The `autoyast` parameter tells linuxrc to treat `info` as an AutoYaST profile.

5.1.6 Initiating and Monitoring the Autoinstallation

After you have provided all the infrastructure mentioned above (profile, installation source, and `info` file), you can go ahead and start the autoinstallation. Depending on the scenario chosen for booting and monitoring the process, physical interaction with the client may be needed:

- If the client system boots from any kind of physical media, either product media or custom CDs, you need to insert these into the client's drives.
- If the client is not switched on via Wake on LAN, you need to at least switch on the client machine.
- If you have not opted for remote controlled autoinstallation, the graphical feedback from AutoYaST is sent to the client's attached monitor or, if you use a headless client, to a serial console.

To enable remote controlled autoinstallation, use the VNC or SSH parameters described in Section 5.1.5, “Creating the `info` File” (page 94) and connect to the client from another machine as described in Section 4.5, “Monitoring the Installation Process” (page 81).

5.2 Rule-Based Autoinstallation

The following sections introduce the basic concept of rule-based installation using AutoYaST and provide an example scenario that enables you to create your own custom autoinstallation setup.

5.2.1 Understanding Rule-Based Autoinstallation

Rule-based AutoYaST installation allows you to cope with heterogeneous hardware environments:

- Does your site contain hardware of different vendors?

- Are the machines on your site of different hardware configuration (for example, using different devices or using different memory and disk sizes)?

- Do you intend to install across different domains and need to distinguish between them?

What rule-based autoinstallation does is, basically, generate a custom profile to match a heterogeneous scenario by merging several profiles into one. Each rule describes one particular distinctive feature of your setup (such as disk size) and tells AutoYaST which profile to use when the rule matches. Several rules describing different features of your setup are combined in an AutoYaST `rules.xml` file. The rule stack is then processed and AutoYaST generates the final profile by merging the different profiles matching the AutoYaST rules into one. To illustrate this procedure, refer to Section 5.2.2, "Example Scenario for Rule-Based Autoinstallation" (page 99).

Rule-based AutoYaST offers you great flexibility in planning and executing your SUSE Linux Enterprise deployment. You can:

- Create rules for matching any of the predefined system attributes in AutoYaST

- Combine multiple system attributes (such as disk size and kernel architecture) into one rule by using logical operators

- Create custom rules by running shell scripts and passing their output to the AutoYaST framework. The number of custom rules is limited to five.

NOTE

For more information about rule creation and usage with AutoYaST, refer to the package's documentation under `/usr/share/doc/packages/autoyast2/html/index.html`, Chapter *Rules and Classes*.

To prepare for a rule-based AutoYaST mass installation, proceed as follows:

1. Create several AutoYaST profiles that contain the installation details needed for your heterogeneous setup as described in Section 5.1.1, "Creating an AutoYaST Profile" (page 86).

2. Define rules to match the system attributes of your hardware setup as shown in Section 5.2.2, "Example Scenario for Rule-Based Autoinstallation" (page 99).

3 Determine the source of the AutoYaST profile and the parameter to pass to the installation routines as described in Section 5.1.2, “Distributing the Profile and Determining the autoyast Parameter” (page 88).

4 Determine the source of the SUSE Linux Enterprise installation data as described in Section 5.1.3, “Providing the Installation Data” (page 91)

5 Pass the command line to the installation routines by adding the parameters manually or by creating an `info` file as described in Section 5.1.5, “Creating the `info` File” (page 94).

6 Determine and set up the boot scenario for autoinstallation as described in Section 5.1.4, “Setting Up the Boot Scenario” (page 91).

7 Start the autoinstallation process as described in Section 5.1.6, “Initiating and Monitoring the Autoinstallation” (page 97).

5.2.2 Example Scenario for Rule-Based Autoinstallation

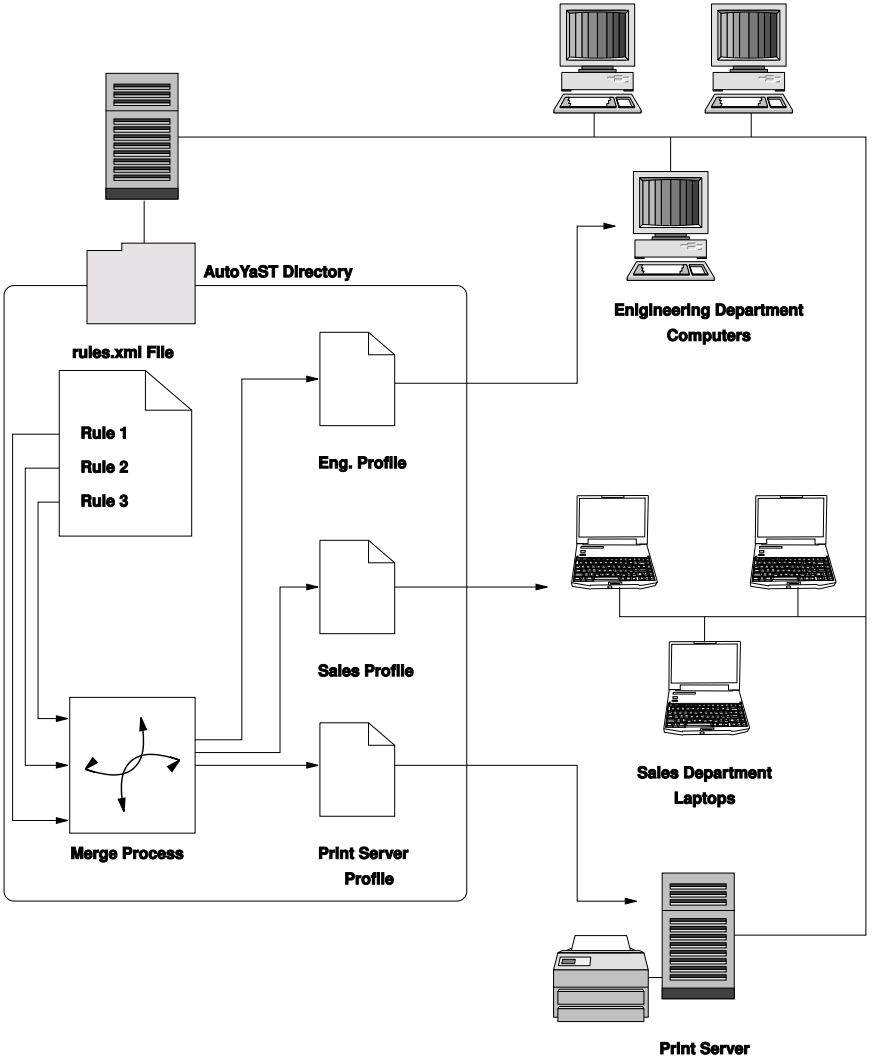
To get a basic understanding of how rules are created, think of the following example, depicted in Figure 5.2, “AutoYaST Rules” (page 100). One run of AutoYaST installs the following setup:

A Print Server
: This machine just needs a minimal installation without a desktop environment and a limited set of software packages.

Workstations in the Engineering Department
: These machines need a desktop environment and a broad set of development software.

Laptops in the Sales Department
: These machines need a desktop environment and a limited set of specialized applications, such as office and calendaring software.

Figure 5.2 *AutoYaST Rules*



In a first step, use one of the methods outlined in Section 5.1.1, “Creating an AutoYaST Profile” (page 86) to create profiles for each use case. In this example, you would create `print.xml`, `engineering.xml`, and `sales.xml`.

In the second step, create rules to distinguish the three hardware types from one another and to tell AutoYaST which profile to use. Use an algorithm similar to the following to set up the rules:

1. Does the machine have an IP of *192.168.27.11*? Then make it the print server.

2. Does the machine have PCMCIA hardware and feature an Intel chipset? Then consider it an Intel laptop and install the sales department software selection.

3. If none of the above is true, consider the machine a developer workstation and install accordingly.

Roughly sketched, this translates into a `rules.xml` file with the following content:

```
<?xml version="1.0"?>
<!DOCTYPE autoinstall SYSTEM "/usr/share/autoinstall/dtd/rules.dtd">
<autoinstall xmlns="http://www.suse.com/1.0/yast2ns"
xmlns:config="http://www.suse.com/1.0/configns">
  <rules config:type="list">
    <rule>
       <hostaddress>
     <match>192.168.27.11</match>
            <match_type>exact</match_type>
       </hostaddress>
       <result>
            <profile>print.xml</profile>
            <continue config:type="boolean">false</continue>
        </result>
    </rule>
    <rule>
       <haspcmcia>
            <match>1</match>
            <match_type>exact</match_type>
       </haspcmcia>
       <custom1>
           <script>
if grep -i intel /proc/cpuinfo > /dev/null; then
echo -n "intel"
else
echo -n "non_intel"
fi;
           </script>
           <match>*</match>
           <match_type>exact</match_type>
       </custom1>
       <result>
            <profile>sales.xml</profile>
            <continue config:type="boolean">false</continue>
        </result>
```

```
        <operator>and</operator>
    </rule>
    <rule>
        <haspcmcia>
            <match>0</match>
            <match_type>exact</match_type>
        </haspcmcia>
  <result>
            <profile>engineering.xml</profile>
            <continue config:type="boolean">false</continue>
        </result>
    </rule>
  </rules>
</autoinstall>
```

When distributing the rules file, make sure that the `rules` directory resides under the `profiles` directory specified in the `autoyast=`*`protocol`*`:`*`serverip`*`/profiles/` URL. AutoYaST looks for a `rules` subdirectory containing a file named `rules.xml` first then loads and merges the profiles specified in the rules file.

The rest of the autoinstallation procedure is carried out as usual.

5.3 For More Information

For in-depth information about the AutoYaST technology, refer to the documentation installed along with the software. It is located under `/usr/share/doc/packages/autoyast2`. The most recent edition of this documentation can be found at http://www.suse.de/~ug/autoyast_doc/index.html.

Deploying Customized Preinstallations

6

Rolling out customized preinstallations of SUSE Linux Enterprise to a large number of identical machines spares you from installing each one of them separately and provides a standardized installation experience for the end users. With YaST firstboot, create customized preinstallation images and determine the workflow for the final personalization steps that involve end user interaction. This is different from AutoYaST that allows completely automated installations; for more information, see Chapter 5, *Automated Installation* (page 85).

Creating a custom installation, rolling it out to your hardware, and personalizing the final product involves the following steps:

1. Prepare the master machine whose disk should be cloned to the client machines. For more information, refer to Section 6.1, “Preparing the Master Machine” (page 104).

2. Customize the firstboot workflow. For more information, refer to Section 6.2, “Customizing the Firstboot Installation” (page 104).

3. Clone the master machine's disk and roll this image out to the clients' disks. For more information, refer to Section 6.3, “Cloning the Master Installation” (page 113).

4. Have the end user personalize the instance of SUSE Linux Enterprise. For more information, refer to Section 6.4, “Personalizing the Installation” (page 113).

6.1 Preparing the Master Machine

To prepare a master machine for a firstboot workflow, proceed as follows:

1. Insert the installation media into the master machine.
2. Boot the machine.
3. Perform a normal installation including all necessary configuration steps and wait for the installed machine to boot. Also install the `yast2-firstboot` package.
4. To define your own workflow of YaST configuration steps for the end user or add your own YaST modules to this workflow, proceed to Section 6.2, “Customizing the Firstboot Installation” (page 104). Otherwise proceed directly to Step 5 (page 104).
5. Enable firstboot as `root`:
   - **5a** Create an empty file `/etc/reconfig_system` to trigger firstboot's execution. This file will be deleted, once the firstboot configuration has been successfully accomplished. Create this file using the following command:

     ```
     touch /etc/reconfig_system
     ```

   - **5b** Enable the firstboot service through the YaST Runlevel Editor.
6. Proceed to Section 6.3, “Cloning the Master Installation” (page 113).

6.2 Customizing the Firstboot Installation

Customizing the firstboot installation may involve several different components. Customizing them is optional. If you do not make any changes, firstboot performs the installation using the default settings. The following options are available:

- Customizing messages to the user as described in Section 6.2.1, “Customizing YaST Messages” (page 105).
- Customizing licenses and license actions as described in Section 6.2.2, “Customizing the License Action” (page 106).
- Customizing the release notes to display as described in Section 6.2.3, “Customizing the Release Notes” (page 107).
- Customizing the order and number of components involved in the installation as described in Section 6.2.4, “Customizing the Workflow” (page 107).
- Configuring additional optional scripts as described in Section 6.2.5, “Configuring Additional Scripts” (page 112).

To customize any of these components, adjust the following configuration files:

`/etc/sysconfig/firstboot`
: Configure various aspects of firstboot, such as release notes, scripts, and license actions.

`/etc/YaST2/firstboot.xml`
: Configure the installation workflow by enabling or disabling components or adding custom ones.

6.2.1 Customizing YaST Messages

By default, an installation of SUSE Linux Enterprise contains several default messages that are localized and displayed at certain stages of the installation process. These include a welcome message, a license message, and a congratulatory message at the end of installation. You can replace any of these with your own versions and include localized versions of them in the installation. To include your own welcome message, proceed as follows:

1. Log in as `root`.
2. Open the `/etc/sysconfig/firstboot` configuration file and apply the following changes:

2a Set `FIRSTBOOT_WELCOME_DIR` to the directory path where you want to store the files containing the welcome message and the localized versions, for example:

```
FIRSTBOOT_WELCOME_DIR="/usr/share/firstboot/"
```

2b If your welcome message has filenames other than `welcome.txt` and `welcome_`*`locale`*`.txt` (where *`locale`* matches the ISO 639 language codes such as “cs” or “de”), specify the filename pattern in `FIRSTBOOT_WELCOME_PATTERNS`. For example:

```
FIRSTBOOT_WELCOME_PATTERNS="mywelcome.txt"
```

If unset, the default value of `welcome.txt` is assumed.

3 Create the welcome file and the localized versions and place them in the directory specified in the `/etc/sysconfig/firstboot` configuration file.

Proceed in a similar way to configure customized license and finish messages. These variables are `FIRSTBOOT_LICENSE_DIR` and `FIRSTBOOT_FINISH_FILE`.

6.2.2 Customizing the License Action

You can customize the way the installation system reacts to a user not accepting the license agreement. There are three different ways in which the system could react to a user's failure to accept the license:

halt
: The firstboot installation is aborted and the entire system shuts down. This is the default setting.

continue
: The firstboot installation continues.

abort
: The firstboot installation is aborted, but the system tries to boot.

Make your choice and set `LICENSE_REFUSAL_ACTION` to the appropriate value.

6.2.3 Customizing the Release Notes

Depending on whether you have changed the instance of SUSE Linux Enterprise you are deploying with firstboot, you probably need to educate the end users about important aspects of their new operating system. A standard installation uses release notes, displayed during one of the final stages of the installation, to provide important information to the users. To have your own modified release notes displayed as part of a firstboot installation, proceed as follows:

1. Create your own release notes file. Use the RTF format as in the example file in `/usr/share/doc/release-notes` and save the result as `RELEASE-NOTES.en.rtf` (for English).

2. Store optional localized versions next to the original version and replace the `en` part of the filename with the actual ISO 639 language code, such as `de` for German.

3. Open the firstboot configuration file from `/etc/sysconfig/firstboot` and set `FIRSTBOOT_RELEASE_NOTES_PATH` to the actual directory where the release notes files are stored.

6.2.4 Customizing the Workflow

By default, a standard firstboot workflow includes the following components:

- Language Selection
- Welcome
- License Agreement
- Host Name
- Network
- Time and Date
- Desktop

- root Password
- User Authentication Method
- User Management
- Hardware Configuration
- Finish Setup

This standard layout of a firstboot installation workflow is not mandatory. You can enable or disable certain components or hook your own modules into the workflow. To modify the firstboot workflow, manually edit the firstboot configuration file `/etc/YaST2/firstboot.xml`. This XML file is a subset of the standard `control.xml` file that is used by YaST to control the installation workflow.

The following overview provides you with enough background to modify the firstboot installation workflow. In it, see the basic syntax of the firstboot configuration file and how the key elements are configured.

Example 6.1 *Configuring the Proposal Screens*

```
…
<proposals config:type="list">❶
    <proposal>❷
        <name>firstboot_hardware</name>❸
        <mode>installation</mode>❹
        <stage>firstboot</stage>❺
        <label>Hardware Configuration</label>❻
        <proposal_modules config:type="list">❼
            <proposal_module>printer</proposal_module>❽
        </proposal_modules>
    </proposal>
    <proposal>
    …
    </proposal>
</proposals>
```

❶ The container for all proposals that should be part of the firstboot workflow.

❷ The container for an individual proposal.

❸ The internal name of the proposal.

❹ The mode of this proposal. Do not make any changes here. For a firstboot installation, this must be set to `installation`.

❺ The stage of the installation process at which this proposal is invoked. Do not make any changes here. For a firstboot installation, this must be set to `firstboot`.

❻ The label to be displayed on the proposal.

❼ The container for all modules that are part of the proposal screen.

❽ One or more modules that are part of the proposal screen.

The next section of the firstboot configuration file consists of the workflow definition. All modules that should be part of the firstboot installation workflow must be listed here.

Example 6.2 *Configuring the Workflow Section*

```
<workflows  config:type="list">
    <workflow>
        <defaults>
            <enable_back>yes</enable_back>
            <enable_next>yes</enable_next>
            <archs>all</archs>
        </defaults>
        <stage>firstboot</stage>
        <label>Configuration</label>
        <mode>installation</mode>
        … <!-- list of modules  -->
        </modules>
    </workflow>
</workflows>
…
```

The overall structure of the `workflows` section is very similar to that of the `proposals` section. A container holds the workflow elements and the workflow elements all include stage, label and mode information just as the proposals introduced in Example 6.1, “Configuring the Proposal Screens” (page 108). The most notable difference is the `defaults` section, which contains basic design information for the workflow components:

`enable_back`
: Include the *Back* button in all dialogs.

`enable_next`
Include the *Next* button in all dialogs.

`archs`
Specify the hardware architectures on which this workflow should be used.

Example 6.3 *Configuring the List of Workflow Components*

```
<modules  config:type="list">❶
    <module>❷
        <label>Language</label>❸
        <enabled config:type="boolean">false</enabled>❹
        <name>firstboot_language</name>❺
    </module>
<modules>
```

❶ The container for all components of the workflow.

❷ The module definition.

❸ The label displayed with the module.

❹ The switch to enable or disable this component in the workflow.

❺ The module name. The module itself must be located under `/usr/share/YaST2/clients` and have the `.ycp` file suffix.

To make changes to the number or order of proposal screens during the firstboot installation, proceed as follows:

1. Open the firstboot configuration file at `/etc/YaST2/firstboot.xml`.
2. Delete or add proposal screens or change the order of the existing ones:
   - To delete an entire proposal, remove the `proposal` element including all its subelements from the `proposals` section and remove the respective `module` element (with subelements) from the workflow.
   - To add a new proposal, create a new `proposal` element and fill in all the required subelements. Make sure that the proposal exists as a YaST module in `/usr/share/YaST2/clients`.

- To change the order of proposals, move the respective `module` elements containing the proposal screens around in the workflow. Note that there may be dependencies to other installation steps that require a certain order of proposals and workflow components.

3 Apply your changes and close the configuration file.

You can always change the workflow of the configuration steps when the default does not meet your needs. Enable or disable certain modules in the workflow or add your own custom ones.

To toggle the status of a module in the firstboot workflow, proceed as follows:

1 Open the `/etc/YaST2/firstboot.xml` configuration file.

2 Change the value for the `enabled` element from `true` to `false` to disable the module or from `false` to `true` to enable it again.

```
<module>
    <label>Time and Date</label>
    <enabled config:type="boolean">true</enabled>
    <name>firstboot_timezone</name>
</module>
```

3 Apply your changes and close the configuration file.

To add a custom made module to the workflow, proceed as follows:

1 Create your own YaST module and store the module file *`module_name`*`.ycp` in `/usr/share/YaST2/clients`.

2 Open the `/etc/YaST2/firstboot.xml` configuration file.

3 Determine at which point of the workflow your new module should be run. In doing so, make sure that possible dependencies to other steps in the workflow are taken into account and resolved.

4 Create a new `module` element inside the `modules` container and add the appropriate subelements:

```
<modules config:type="list">
     …
     <module>
        <label>my_module</label>
        <enabled config:type="boolean">true</enabled>
        <name>filename_my_module</name>
     </module>
</modules>
```

4a Enter the label to display on your module in the `label` element.

4b Make sure that `enabled` is set to `true` to have your module included in the workflow.

4c Enter the filename of your module in the `name` element. Omit the full path and the `.ycp` suffix.

5 Apply your settings and close the configuration file.

TIP: For More Information

For more information about YaST development, refer to http://developer.novell.com/wiki/index.php/YaST.

6.2.5 Configuring Additional Scripts

firstboot can be configured to execute additional scripts after the firstboot workflow has been completed. To add additional scripts to the firstboot sequence, proceed as follows:

1 Open the `/etc/sysconfig/firstboot` configuration file and make sure that the path specified for `SCRIPT_DIR` is correct. The default value is `/usr/share/firstboot/scripts`.

2 Create your shell script, store it in the specified directory, and apply the appropriate file permissions.

6.3 Cloning the Master Installation

Clone the master machine's disk using any of the imaging mechanisms available to you and roll these images out to the target machines.

6.4 Personalizing the Installation

As soon as the cloned disk image is booted, firstboot starts and the installation proceeds exactly as laid out in Section 6.2.4, “Customizing the Workflow” (page 107). Only the components included in the firstboot workflow configuration are started. Any other installation steps are skipped. The end user adjusts language, keyboard, network, and password settings to personalize the workstation. Once this process is finished, a firstboot installed system behaves as any other instance of SUSE Linux Enterprise.

7 Advanced Disk Setup

Sophisticated system configurations require particular disk setups. All common partitioning tasks can be done with YaST. To get persistent device naming with block devices, use the block devices below `/dev/disk/by-id/`. 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 easy creation of data backups. Redundant Array of Independent Disks (RAID) offers increased data integrity, performance, and fault tolerance. SUSE® Linux Enterprise Server also supports multipath I/O. For details, see the chapter about multipath I/O in *Storage Administration Guide*. Starting with SUSE Linux Enterprise 10, there is also the option to use iSCSI as a networked disk. Read more about iSCSI in Chapter 12, *Mass Storage over IP Networks—iSCSI* (page 271).

7.1 LVM Configuration

This section briefly describes the principles behind LVM and its basic features that make it useful under many circumstances. In Section 7.1.2, "LVM Configuration with YaST" (page 118), 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.

7.1.1 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 "Partition Types" (page 156) and Section 8.5.7, "Using the YaST Partitioner" (page 155).

Figure 7.1 *Physical Partitioning versus LVM*

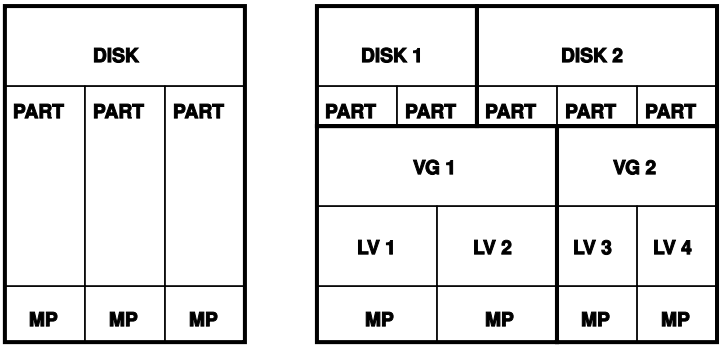


Figure 7.1, "Physical Partitioning versus LVM" (page 116) compares physical partitioning (left) with LVM segmentation (right). On the left side, one single disk has been divided into three physical partitions (PART), each with a mount point (MP) assigned so that the operating system can access them. 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.

Instead of LVM 2, you can use EVMS (Enterprise Volume Management System), which offers a uniform interface for logical volumes and RAID volumes. Like LVM 2, EVMS makes use of the device mapper in kernel 2.6.

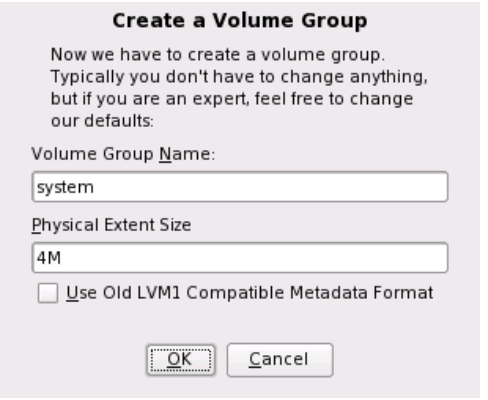
7.1.2 LVM Configuration with YaST

The YaST LVM configuration can be reached from the YaST Expert Partitioner (see Section 8.5.7, “Using the YaST Partitioner” (page 155)). 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 7.2, “Creating a Volume Group” (page 118)). 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 SUSE Linux Enterprise® 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 7.2 *Creating a Volume Group*

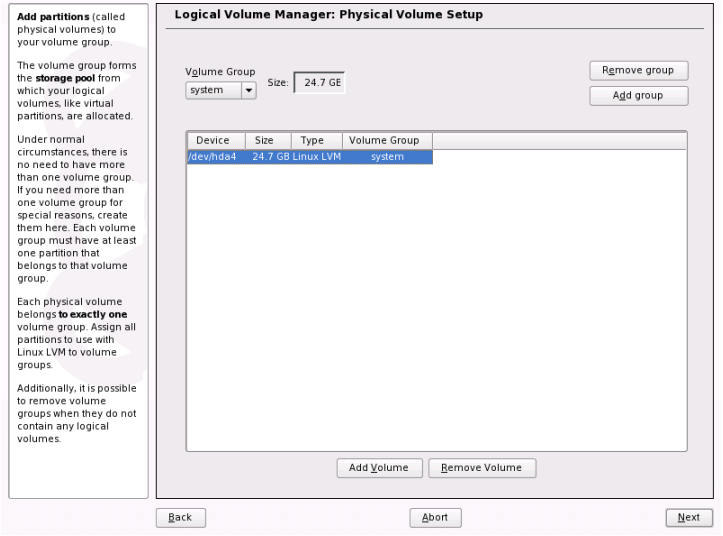


Configuring Physical Volumes

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 7.3 *Physical Volume Setup*

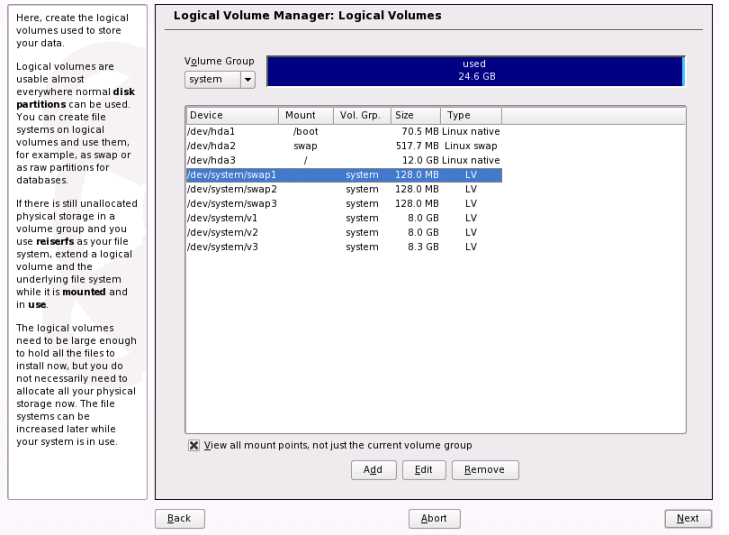


To add a previously unassigned partition to the selected volume group, first click the partition then *Add Volume*. At this point, the name of the volume group is entered next to the selected partition. Assign all partitions reserved for LVM to a volume group. Otherwise, the space on the partition remains 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 7.4 *Logical Volume Management*



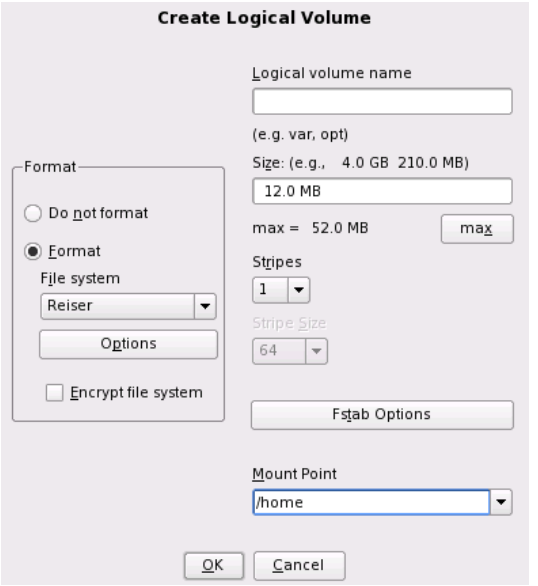
To create a new logical volume, click *Add* and fill out the pop-up that opens. As for partitioning, enter the size, file system, and mount point. Normally, a file system, such as reiserfs or ext2, is created on a logical volume and is then designated a mount point. The files stored on this logical volume 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 7.5 *Creating Logical Volumes*



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.

7.1.3 Storage Management with EVMS

The Enterprise Volume Management System 2 (EVMS2) is a rich, extensible volume manager with built-in cluster awareness. Its plug-in framework allows plug-ins to add functionality for support and knowledge of any partition type. Being cluster-aware, EVMS2 guarantees that managed devices are named identically on each node in the cluster for easier management.

EVMS2 provides a unified interface (`evmsgui` and command line) for managing the following storage resources:

- Physical disks and logical devices on local media and SAN-based media, including iSCSI
- Software RAIDs 0, 1, 4, and 5 for high availability
- Cluster-aware multipath I/O for fault tolerance
- Cluster storage objects with the Cluster Segment Manager (CSM) plug-in
- Volumes for all file systems that have a file system interface module (FSIM) for EVMS2
- Snapshots of volumes

In SUSE Linux Enterprise Server 10, new features include the following:

- EVMS2 and CLVM2 (Cluster Linux Volume Manager 2) use the same multidisk (MD) drivers and device mapper (DM) drivers in the kernel.
- File system plug-ins are available for Heartbeat 2 Cluster Manager and Oracle Cluster File System 2.

EVMS Devices

The EVMS Administration Utility distinguishes five different levels of devices:

Disks
: This is the lowest level of device. All devices that may be accessed as a physical disk are treated as disks.

Segments
: Segments consist of partitions and other memory regions on a disk, such as the master boot record (MBR).

Containers
: These are the counterparts of volume groups in LVM.

Regions
: The available devices are grouped into LVM2 and RAID here.

Volumes
: All devices, regardless of whether they are represented by a real partition, a logical volume, or a RAID device are available with their respective mount points.

If you choose to use EVMS, you must replace your device names with the EVMS device names. Simple partitions are found in `/dev/evms/`, logical volumes in `/dev/evms/lvm/`, and RAID devices in `/dev/evms/md`. To activate EVMS at boot time, add `boot.evms` to the boot scripts in the YaST runlevel editor. See also Section 20.2.3, “Configuring System Services (Runlevel) with YaST” (page 400).

For More Information

For information about using EVMS to manage storage resources, see the *Storage Administration Guide* that is available in `/usr/share/doc/manual/sles-stor_evms_en` after installing the package `sles-stor_evms_en`. More common information about EVMS is also available in EVMS User Guide [`http://evms.sourceforge.net/users_guide/`] at the EVMS project [`http://evms.sourceforge.net/`], hosted on SourceForge*.

7.2 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.

7.2.1 RAID Levels

SUSE® Linux Enterprise offers the option of combining several hard disks into one soft RAID system with the help of YaST—a very reasonable alternative to hardware RAID. 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 block*s, 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.

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.

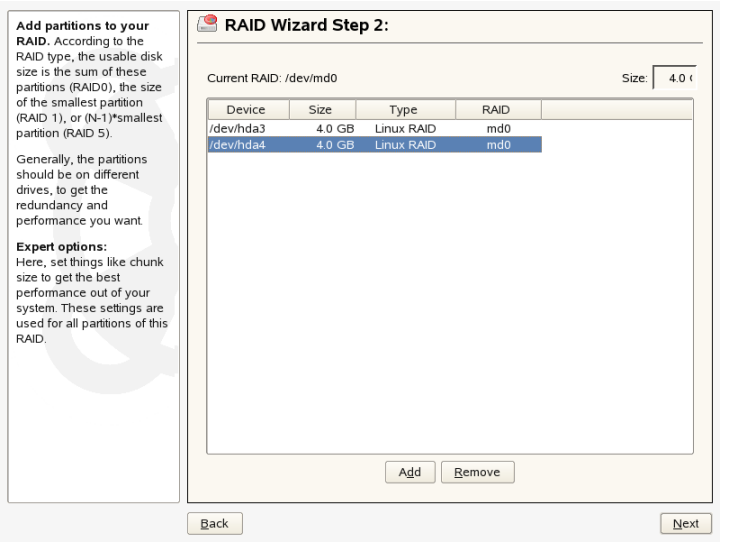
7.2.2 Soft RAID Configuration with YaST

The YaST soft RAID configuration can be reached from the YaST Expert Partitioner, described in Section 8.5.7, “Using the YaST Partitioner” (page 155). 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.

In the next dialog, choose between RAID levels 0, 1, and 5 (see Section 7.2.1, “RAID Levels” (page 124) for details). After *Next* is clicked, the following dialog lists all partitions with either the “Linux RAID” or “Linux native” type (see Figure 7.6, “RAID Partitions” (page 126)). 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 7.6 *RAID Partitions*



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 7.7, “File System Settings” (page 127)).

Figure 7.7 *File System Settings*



As with conventional partitioning, set the file system to use as well as encryption and the mount point for the RAID volume. After completing the configuration with *Finish*, see the `/dev/md0` device and others indicated with *RAID* in the expert partitioner.

7.2.3 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.

7.2.4 For More Information

Configuration instructions and more details for soft RAID can be found in the HOWTOs at:

- http://www.novell.com/documentation/sles10/stor_evms/data/bookinfo.html
- /usr/share/doc/packages/mdadm/Software-RAID.HOWTO.html
- 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&r=1&w=2.

System Configuration with YaST

8

In SUSE Linux Enterprise, YaST handles both the installation and configuration of your system. This chapter describes the configuration of system components (hardware), network access, and security settings, and administration of users. Find a short introduction to the text-based YaST interface in Section 8.12, “YaST in Text Mode” (page 185). For a description of manual system configuration, see Section 20.3, “System Configuration via /etc/sysconfig” (page 402).

Configure the system with YaST using various YaST modules. Depending on the hardware platform and the installed software, there are different ways to access YaST in the installed system.

In KDE or GNOME, start the YaST Control Center from the main menu. Before YaST starts, you are prompted to enter the `root` password, because YaST needs system administrator permissions to change the system files.

To start YaST from the command line, enter the commands `su` (for changing to the user `root`) and `yast2`. To start the text version, enter `yast` instead of `yast2`. Also use the command `yast` to start the program from one of the virtual consoles.

For hardware platforms that do not support a display device of their own and for remote administration on other hosts, run YaST remotely. First, open a console on the host on which to display YaST and enter the command
`ssh -X root@<system-to-configure>` to log in to the system to configure as `root` and redirect the X server output to your terminal. Following the successful SSH login, enter `yast2` to start YaST in graphical mode.

To start YaST in text mode on another system, use `ssh root@<system-to-configure>` to open the connection. Then start YaST with `yast`.

To save time, the individual YaST modules can be started directly. To start a module, enter `yast2 module_name`. View a list of all module names available on your system with `yast2 -l` or `yast2 --list`. Start the network module, for example, with `yast2 lan`.

8.1 YaST Language

To change the language of YaST, select *System* > *Language Selection* in the YaST Control Center. Choose a language, exit the YaST Control Center, log out of the system, then log in again. The next time you start YaST, the new language setting is used. This also changes the language for the entire system.

If you need work in a different language but do not want to change the system language setting, run the YaST with the `LANG` variable set to your preferred language. Use a long language code in the format `langcode_statecode`. For example, for American English, enter `LANG="en_US" yast2`.

This command starts YaST using the specified language. The language is only valid for this YaST session. The language settings of the terminal, other users, and your other sessions remain unchanged.

If you run YaST remotely over SSH, YaST uses the language settings of your local system.

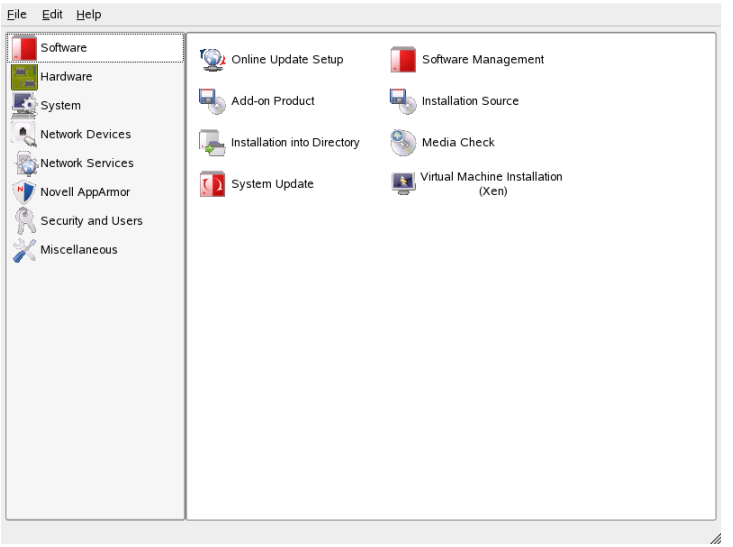
8.2 The YaST Control Center

When you start YaST in the graphical mode, the YaST Control Center, as shown in Figure 8.1, “The YaST Control Center” (page 131), opens. The left frame contains the available categories. When you click a category, its contents are listed in the right frame. Then select the desired module. For example, if you select *Hardware* and click *Sound* in the right frame, a configuration dialog opens for the sound card. The configuration

of the individual items usually consists of several steps. Press *Next* to proceed to the following step.

The left frame of most modules displays the help text, which offers suggestions for configuration and explains the required entries. To get help in modules without a help frame, press F1 or choose *Help*. After selecting the desired settings, complete the procedure by pressing *Accept* on the last page of the configuration dialog. The configuration is then saved.

Figure 8.1 *The YaST Control Center*



NOTE: YaST Software Management Gtk and Qt Front-Ends

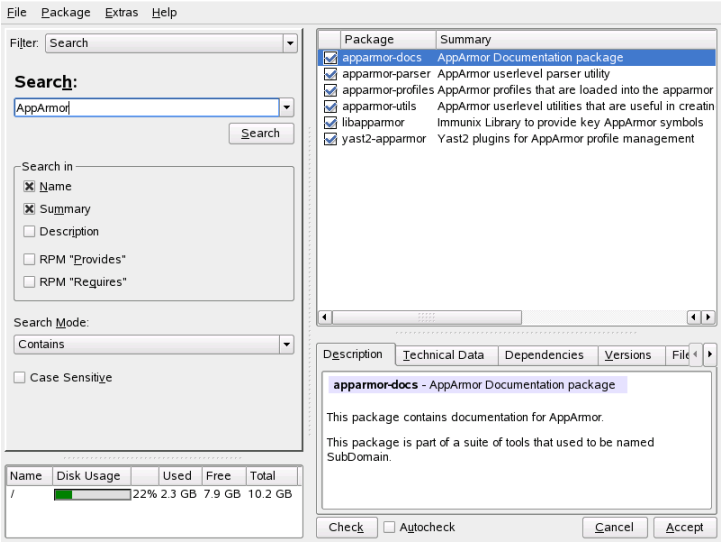
YaST comes with two front-ends depending on the desktop installed on your system. By default, the YaST gtk front-end runs on the GNOME desktop, and the YaST qt front-end on the other desktops. This is defined with the `WANT_UI` variable in the `/sbin/yast2` script. Feature-wise, the gtk front-end is very similar to the qt front-end described in the manuals. One exception is the gtk software management module, which differs considerably from the qt port.

8.3 Software

8.3.1 Installing and Removing Software

To install, uninstall, and update software on your machine, use *Software* > *Software Management*. This opens a package manager dialog as shown in Figure 8.2, “YaST Package Manager” (page 132).

Figure 8.2 *YaST Package Manager*



In SUSE® Linux Enterprise, software is available in the form of RPM packages. Normally, a package contains everything needed for a program: the program itself, the configuration files, and all documentation. A list of individual packages is displayed to the right in the individual package window. The content of this list is determined by the currently selected filter. If, for example, the *Patterns* filter is selected, the individual package window displays all packages of the current selection.

In the package manager, each package has a status that determines what to do with the package, such as “Install” or “Delete.” This status is shown by a symbol in a status box at the beginning of the line. Change the status by clicking or selecting the desired status from the menu that opens when the item is right-clicked. Depending on the current sit-

uation, some of the possible status flags may not be available for selection. For example, a package that has not yet been installed cannot be set to "Delete." View the available status flags with *Help* > *Symbols*.

The font color used for various packages in the individual package window provides additional information. Installed packages for which a newer version is available on the installation media are displayed in blue. Installed packages whose version numbers are higher than those on the installation media are displayed in red. However, because the version numbering of packages is not always linear, the information may not be perfect, but should be sufficient to indicate problematic packages. If necessary, check the version numbers.

Installing Packages

To install packages, select packages for installation and click *Accept*. Selected packages should have the *Install* status icon. The package manager automatically checks the dependencies and selects any other required packages (resolution of dependencies). To view other packages required for installation before clicking *Accept*, choose *Extras* > *Show Automatic Package Changes* from the main menu. After installing packages, continue working with the package manager by clicking *Install More* or close it by clicking *Finish*.

The package manager provides preselected groups for installation. You can select an entire group instead of single packages. To view these groups, use *Filter* in the left frame.

TIP: List of All Available Packages

To display all packages on your installation media, use the filter *Package Groups* and select *zzz All* at the bottom of the tree. SUSE Linux Enterprise contains a number of packages and it might take some time to display this long list.

Installing and Removing Patterns

The *Patterns* filter groups the program packages according to application purpose, such as file or print server. The various groups of the *Patterns* filter are listed with the installed packages preselected.

Click the status box at the beginning of a line to install or uninstall this pattern. Select a status directly by right-clicking the pattern and using the context menu. From the individual package overview to the right, which displays the packages included in the current pattern, select and deselect individual packages.

Installing and Removing Language Support

To find language-specific packages, such as translated texts for the user interface of programs, documentation, and fonts, use the *Languages* filter. This filter shows a list of all languages supported by SUSE Linux Enterprise. If you select one of these, the right frame shows all packages available for this language. Among these, all packages applying to your current software selection are automatically tagged for installation.

To uninstall a language from your system, select a language from the language list and uncheck the status box at the beginning of a line.

NOTE

Because language-specific packages may depend on other packages, the package manager may select additional packages for installation.

Packages and Installation Sources

If you want to find only packages from the specific source, use the *Installation Sources* filter. In the default configuration, this filter shows a list of all packages from the selected source. To restrict the list, use a secondary filter.

To view a list of the all installed packages from the selected installation source, select the filter *Installation Sources* then select *Installation Summary* from *Secondary Filters* and deactivate all check boxes except *Keep*.

The package status in the individual package window can be changed as usual. However, the changed package may no longer meet the search criteria. To remove such packages from the list, update the list with *Update List*.

Installing Source Packages

A package containing the source files for the program is usually available. The sources are not needed for running the program, but you may want to install the sources to compile a custom version of the program.

To install sources for selected program, mark the check box in the *Source* column. If you cannot see a check box, your installation sources do not contain the source of the package.

Saving the Package Selection

If you want to install the same packages on several computers, you can save your configuration to file and use it for other systems. To save your package selection, choose *File* > *Export* from the menu. To import a prepared selection, use *File* > *Import*.

IMPORTANT: Hardware Compatibility

Because this function saves the exact package list, it is only reliable when the hardware is identical on the source and target systems. For more complicated situations, AutoYaST, described in Chapter 5, *Automated Installation* (page 85), may be a better choice.

Removing Packages

To remove packages, assign the correct status to the packages to remove and click *Accept*. Selected packages should have the *Delete* status. If a package required by other installed packages is marked for deletion, the package manager issues an alert with detailed information and alternative solutions.

Reinstalling Packages

If you find damaged files that belong to package or you want to reinstall the original version of a package from your installation media, reinstall the package. To reinstall packages, select packages for reinstallation and click *Accept*. Selected packages should have the *Update* status. If any dependency issues arise with installed packages, the package manager issues an alert with detailed information and alternative solutions.

Searching for Packages, Applications, and Files

To find a specific package, use the *Search* filter. Enter a search string and click *Search*. By specifying various search criteria, you can restrict the search to display a few or even only one package. You can also define special search patterns using wild cards and regular expressions in *Search Mode*.

TIP: Quick Search

In addition to the *Search* filter, all lists of the package manager feature a quick search. Simply enter a letter to move the cursor to the first package in the list whose name begins with this letter. The cursor must be in the list (by clicking the list).

To find a package by name, select *Name*, enter the name of the package to find in the search field, and click *Search*. To find a package by text in the description, select *Summary* and *Descriptions*, enter a search string, and click *Search*.

To search for the package that contains a certain file, enter the name of the file, select *RPM "Provides"*, and click *Search*. To find all packages that depend on a particular package, select *RPM "Requires"*, enter the name of package, and click *Search*.

If you are familiar with the package structure of SUSE Linux Enterprise, you can use the *Package Groups* filter to find packages by subject. This filter sorts the program packages by subjects, such as applications, development, and hardware, in a tree structure to the left. The more you expand the branches, the more specific the selection is. This means fewer packages are displayed in the individual package window.

Installation Summary

After selecting the packages for installation, update, or deletion, view the installation summary with *Installation Summary*. It shows how packages will be affected when you click *Accept*. Use the check boxes to the left to filter the packages to view in the individual package window. For example, to check which packages are already installed, deactivate all check boxes except *Keep*.

The package status in the individual package window can be changed as usual. However, the respective package may no longer meet the search criteria. To remove such packages from the list, update the list with *Update List*.

Information about Packages

Get information about the selected package with the tabs in the bottom right frame. If another version of the package is available, you get information about both versions.

The *Description* tab with the description of the selected package is automatically active. To view information about package size, version, installation media, and other technical details, select *Technical Data*. Information about provided and required files is in *Dependencies*. To view available versions with their installation sources, click *Versions*.

Disk Usage

During the selection of the software, the resource window at the bottom left of the module displays the prospective disk usage of all mounted file systems. The colored bar graph grows with every selection. As long as it remains green, there is sufficient space. The bar color slowly changes to red as you approach the limit of disk space. If you select too many packages for installation, an alert is displayed.

Checking Dependencies

Some packages depend on other packages. This means that the software of the package only works properly if another package is also installed. There are some packages with identical or similar functionality. If these packages use the same system resource, they should not be installed at the same time (package conflict).

When the package manager starts, it examines the system and displays installed packages. When you select to install and remove packages, the package manager can automatically check the dependencies and select any other required packages (resolution of dependencies). If you select or deselect conflicting packages, the package manager indicates this and submits suggestions for solving the problem (resolution of conflicts).

To activate the automatic dependency check, select *Autocheck*, located under the information window. With *Autocheck* activated, any change of a package status triggers an automatic check. This is a useful feature, because the consistency of the package selection is monitored permanently. However, this process consumes resources and can slow down the package manager. For this reason, the automatic check is not activated by default. Regardless of the state of *Autocheck*, a consistency check is performed when you confirm your selection with *Accept*.

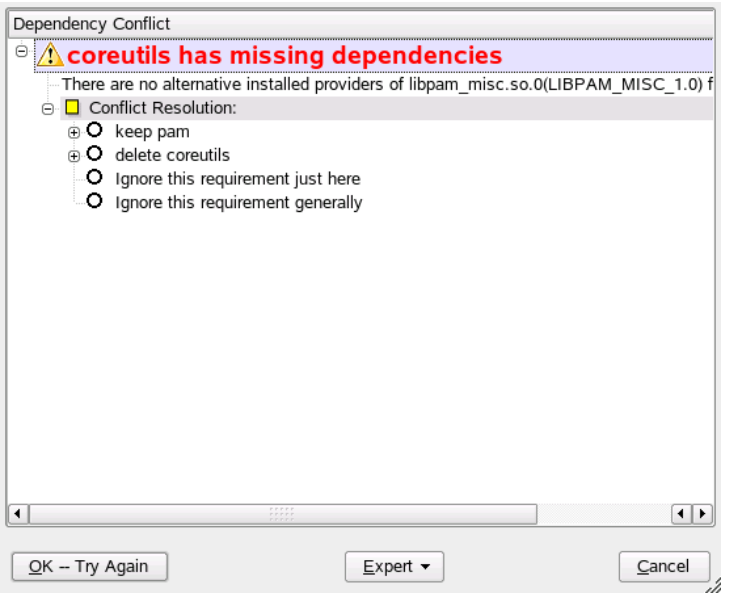
If you click *Check*, located under the information window, the package manager checks if the current package selection results in any unresolved package dependencies or conflicts. In the event of unresolved dependencies, the required additional packages are selected automatically. For package conflicts, the package manager opens a dialog that shows the conflict and offers various options for solving the problem.

For example, `sendmail` and `postfix` may not be installed concurrently. Figure 8.3, "Conflict Management of the Package Manager" (page 138) shows the conflict message prompting you to make a decision. `postfix` is already installed. Accordingly, you can refrain from installing `sendmail`, remove `postfix`, or take the risk and ignore the conflict.

WARNING: Handling Package Conflicts

Unless you are very experienced, follow the suggestions of YaST when handling package conflicts, because otherwise the stability and functionality of your system could be endangered by the existing conflict.

Figure 8.3 *Conflict Management of the Package Manager*



Installing -devel Packages

The package manager provides functions for quick and easy installation of devel and debug packages. To install all devel packages for your installed system, choose *Extras* > *Install All Matching — -devel Packages*. To install all debug packages for your installed system, choose *Extras* > *Install All Matching — -debuginfo Packages*.

8.3.2 Installing Add-On Products

Add-on products are extensions for your system. You can install a third party add-on product or a special extension of your SUSE Linux Enterprise, for example, the SDK add-on or a CD with binary drivers. To install a new add-on, use *Software* > *Add-On Product*. You can select various types of product media, like CD, FTP or local directory. You can work also directly with ISO files. To add an add-on as ISO file media, select *Local Directory* then choose *ISO Images*.

After successfully adding the add-on media, the package manager window appears. If the add-on provides a new pattern, see the new item in the *Patterns* filter. To view the list of all packages from the selected installation source, select the filter *Installation Sources* and choose the installation source to view. To view packages from a selected add-on by package groups, select the secondary filter *Package Groups*.

TIP: Creating Custom Add-On Products

Create your own add-on products with YaST Add-On Creator. Read about the YaST add-on creator at http://developer.novell.com/wiki/index.php/Creating_Add-On_Media_with_YaST. Find technical background information at http://developer.novell.com/wiki/index.php/Creating_Add-Ons.

8.3.3 Selecting the Installation Source

You can use multiple installation sources of several types. Select them and enable their use for installation or update using *Software* > *Installation Source*. For example, you can specify SUSE Software Development Kit as an installation source. When started, it displays a list of all previously registered sources. Following a normal installation from CD, only the installation CD is listed. Click *Add* to include additional sources in

this list. Sources can be CDs, DVDs, or network sources, such as NFS and FTP servers. Even directories on the local hard disk can be selected as the installation medium. See the detailed YaST help text for more details.

All registered sources have an activation status in the first column of the list. Enable or disable individual installation sources by clicking *Activate or Deactivate*. During the installation of software packages or updates, YaST selects a suitable entry from the list of activated installation sources. When you exit the module with *Close*, the current settings are saved and applied to the configuration modules *Software Management* and *System Update*.

8.3.4 Registering SUSE Linux Enterprise

To get technical support and product updates, your system must be registered and activated. If you skipped the registration during installation, register with the help of the *Novell Customer Center Configuration* module from *Software*. This dialog is the same as that described in Section 3.14.4, “Novell Customer Center Configuration” (page 40).

8.3.5 YaST Online Update

Install important updates and improvements with YaST Online Update. The current updates for your SUSE Linux Enterprise are available from the product specific update catalogs containing patches. To add or remove catalogs, use the *Software* > *Installation Source* module, described in Section 8.3.3, “Selecting the Installation Source” (page 139).

NOTE: Error on Accessing the Update Catalog

If you are not able to access the update catalog, this might be due to an expired subscription. Normally, SUSE Linux Enterprise comes with a one or three years subscription, during which you have access to the update catalog. This access will be denied once the subscription ends.

In case of an access denial to the update catalog you will see a warning message with a recommendation to visit the Novell Customer Center and check your subscription. The Novell Customer Center is available at `http://www.novell.com/center/`.

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.

Definition of Terms

Package
: A package is a compressed file in rpm format that contains the files for a particular program.

Patch
: A patch consists of one or more packages—either full packages or patchrpm or deltarpm packages— and may also introduce dependencies to packages that are not installed yet.

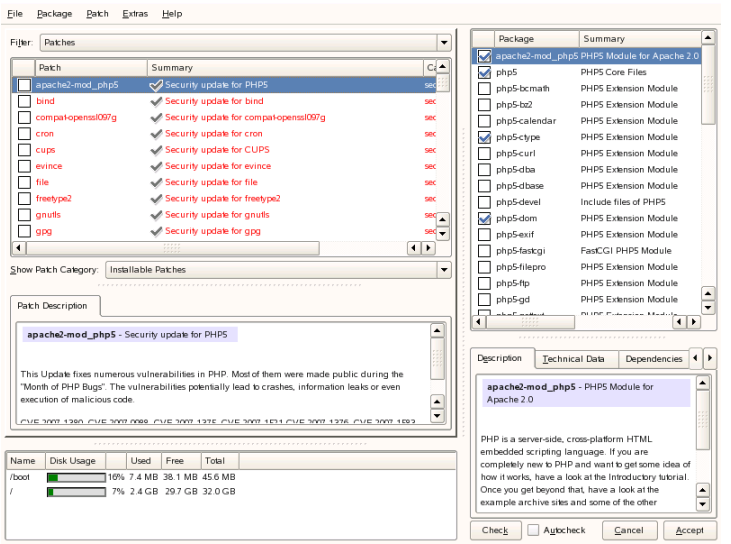
patchrpm
: A patchrpm consists only of files that have been updated since it was first released for SUSE Linux Enterprise 10. Its download size is usually considerably smaller than the size of a package.

deltarpm
: A deltarpm consists only of the binary diff between two defined versions of a package and therefore, has the smallest download size. Before being installed, the rpm package has to be rebuild on the local machine.

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 8.4 *YaST Online Update*



The patch display lists the available patches for SUSE Linux Enterprise. The patches are sorted by security relevance. 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)
: Currently not installed patches 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 SUSE Linux Enterprise.

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.

If you install an up-to-date package from a catalog other than the update catalog, the requirements of a patch for this package may be fulfilled with this installation. In this case a check mark is displayed in front of the patch summary. The patch will be visible in the list until you mark it for installation. This will in fact not install the patch (because the package already is up-to-date), but mark the patch as having been installed.

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.*

TIP: Disabling deltarpms

Since rebuilding rpm packages from deltarpms is a memory and CPU time consuming task, certain setups or hardware configuration might require to disable the usage of deltarpms for performance sake. To disable the use of deltarpms edit the file `/etc/zypp/zypp.conf` and set `download.use_deltarpm` to `false`.

Another alternative for updating software is the ZENworks updater applet for KDE and GNOME. The ZENworks updater helps monitor new patches. It also provides a quick update function. For more information, refer to Section 9.2, “Managing Packages with the ZEN Tools” (page 204).

8.3.6 Automatic Online Update

YaST also offers the possibility to set up an automatic update. Select *Software* > *Automatic Online Update*. 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. Check *Skip Interactive Patches* for the update procedure to proceed automatically. In this case, run a manual *Online Update* from time to install patches that require interaction.

When *Only Download Patches* is checked, the patches are downloaded at the specified time but not installed. They must be installed manually. The patches are downloaded to the rug cache directory, `/var/cache/zmd/web`, by default. Use the command `rug get-prefs cache-directory` to get the current rug cache directory. For more information about rug, see Section 9.1, “Update from the Command Line with rug” (page 200).

8.3.7 Updating from a Patch CD

NOTE

On s390 systems, the Patch CD update option is not available.

The *Patch CD Update* module from the *Software* section installs patches from CD, not from an FTP server. The advantage lies in a much faster update with CD. After the patch CD is inserted, all patches on the CD are displayed in the dialog. Select the desired packages for installation from the list of patches. The module issues an error message if no patch CD is present. Insert the patch CD then restart the module.

8.3.8 Updating the System

Update the version of SUSE Linux Enterprise installed on your system with *Software* > *System Update*. During operation, you can only update application software, not the base system. To update the base system, boot the computer from an installation medium, such as CD. When selecting the installation mode in YaST, select *Update*.

The procedure for updating the system is similar to a new installation. Initially, YaST examines the system, determines a suitable update strategy, and presents the results in a suggestion dialog. Click *Change* or the individual items to change any details.

Update Options

Set the update method for your system. Two options are available.

Update with Installation of New Software and Features Based on the Selection
: To update the entire system to the latest versions of software, select one of the predefined selections. These selections ensure that packages that did not exist previously are also installed.

Only Update Installed Packages
: This option merely updates packages that already exist on the system. No new features are installed.

Additionally, you can use *Delete Outdated Packages* to remove packages that do not exist in the new version. By default, this option is preselected to prevent outdated packages from unnecessarily occupying hard disk space.

Packages

Click *Packages* to start the package manager and select or deselect individual packages for update. Any package conflicts should be resolved with the consistency check. The use of the package manager is covered in detail in Section 8.3.1, “Installing and Removing Software” (page 132).

Backup

During the update, the configuration files of some packages may be replaced by those of the new version. Because you may have modified some of the files in your current system, the package manager normally makes backup copies of the replaced files. With this dialog, determine the scope of these backups.

IMPORTANT: Scope of the Backup

This backup does not include the software. It only contains configuration files.

Language

Primary and other languages currently installed on the system are listed here. Change them by clicking *Language* in the displayed configuration or with *Change* > *Language*. Optionally, adapt the keyboard layout and time zone to the region where the primary language is spoken. Find more about language selection in Section 8.5.15, “Language Selection” (page 163).

Important Information about Updates

The system update is a very complex procedure. For each program package, YaST must first check which version is installed on the computer then determine what needs to be done to replace the old version with the new version correctly. YaST also tries to adopt any personal settings of the installed packages.

In most cases, YaST replaces old versions with new ones without problems. A backup of the existing system should be performed prior to updating to ensure that existing configurations are not lost during the update. Conflicts can then be resolved manually after the update has finished.

8.3.9 Installing into a Directory

This YaST module allows you to install packages into a directory specified by you. Select where to place the root directory, how to name directories, and the type of system and software to install. After entering this module, YaST determines the system settings and lists the default directory, installation instructions, and software to install. Edit these settings by clicking *Change*. All changes must be confirmed by clicking *Accept*. After changes have been made, click *Next* until informed that the installation is complete. Click *Finish* to exit the dialog.

8.3.10 Checking Media

If you encounter any problems using the SUSE Linux Enterprise installation media, you can check the CDs or DVDs with *Software* > *Media Check*. Media problems are more likely to occur with media you burn yourself. To check that a SUSE Linux Enterprise CD or DVD is error-free, insert the medium into the drive and run this module. Click *Start* for YaST to check the MD5 checksum of the medium. This may take several minutes. If any errors are detected, you should not use this medium for installation.

8.4 Hardware

New hardware must first be installed or connected as directed by the vendor. Turn on external devices and start the appropriate YaST module. Most devices are automatically detected by YaST and the technical data is displayed. If the automatic detection fails, YaST offers a list of devices (model, vendor, etc.) from which to select the suitable device. Consult the documentation enclosed with your hardware for more information.

IMPORTANT: Model Designations

If your model is not included in the device list, try a model with a similar designation. However, in some cases the model must match exactly, because similar designations do not always indicate compatibility.

8.4.1 Infrared Device

Configure an infrared device with *Hardware* > *Infrared Device*. Click *Start IrDa* to begin configuration. You can configure *Port* and *Limit Baud Rate* here.

8.4.2 Graphics Card and Monitor

Configure graphics cards and monitors with *Hardware* > *Graphics Card and Monitor*. It uses the the SaX2 interface, described in Section 8.14, "SaX2" (page 191).

8.4.3 Printer

Configure a printer with *Hardware* > *Printer*. If a printer is properly connected to the system, it should be detected automatically. Find detailed instructions for configuring printers with YaST in Section 23.4, "Setting Up a Printer" (page 443).

8.4.4 Hard Disk Controller

Normally, the hard disk controller of your system is configured during the installation. If you add controllers, integrate these into the system with *Hardware* > *Disk Controller*. You can also modify the existing configuration, but this is generally not necessary.

The dialog presents a list of detected hard disk controllers and enables assignment of the suitable kernel module with specific parameters. Use *Test Loading of Module* to check if the current settings work before they are saved permanently in the system.

WARNING: Configuration of the Hard Disk Controller

It is advised to test the settings before making them permanent in the system. Incorrect settings can prevent the system from booting.

8.4.5 Hardware Information

Display detected hardware and technical data using *Hardware* > *Hardware Information*. Click any node of the tree for more information about a device. This module is especially useful, for example, when submitting a support request for which you need information about your hardware.

Save the hardware information displayed to a file by clicking *Save to File*. Select the desired directory and filename then click *Save* to create the file.

8.4.6 IDE DMA Mode

Activate and deactivate the DMA mode for your IDE hard disks and your IDE CD and DVD drives in the installed system with *Hardware* > *IDE DMA Mode*. This module does not have any effect on SCSI devices. DMA modes can substantially increase the performance and data transfer speed in your system.

During installation, the current SUSE Linux Enterprise kernel automatically activates DMA for hard disks but not for CD drives, because default DMA activation for all drives often causes problems with CD drives. Use the DMA module to activate DMA for your drives. If the drive supports the DMA mode without any problems, the data transfer rate of your drive can be increased by activating DMA.

NOTE

DMA (direct memory access) means that your data can be transferred directly to the RAM, bypassing the processor control.

8.4.7 IBM System z: DASD Devices

To add a DASD to the installed system, there are two possibilities:

YaST
: To add a DASD to an installed system, use the YaST DASD module (*Hardware > DASD*). In the first screen, select the disks to make available to your Linux installation and click *Perform Action*. Select *Activate* then leave the dialog with *Next*.

Command Line
: Issue the following command:

```
dasd_configure 0.0.0150 1 0
```

Replace *0.0.0150* with the actual channel number to which the DASD is attached. The last zero of the command line should be `1` if the DASD should be accessed in DIAG mode.

NOTE

In either case, you must run the commands

```
mkinitrd
zipl
```

to make the changes persistent.

8.4.8 IBM System z: ZFCP

To add further FCP-attached SCSI devices to the installed system, use the YaST ZFCP module (*Hardware > ZFCP*). Select *Add* to add an additional device. Select the *Channel Number* (adapter) from the list and specify both *WWPN* and *FCP-LUN*. Finalize the setup by selecting *Next* and *Close*. Verify that the device has been added by checking the output of `cat /proc/scsi/scsi`.

NOTE

To make the changes persistent through a reboot, run the following commands:

```
mkinitrd
zipl
```

8.4.9 Joystick

Configure a joystick connected to the sound card with *Hardware* > *Joystick*. Select your joystick type in the list provided. If your joystick is not listed, select *Generic Analog Joystick*. After selecting your joystick, make sure that it is connected then click *Test* to test the functionality. Click *Continue* and YaST installs the required files. After the *Joystick Test* window appears, test the joystick by moving it in all directions and pressing all buttons. Each movement should be displayed in the window. If you are satisfied with the settings, click *OK* to return to the module and *Finish* to complete configuration.

If you have a USB device, this configuration is not necessary. Plug in the joystick and start using it.

8.4.10 Keyboard Layout

To configure the keyboard for the console, run YaST in text mode then use *Hardware* > *Keyboard Layout*. After clicking the module, the current layout is displayed. To choose another keyboard layout, select the desired layout from the list provided. Test the layout in *Test* by pressing keys on the keyboard.

Fine-tune the settings by clicking *Expert Settings*. Adjust the key repeat rate and delay and configure the start-up state by choosing the desired settings in *Start-Up States*. For *Devices to Lock*, enter a space-separated list of devices to which to apply the Scroll Lock, Num Lock, and Caps Lock settings. Click *OK* to complete the fine-tuning. Finally, after all selections have been made, click *Accept* for your changes to take effect.

To set up the keyboard for the graphical environment, run the graphical YaST then select *Keyboard Layout*. Find information about the graphical configuration in Section 8.14.3, “Keyboard Properties” (page 195).

8.4.11 Mouse Model

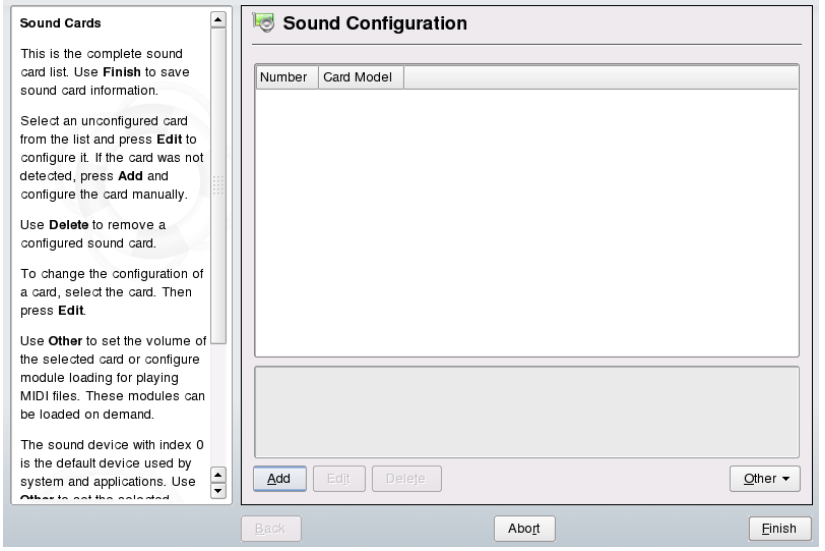
When configuring the mouse for the graphical environment, click *Mouse Model* to access the SaX2 mouse configuration. Refer to Section 8.14.2, “Mouse Properties” (page 194) for details.

To configure your mouse for the text environment, use YaST in text mode. After entering text mode and selecting *Hardware* > *Mouse Model*, use the keyboard arrow keys to choose your mouse from the provided list. Then click *Accept* to save the settings and exit the module.

8.4.12 Sound

Most sound cards are detected automatically and configured with reasonable values during initial installation. To install a card added later or modify settings, use *Hardware* > *Sound*. It is also possible to switch the sequence of the cards.

Figure 8.5 *Sound Configuration*



If YaST cannot detect your sound card automatically, proceed as follows:

1. Click *Add* to open a dialog in which to select a sound card vendor and model. Refer to your sound card documentation for the information required. Find a reference list of sound cards supported by ALSA with their corresponding sound modules in `/usr/share/doc/packages/alsa/cards.txt` and at `http://www.alsa-project.org/alsa-doc/`. After making your selection, click *Next*.

2 In *Sound Card Configuration*, choose the configuration level in the first setup screen:

Quick automatic setup
: You are not required to go through any of the further configuration steps and no sound test is performed. The sound card is configured automatically.

Normal setup
: Adjust the output volume and play a test sound.

Advanced setup with possibility to change options
: Customize all settings manually.

In this dialog, there is also a shortcut to the joystick configuration. Click *Joystick configuration* and select the joystick type in the following dialog to configure a joystick. Click *Next* to continue.

3 In *Sound Card Volume*, test your sound configuration and make adjustments to the volume. You should start at about ten percent to avoid damage to your hearing or the speakers. A test sound should be audible when you click *Test*. If you cannot hear anything, increase the volume. Press *Next* > *Finish* to complete the sound configuration.

To change the configuration of a sound card, go to the *Sound Configuration* dialog, select a displayed *Card Model*, and click *Edit*. Use *Delete* to remove a sound card completely.

Click *Other* to customize one of the following options manually:

Volume
: Use this dialog for setting the volume.

Start Sequencer
: For playback of MIDI files, check this option.

Set as Primary Card
: Click *Set as Primary Card* to adjust the sequence of your sound cards. The sound device with index 0 is the default device used by the system and the applications.

The volume and configuration of all sound cards installed are saved when you click *Finish* in the YaST sound module. The mixer settings are saved to the file `/etc/`

`asound.conf` and the ALSA configuration data is appended to the end of the files `/etc/modprobe.d/sound` and `/etc/sysconfig/hardware`.

8.5 System

This group of modules is designed to help you manage your system. All modules in this group are system-related and serve as valuable tools for ensuring that your system runs properly and your data is managed efficiently.

TIP: IBM System z: Continuing

For IBM System z, continue with Section 8.5.3, “Boot Loader Configuration” (page 154).

8.5.1 Backup

Create a backup of both your system and data using *System* > *System Backup*. However, the backup created by the module does not include the entire system. The system is backed up by saving important storage areas on your hard disk that may be crucial when trying to restore a system, such as the partition table or master boot record (MBR). It can also include the XML configuration acquired from the installation of the system, which is used for AutoYaST. Data is backed up by saving changed files of packages accessible on installation media, entire packages that are unaccessible (such as online updates), and files not belonging to packages, such as many of the configuration files in `/etc` or the directories under `/home`.

8.5.2 Restoration

With *System* > *System Restoration*, restore your system from a backup archive created with *System Backup*. First, specify where the archives are located (removable media, local hard disks, or network file systems). Click *Next* to view the description and contents of the individual archives and select what to restore from the archives.

You can also uninstall packages that were added since the last backup and reinstall packages that were deleted since the last backup. These two steps enable you to restore the exact system state at the time of the last backup.

WARNING: System Restoration

Because this module normally installs, replaces, or uninstalls many packages and files, use it only if you have experience with backups. Otherwise you may lose data.

8.5.3 Boot Loader Configuration

To configure booting for systems installed on your computer, use the *System* > *Boot Loader* module. A detailed description of how to configure the boot loader with YaST is available in Section 21.3, “Configuring the Boot Loader with YaST” (page 416).

8.5.4 Clustering

Find information about Heartbeat and high availability configuration with YaST in *Heartbeat Guide*.

8.5.5 LVM

The logical volume manager (LVM) is a tool for custom partitioning of hard disks with logical drives. Find information about LVM in Section 7.1, “LVM Configuration” (page 115).

8.5.6 EVMS

The *enterprise volume management system* (EVMS) is, like LVM, a tool for custom partitioning and grouping of hard disks into virtual volumes. It is flexible, extensible, and can be tailored using a plug-in model to individual needs of various volume management systems.

EVMS is compatible with existing memory and volume management systems, like DOS, Linux LVM, GPT (GUID partition table), IBM System z, Macintosh, and BSD partitions. More information is provided at `http://evms.sourceforge.net/`.

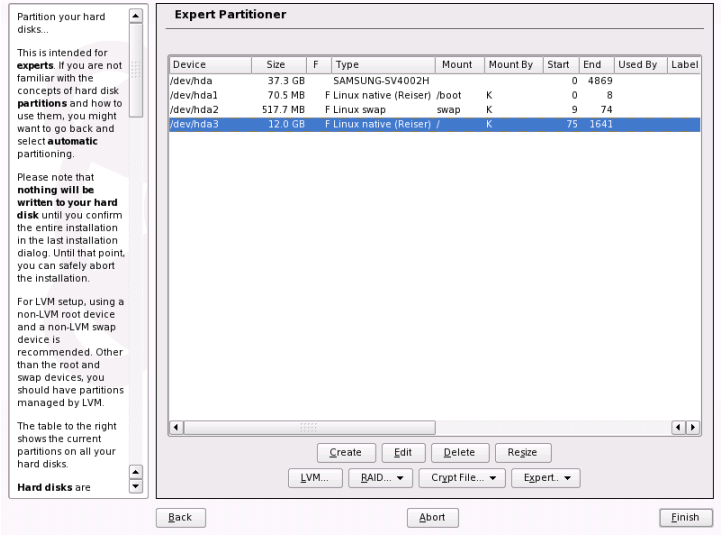
8.5.7 Using the YaST Partitioner

With the expert partitioner, shown in Figure 8.6, “The YaST Partitioner” (page 155), manually modify the partitioning of one or several hard disks. Partitions can be added, deleted, resized, and edited. Also access the soft RAID, EVMS, 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 8.6 *The YaST Partitioner*



TIP: IBM System z: Device Names

IBM System z recognize only DASD and SCSI hard disks. IDE hard disks are not supported. This is why these devices appear in the partition table as `dasda` or `sda` for the first recognized device.

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/hda` or `/dev/sda` (or `/dev/dasda`). Partitions are listed as parts of these devices, such as `/dev/hda1` or `/dev/sda1` (or `/dev/dasda1`, respectively). 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 SUSE Linux Enterprise®, 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 SUSE Linux Enterprise and retain the third and first for other operating systems.

Partition Types

TIP: IBM System z: Hard Disks

On the IBM System z platforms, SUSE Linux Enterprise Server supports SCSI hard disks as well as DASDs (direct access storage devices). While SCSI disks can be partitioned as described below, DASDs can have no more than three partition entries in their partition tables.

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 are 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.

TIP: Hard Disks with a GPT Disk Label

For architectures using the GPT disk label, the number of primary partitions is not restricted. Consequently, there are no logical partitions.

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 “Partition Types” (page 156)).

3. Select the file system to use and a mount point. YaST suggests a mount point for each partition created. Refer to Chapter 25, *File Systems in Linux* (page 471) 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 “Editing a Partition” (page 158).

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.

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*, *Linux EVMS*, and *Linux RAID*. For LVM and RAID details, refer to Section 7.1, "LVM Configuration" (page 115) and Section 7.2, "Soft RAID Configuration" (page 123).

   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 25, *File Systems in Linux* (page 471).

   File System Options
   : Set various parameters for the selected file system here. The defaults are acceptable for most situations.

   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 47, *Encrypting Partitions and Files* (page 869).

   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.

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.

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.

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.

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 8.1 */etc/fstab: Partition Data*

```
/dev/sda1    /data1    auto      noauto,user 0 0
/dev/sda5    /data2    auto      noauto,user 0 0
/dev/sda6    /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".

Partitioning and LVM

From the expert partitioner, access the LVM configuration with *LVM* (see Section 7.1, "LVM Configuration" (page 115)). 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.

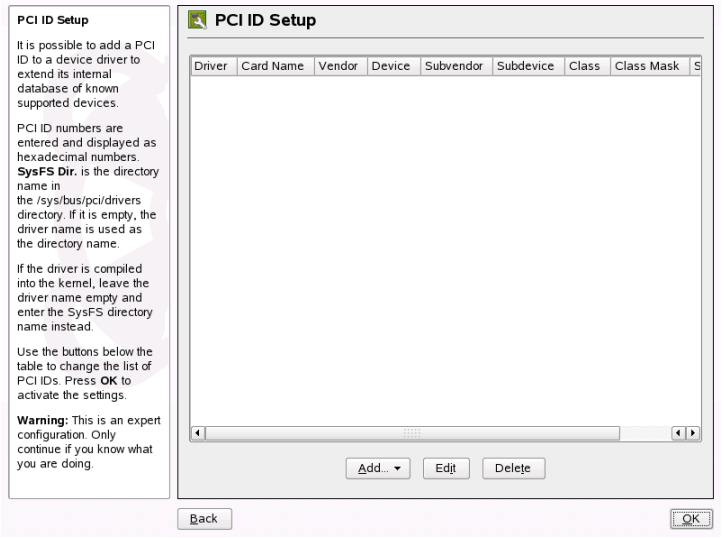
8.5.8 PCI Device Drivers

TIP: IBM System z: Continuing

For IBM System z, continue with Section 8.5.12, “System Services (Runlevel)” (page 162).

Each kernel driver contains a list of device IDs of all devices it supports. If a new device is not in any driver's database, the device is treated as unsupported, even if it can be used with an existing driver. With this YaST module from *System* section, you can add PCI IDs. Only advanced users should attempt to use this YaST module.

Figure 8.7 *Adding a PCI ID*



To add an ID, click *Add* and select how to assign it: by selecting a PCI device from a list or by manually entering PCI values. In the first option, select the PCI device from the provided list then enter the driver or directory name. If the directory is left empty, the driver name is used as the directory name. When assigning PCI ID values manually, enter the appropriate data to set up a PCI ID. Click *OK* to save your changes.

To edit a PCI ID, select the device driver from the list and click *Edit*. Edit the information and click *OK* to save your changes. To delete an ID, select the driver and click *Delete*. The ID immediately disappears from the list. When finished, click *OK*.

8.5.9 Power Management

The *System* > *Power Management* module helps you work with saving energy technologies. It is especially important on laptops to extend their operational time. Find detailed information about using this module in Section 28.6, "The YaST Power Management Module" (page 528).

8.5.10 Powertweak Configuration

Powertweak is a SUSE Linux utility for tweaking your system to peak performance by tuning some kernel and hardware configurations. It should be used only by advanced users. After starting it with *System* > *Powertweak*, it detects your system settings and lists them in tree form in the left frame of the module. You can also use *Search* to find a configuration variable. Select the option to tweak to display it on the screen along with its directory and settings. To save the settings, click *Finish* then confirm it by clicking *OK*.

8.5.11 Profile Manager

Create, manage, and switch among system configurations with *System* > *Profile Management*, the YaST system configuration profile management (SCPM) module. This is especially useful for mobile computers that are used in different locations (in different networks) and by different users. Nevertheless, this feature is useful even for stationary machines, because it enables the use of various hardware components or test configurations.

8.5.12 System Services (Runlevel)

Configure runlevels and the services that start in them with *System* > *System Services (Runlevel)*. For more information about the runlevels in SUSE Linux Enterprise and a description of the YaST runlevel editor, refer to Section 20.2.3, "Configuring System Services (Runlevel) with YaST" (page 400).

8.5.13 /etc/sysconfig Editor

The directory `/etc/sysconfig` contains the files with the most important settings for SUSE Linux Enterprise. Use *System* > */etc/sysconfig Editor* to modify the values and save them to the individual configuration files. Generally, manual editing is not necessary, because the files are automatically adapted when a package is installed or a service is configured. More information about `/etc/sysconfig` and the YaST sysconfig editor is available in Section 20.3.1, “Changing the System Configuration Using the YaST sysconfig Editor” (page 402).

8.5.14 Time and Date Configuration

The time zone is initially set during installation, but you can change it with *System* > *Date and Time*. Also use this to change the current system date and time.

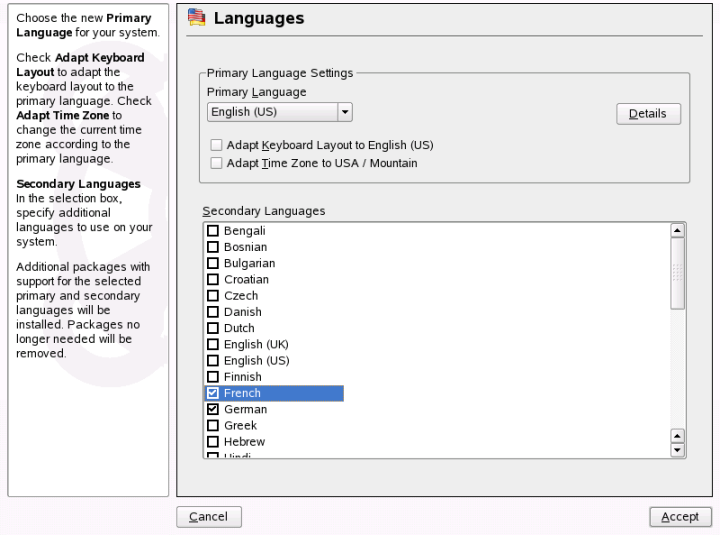
To change the time zone, select the region in the left column and the location or time zone in the right column. With *Hardware Clock Set To*, set whether the system clock should use *Local Time* or *UTC* (Coordinated Universal Time). *UTC* is often used in Linux systems. Machines with additional operating systems, such as Microsoft Windows, mostly use local time.

Set the current system time and date with *Change*. In the dialog that opens, modify the time and date by entering new values or adjusting them with the arrow buttons. Press *Apply* to save the changes.

8.5.15 Language Selection

The primary and secondary languages for your system are set during installation. However, they can be changed at any time using *System* > *Language*. The primary language set in YaST applies to the entire system, including YaST and the desktop environment. This is the language you expect to use most of the time. Secondary languages are languages that are sometimes needed by users for a variety of purposes, such as desktop language or word processing.

Figure 8.8 *Setting the Language*



Select the main language to use for your system in *Primary Language*. To adjust the keyboard or time zone to this setting, enable *Adapt Keyboard Layout* or *Adapt Time Zone*.

Set how locale variables are set for the `root` user with *Details*. Also use *Details* to set the primary language to a dialect not available in the main list. These settings are written into the file `/etc/sysconfig/language`.

8.6 Network Devices

All network devices connected to the system must be initialized before they can be used by a service. The detection and configuration of these devices is done in the module group *Network Devices*.

8.6.1 DSL, ISDN, Modem, or Network Card

To configure a DSL, ISDN, or network interface or a modem, select the appropriate module from the *Network Devices* section. For a device that is detected automatically,

select it from the list then click *Edit*. If your device has not been detected, click *Add* and select it manually. To edit an existing device, select it then click *Edit*. For more detailed information, see Section 30.4, “Configuring a Network Connection with YaST” (page 565). For wireless network interfaces, see Chapter 29, *Wireless Communication* (page 533).

TIP: CDMA and GPRS Modems

You can configure supported CDMA and GPRS modems as regular modems in the YaST modem module.

8.7 Network Services

This group contains tools to configure all kinds of services in the network. These include name resolution, user authentication, and file services.

8.7.1 Mail Transfer Agent

You can configure your mail settings in *Network Services* > *Mail Transfer Agent* if you send your e-mail with sendmail, postfix, or the SMTP server of your provider. You can fetch mail via the fetchmail program, for which you can also enter the details of the POP3 or IMAP server of your provider. Alternatively, use a mail program of your choice, such as KMail or Evolution, to set your access data. In this case, you do not need this module.

To configure your mail with YaST, specify the type of your connection to the Internet in the first dialog. Choose one of the following options:

Permanent
: Select this option if you have a dedicated line to the Internet. Your machine is online permanently, so no dial-up is required. If your system is part of a local network with a central e-mail server, select this option to ensure permanent access to your e-mail messages.

Dial-Up
: This item is relevant for users who have a computer at home, are not located in a network, and occasionally connect to the Internet.

No Connection
: If you do not have access to the Internet and are not located in a network, you cannot send or receive e-mail.

Activate virus scanning for your incoming and outgoing e-mail with AMaViS by selecting that option. The package is installed automatically as soon as you activate the mail filtering feature. In the following dialogs, specify the outgoing mail server (usually the SMTP server of your provider) and the parameters for incoming mail. Set the diverse POP or IMAP servers for mail reception by various users. Using this dialog, you can also assign aliases, use masquerading, or set up virtual domains. Click *Finish* to exit the mail configuration.

8.7.2 Mail Server

IMPORTANT: LDAP-Based Mail Server Configuration

The mail server module of SUSE Linux Enterprise only works if the users, groups, and the DNS and DHCP services are managed with LDAP.

The mail server module allows configuration of SUSE Linux Enterprise as a mail server. YaST assists with the following steps of the configuration process:

Global Settings
: Configures the identification of the local mail server and the maximum size of incoming or outgoing messages and the type of mail transport.

Local Delivery
: Configures the type of local mail delivery.

Mail Transport
: Configures special transport routes for mail depending on its target address.

SPAM Prevention
: Configures the SPAM protection settings of the mail server. This activates the tool AMaViS. Set up the type and strictness of the SPAM check.

Mail Server Relaying
: Determines from which networks the mail server cannot be used for sending non-local mail.

Fetching Mail
: Configures mail pick-up from external mail accounts over various protocols.

Mail Server Domains
: This determines for which domains the mail server should be responsible. At least one master domain must be configured if the server should not run as a null client used exclusively for sending mail without receiving any.

  Distinguish among three domain types:

  main
  : Main or master domain of the local mail server

  local
  : All users who can receive mail in a master domain can also receive mail in a local domain. In the case of a message within the local domain, only the portion before the `@` is evaluated.

  virtual
  : Only users with an explicit address within a virtual domain receive mail. Virtual mail addresses are set up in the user management module of YaST.

8.7.3 Other Available Services

Many other network modules are available in YaST *Network Services*.

DHCP Server
: Use this to set up a custom DHCP server in only a few steps. Chapter 34, *DHCP* (page 641) provides basic knowledge about the subject and a step-by-step description of the configuration process.

DNS Server
: Configuring a DNS server that is responsible for name resolution is recommended for larger networks. You can use *DNS Server* for this as described in Section 33.2, “Configuration with YaST” (page 616). Chapter 33, *The Domain Name System* (page 615) provides background information about DNS.

DNS and Hostname
: Use this module to configure the hostname and DNS if these settings were not already made while configuring the network devices. Also use it to change the host-

name and domain name. If the provider has been configured correctly for DSL, modem, or ISDN access, the list of name servers contains the entries that were extracted automatically from the provider data. If you are located in a local network, you might receive your hostname via DHCP, in which case you should not modify the name.

HTTP Server
: To run your own Web server, configure Apache in *HTTP Server*. Find more information in Chapter 40, *The Apache HTTP Server* (page 745).

Hostnames
: When booting and in small networks, you can use *Hostnames* for hostname resolution instead of DNS. The entries in this module reflect the data of the file `/etc/hosts`. For more information, read Section " `/etc/hosts` " (page 592).

Kerberos Client
: If you have a Kerberos server in your network for network authentication, use *Kerberos Client*. A detailed description of the client configuration with YaST is available in Section 46.6, "Configuring a Kerberos Client with YaST" (page 857).

LDAP Client
: If using LDAP for user authentication in the network, configure the client in *LDAP Client*. Information about LDAP and a detailed description of the client configuration with YaST are available in Section 36.6, "Configuring an LDAP Client with YaST" (page 688).

LDAP Server
: The LDAP server can keep various data in a central directory and distribute it to all clients in your network. Mostly it is used to store shared contact information but its function is not limited to that. An LDAP server can be used also for authentication. Information about LDAP and a detailed description of the server configuration with YaST are available in Chapter 36, *LDAP—A Directory Service* (page 667).

NFS Client
: With NFS client, mount directories provided by NFS server in your own file trees. Use *NFS Client* to configure your system to access an NFS server in the network.

NFS Server

With NFS, run a file server that all members of your network can access. This file server can be used to make certain applications, files, and storage space available to users. In *NFS Server*, you can configure your host as an NFS server and determine the directories to export for general use by the network users. All users with the appropriate permissions can mount these directories in their own file trees. A description of the YaST module and background information about NFS are provided in Chapter 38, *Sharing File Systems with NFS* (page 719).

NIS Client

If you run NIS server to administer user data on a central place and distribute it to the clients, configure the client here. Detailed information about NIS client and configuration with YaST is available in Section 35.2, "Configuring NIS Clients" (page 665).

NIS Server

If you run more than one system, local user administration (using the files `/etc/passwd` and `/etc/shadow`) is impractical and requires a lot of maintenance. In this case, administer user data on a central server and distribute it to the clients from there. NIS is one option for this. Detailed information about NIS and its configuration with YaST is available in Section 35.1.1, "Configuring a NIS Master Server" (page 660).

NTP Client

NTP (network time protocol) is a protocol for synchronizing hardware clocks over a network. Information about NTP and instructions for configuring it with YaST are available in Chapter 32, *Time Synchronization with NTP* (page 609).

Network Services (xinetd)

Configure the network services (such as finger, talk, and ftp) to start when SUSE Linux Enterprise boots using *Network Services*. These services enable external hosts to connect to your computer. Various parameters can be configured for every service. By default, the master service that manages the individual services (inetd or xinetd) is not started.

When this module starts, choose whether to start inetd or xinetd. The selected daemon can be started with a standard selection of services. Alternatively, compose your own selection of services with *Add*, *Delete*, and *Edit*.

WARNING: Configuring Network Services (xinetd)

The composition and adjustment of network services on a system is a complex procedure that requires a comprehensive understanding of the concept of Linux services. The default settings are usually sufficient.

Proxy

Configure Internet proxy client settings in *Proxy*. Click *Enable Proxy* then enter the desired proxy settings. You can test these settings by clicking *Test Proxy Settings*. A small window informs you whether your proxy settings work correctly. After your settings have been entered and tested, save them by clicking *Accept*.

Remote Administration

To administer your machine remotely from another machine, use *Remote Administration*. To maintain your system remotely, use a VNC client, such as krdc, or a Java-enabled browser. Although remote administration using VNC is simple and fast, it is less secure than using SSH, so you should always keep this in mind when using a VNC server. Find detailed information about installing with a VNC client in Section 4.1.1, "Simple Remote Installation via VNC—Static Network Configuration" (page 48).

Allow remote administration by selecting *Allow Remote Administration* in *Remote Administration Settings*. Selecting *Do Not Allow Remote Administration* disables this function. Click *Open Port in Firewall* to allow access to your computer. Clicking *Firewall Details* displays network interfaces with open ports in the firewall. Select the desired interface and click *OK* to return to the main dialog. Click *Accept* to complete the configuration.

The YaST *Remote Administration* module is highly recommended for configuring VNC on your machine. Although the SaX2 interface also allows you to set remote access properties, it is not a substitute for YaST. It only enables you to configure your X server as a host for VNC sessions.

Routing

Use *Routing* to configure the paths data takes over the network. In most cases, only enter the IP address of the system through which to send all data in *Default Gateway*. To create more complicated configurations, use *Expert Configuration*.

Samba Server

In a heterogeneous network consisting of Linux and Windows hosts, Samba controls the communication between the two worlds. Information about Samba and the configuration of servers is provided in Chapter 37, *Samba* (page 703).

SLP Server

With *service location protocol* (SLP), you can configure clients in your network without knowledge of server names and services that these servers provide. Detailed information about SLP servers and configuration with YaST are described in Chapter 31, *SLP Services in the Network* (page 605).

TFTP Server

A TFTP server in not an FTP server. While an FTP server uses the File Transfer Protocol (FTP), a TFTP server uses the much simpler Trivial File Transfer Protocol (TFTP) without security features. TFTP servers are usually used to boot diskless workstations, X terminals, and routers. Detailed information about TFTP servers and configuration with YaST are described in Section 4.3.2, "Setting Up a TFTP Server" (page 68).

WOL

WOL (wake on LAN) refers to the possibility of waking up a computer from standby mode over the network using special packages. It only works with motherboards that support this functionality in their BIOS. WOL configuration with YaST is described in Section 4.3.7, "Wake on LAN" (page 75).

Windows Domain Membership

In a heterogeneous network consisting of Linux and Windows hosts, Samba controls the communication between the two worlds. With the *Samba Client* module, you can configure your computer as member of a Windows domain. Find information about Samba and the configuration of clients in Chapter 37, *Samba* (page 703).

iSCSI Target

iSCSI technology provides an easy and reasonably inexpensive solution for connecting Linux computers to central storage systems. To configure the server side, use *Miscellaneous* > *iSCSI Target*. Find more information about configuration of iSCSI with YaST in Chapter 12, *Mass Storage over IP Networks—iSCSI* (page 271).

iSCSI Initiator

To configure a connection to central storage, use *Miscellaneous* > *iSCSI Initiator*. Find more information about configuration of iSCSI with YaST in Chapter 12, *Mass Storage over IP Networks—iSCSI* (page 271).

8.8 AppArmor

Novell AppArmor is designed to provide easy-to-use application security for both servers and workstations. Novell AppArmor is an access control system that lets you specify which files each program may read, write, and execute. To enable or disable Novell AppArmor on your system, use *AppArmor Control Panel*. Information about Novell AppArmor and a detailed description of the configuration with YaST are available in *Novell AppArmor Administration Guide* (↑*Novell AppArmor Administration Guide*).

8.9 Security and Users

A basic aspect of Linux is its multiuser capability. Consequently, several users can work independently on the same Linux system. Each user has a user account identified by a login name and a personal password for logging in to the system. All users have their own home directories where personal files and configurations are stored.

8.9.1 User Management

Create and edit users with *Security and Users* > *User Management*. It provides an overview of users in the system, including NIS, LDAP, Samba, and Kerberos users if requested. If you are part of an extensive network, click *Set Filter* to list all users categorically. You can also customize the filter settings by clicking *Customize Filter*.

TIP: Applying Configuration Changes without Closing the Module

Whenever you need to make multiple configuration changes and want to avoid restarting the user and group configuration module for every single one of these changes, use *Write Changes Now* to save your changes without exiting the configuration module.

Adding Users

To add a new user, proceed as follows:

1. Click *Add*.
2. Enter the necessary data for *User Data*. If you do not need to adjust any more detailed settings for this new user, proceed to Step 5 (page 173).
3. To change a user's ID, home directory name, default home, group, group memberships, directory permissions, or login shell, open the *Details* tab and change the default values.
4. To adjust user's password expiration, length, and expiration warnings, use the *Password Settings* tab.
5. Write the user account configuration by clicking *Accept*.

The new user can immediately log in with the created login name and password.

Deleting Users

To delete a user, proceed as follows:

1. Select the user from the list.
2. Click *Delete*.
3. Determine whether to delete or keep the home directory of the user to delete.
4. Click *Yes* to apply your settings.

Changing the Login Configuration

To change the login configuration, proceed as follows:

1. Select the user from the list.
2. Click *Edit*.

3 Adjust the settings under *User Data*, *Details*, and *Password Settings*.

4 Save the user account configuration by clicking *Accept*.

Managing Encrypted Home Directories

You can create an encrypted home directory as part of the user account creation. To create an encrypted home directory for a user, proceed as follows:

1 Click *Add*.

2 Enter the required data for *User Data*.

3 In the *Details* tab, activate *Use Encrypted Home Directory*.

4 Apply your settings with *Accept*.

To create an encrypted home for an existing user, proceed as follows:

1 Select a user from the list and click *Edit*.

2 In the *Details* tab, enable *Use Encrypted Home Directory*.

3 Enter the password of the selected user.

4 Apply your settings with *Accept*.

To disable the encryption of home directories, proceed as follows:

1 Select a user from the list and click *Edit*.

2 In the *Details* tab, disable *Use Encrypted Home Directory*.

3 Enter the password of the selected user.

4 Apply your settings with *Accept*.

For more information about encrypted homes, see Section 47.2, “Using Encrypted Home Directories” (page 873).

Auto Login

WARNING: Using Auto Login

Using the auto login feature on any system that can be physically accessed by more than one person is a potential security risk. Any user accessing this system can manipulate the data on it. If your system contains confidential data, do not use the auto login functionality.

If you are the only user of your system, you can configure auto login. It automatically logs a user into the system after start. Only one selected user can use the auto login function. Auto login works only with KDM or GDM.

To activate auto login, select the user from the list of users and click *Expert Options* > *Login Settings*. Then choose *Auto Login* and click *OK*.

To deactivate this functionality, select the user and click *Expert Options* > *Login Settings*. Then uncheck *Auto Login* and click *OK*.

Login without a Password

WARNING: Allowing Login without a Password

Using the passwordless login feature on any system that can be physically accessed by more than one person is a potential security risk. Any user accessing this system can manipulate the data on it. If your system contains confidential data, do not use this functionality.

Login without a password automatically logs a user into the system after the user enters the username in the login manager. It is available to multiple users on a system and works only with KDM or GDM.

To activate the function, select the user from the list of users and click *Expert Options* > *Login Settings*. Then choose *Passwordless Login* and click *OK*.

To deactivate this function, select the user for whom to disable this functionality from the list of users and click *Expert Options* > *Login Settings*. Then uncheck *Passwordless Login* and click *OK*.

Disabling User Login

To create a system user that should not be able to log in to the system but under whose identity several system-related tasks should be managed, disable the user login when creating the user account. Proceed as follows:

1. Click *Add*.
2. Enter the required data for *User Data*.
3. Check *Disable User Login*.
4. Apply your settings with *Accept*.

To disable login for an existing user, proceed as follows:

1. Select the user from the list and click *Edit*.
2. Check *Disable User Login* in *User Data*.
3. Apply your settings with *Accept*.

Enforcing Password Policies

On any system with multiple users, it is a good idea to enforce at least basic password security policies. Users should change their passwords regularly and use strong passwords that cannot easily be exploited. For information about how to enforce stricter password rules, refer to Section 8.9.3, “Local Security” (page 179). To enforce password rotation, create a password expiration policy.

To configure the password expiration policy for a new user, proceed as follows:

1. Click *Add*.
2. Enter the required data in *User Data*.
3. Adjust the values in *Password Settings*.
4. Apply your settings with *Accept*.

To change the password expiration policy for an existing user, proceed as follows:

1. Select the user from the list and click *Edit*.
2. Adjust the values in *Password Settings*.
3. Apply your settings with *Accept*.

You can limit the lifetime of any user account by specifying a date of expiration for this particular account. Specify the *Expiration Date* in the `YYYY-MM-DD` format and leave the user configuration. If no *Expiration Date* is given, the user account never expires.

Changing the Default Settings for New Users

When creating new local users, several defaults settings are used by YaST. You can change these default settings to meet your requirements:

1. Select *Expert Options* > *Defaults for New Users*.
2. Apply your changes to any or all of the following items:
   - *Default Group*
   - *Secondary Groups*
   - *Default Login Shell*
   - *Path Prefix for Home Directory*
   - *Skeleton for Home Directory*
   - *Umask for Home Directory*
   - *Default Expiration Date*
   - *Days after Password Expiration Login is Usable*
3. Apply your changes with *Accept*.

Several other security-related default settings can be changed using the *Local Security* module. Refer to Section 8.9.3, “Local Security” (page 179) for information.

Changing the Password Encryption

NOTE

Changes in password encryption apply only to local users.

SUSE Linux Enterprise can use DES, MD5, or Blowfish for password encryption. The default password encryption method is Blowfish. The encryption method is set during installation of the system, as described in Section 3.14.1, “Password for the System Administrator “root”” (page 37). To change the password encryption method in the installed system, select *Expert Options* > *Password Encryption*.

Changing the Authentication and User Sources

The user administration method (such as NIS, LDAP, Kerberos, or Samba) is set during installation, as described in Section 3.14.7, “Users” (page 43). To change the user authentication method in the installed system, select *Expert Options* > *Authentication and User Sources*. The module provides a configuration overview and the option to configure the client. Advanced client configuration is also possible using this module.

8.9.2 Group Management

To create and edit groups, select *Security and Users* > *Group Management* or click *Groups* in the user administration module. Both dialogs have the same functionality, allowing you to create, edit, or delete groups.

The module gives an overview of all groups. As in the user management dialog, change filter settings by clicking *Set Filter*.

To add a group, click *Add* and enter the appropriate data. Select group members from the list by checking the corresponding box. Click *Accept* to create the group. To edit a group, select the group to edit from the list and click *Edit*. Make all necessary changes then save them with *Accept*. To delete a group, simply select it from the list and click *Delete*.

Click *Expert Options* for advanced group management. Find more about these options in Section 8.9.1, “User Management” (page 172).

8.9.3 Local Security

To apply a set of security settings to your entire system, use *Security and Users* > *Local Security*. These settings include security for booting, login, passwords, user creation, and file permissions. SUSE Linux Enterprise offers three preconfigured security sets: *Home Workstation*, *Networked Workstation*, and *Network Server*. Modify the defaults with *Details*. To create your own scheme, use *Custom Settings*.

The detailed or custom settings include:

Password Settings
: To have new passwords checked by the system for security before they are accepted, click *Check New Passwords* and *Test for Complicated Passwords*. Set the minimum password length for newly created users. Define the period for which the password should be valid and how many days in advance an expiration alert should be issued when the user logs in to the text console.

Boot Settings
: Set how the key combination Ctrl + Alt + Del should be interpreted by selecting the desired action. Normally, this combination, when entered in the text console, causes the system to reboot. Do not modify this setting unless your machine or server is publicly accessible and you are afraid someone could carry out this action without authorization. If you select *Stop*, this key combination causes the system to shut down. With *Ignore*, this key combination is ignored.

  If you use the KDE login manager (KDM), set permissions for shutting down the system in *Shutdown Behavior of KDM*. Give permission to *Only root* (the system administrator), *All Users*, *Nobody*, or *Local Users*. If *Nobody* is selected, the system can only be shut down from the text console.

Login Settings
: Typically, following a failed login attempt, there is a waiting period lasting a few seconds before another login is possible. This makes it more difficult for password sniffers to log in. Optionally activate *Record Successful Login Attempts*. If you suspect someone is trying to discover your password, check the entries in the system log files in `/var/log`. To grant other users access to your graphical login screen

over the network, enable *Allow Remote Graphical Login*. Because this access possibility represents a potential security risk, it is inactive by default.

User Addition
: Every user has a numerical and an alphabetical user ID. The correlation between these is established using the file `/etc/passwd` and should be as unique as possible. Using the data in this screen, define the range of numbers assigned to the numerical part of the user ID when a new user is added. A minimum of 500 is suitable for users. Automatically generated system users start with 1000. Proceed in the same way with the group ID settings.

Miscellaneous Settings
: To use predefined file permission settings, select *Easy*, *Secure*, or *Paranoid*. *Easy* should be sufficient for most users. The setting *Paranoid* is extremely restrictive and can serve as the basic level of operation for custom settings. If you select *Paranoid*, remember that some programs might not work correctly or even at all, because users no longer have permission to access certain files.

  Also set which user should launch the `updatedb` program, if installed. This program, which automatically runs on a daily basis or after booting, generates a database (locatedb) in which the location of each file on your computer is stored. If you select *Nobody*, any user can find only the paths in the database that can be seen by any other (unprivileged) user. If `root` is selected, all local files are indexed, because the user `root`, as superuser, may access all directories. Make sure that the options *Current Directory in root's Path* and *Current Directory in Path of Regular Users* are deactivated. Only advanced users should consider using these options because these settings may pose a significant security risk if used incorrectly. To have some control over the system even if it crashes, click *Enable Magic SysRq Keys*.

Click *Finish* to complete your security configuration.

8.9.4 Certificate Management

Certificates are used for communication and can also be found, for example, on company ID cards. To manage them or import a common server certificate, use *Security and Users* > *CA Management*. Detailed information about certificates, their technologies, and management with YaST are provided in Chapter 42, *Managing X.509 Certification* (page 807).

8.9.5 Firewall

SuSEfirewall2 can protect your machine against attacks from the Internet. Configure it with *Security and Users* > *Firewall*. Find detailed information about SuSEfirewall2 in Chapter 43, *Masquerading and Firewalls* (page 823).

TIP: Automatic Activation of the Firewall

YaST automatically starts a firewall with suitable settings on every configured network interface. Start this module only if you want to reconfigure the firewall with custom settings or deactivate it.

8.10 Virtualization

Virtualization makes it possible to run several operating systems on one physical machine. The hardware for the different systems is provided virtually. Virtualization YaST modules provide configuration for the Xen virtualization system. For more information about this technology, see the virtualization manual on http://www.novell.com/documentation/sles10/index.html..

The following modules are available in the *Virtualization* section:

Installing Hypervisor and Tools
: Before you start using Xen, install a kernel with Xen support and related tools. To install them, use *Virtualization* > *Install Hypervisor and Tools*. After installation reboot your system to use the Xen kernel.

Creating Virtual Machines
: After you successfully installed the Xen hypervisor and tools, you can install virtual machines on your virtual server. To install a virtual machine, use *Virtualization* > *Create Virtual Machines*.

8.11 Miscellaneous

The YaST Control Center has several modules that cannot easily be classified into the first six module groups. They can be used for things like viewing log files and installing drivers from a vendor CD.

8.11.1 Custom Installation CD Creation

With *Miscellaneous* > *CD Creator*, you can create a customized installation CD from your original installation set. To start creation, click *Add*. Use the package manager to select the packages or an AutoYaST control file to use a preconfigured AutoYaST profile for creation.

8.11.2 Installation Server Configuration

For network installation, an installation server is required. To configure such a server, use *Miscellaneous* > *Installation Server*. Find more information about the configuration of an installation server with YaST in Section 4.2.1, “Setting Up an Installation Server Using YaST” (page 56).

8.11.3 Autoinstallation

The AutoYaST tool is intended for automated installation. In *Miscellaneous* > *Autoinstallation*, prepare profiles for this tool. Find detailed information about automated installation with AutoYaST in Chapter 5, *Automated Installation* (page 85). The information about using the *Autoinstallation* module is in Section 5.1.1, “Creating an AutoYaST Profile” (page 86).

8.11.4 Support Query

Miscellaneous > *Support Query* offers the possibility to collect all system information needed by the support team to find your problem so you can get help to solve it as soon is possible. Regarding your query, select the problem category in the following window. When all information is gathered, attach it to your support request.

8.11.5 Release Notes

The release notes are an important source about installation, update, configuration, and technical issues. The release notes are continuously updated and published through online update. Use *Miscellaneous* > *Release Notes* to view the release notes.

8.11.6 Start-Up Log

View information concerning the start-up of the computer in *Miscellaneous* > *Start-Up Log*. This is one of the first places you might want to look when encountering problems with the system or when troubleshooting. It shows the boot log `/var/log/boot.msg`, which contains the screen messages displayed when the computer starts. Viewing the log can help determine if the computer started properly and if all services and functions were started correctly.

8.11.7 System Log

Use *Miscellaneous* > *System Log* to view the system log that keeps track of the operations of your computer in `var/log/messages`. Kernel messages, sorted according to date and time, are also recorded here. View the status of certain system components using the box at the top. The following options are possible from the system log and boot log modules:

/var/log/messages
: This is the general system log file. Here, view kernel messages, users logging in as `root`, and other useful information.

/proc/cpuinfo
: This displays processor information, including its type, make, model, and performance.

/proc/dma
: This shows which DMA channels are currently being used.

/proc/interrupts
: This shows which interrupts are in use and how many of each have been in use.

/proc/iomem
: This displays the status of input/output memory.

/proc/ioports
: This shows which I/O ports are in use at the moment.

/proc/meminfo
: This displays memory status.

/proc/modules
: This displays the individual modules.

/proc/mounts
: This displays devices currently mounted.

/proc/partitions
: This shows the partitioning of all hard disks.

/proc/version
: This displays the current version of Linux.

/var/log/YaST2/y2log
: This displays all YaST log messages.

/var/log/boot.msg
: This displays information concerning the start-up of the system.

/var/log/faillog
: This displays login failures.

/var/log/warn
: This displays all system warnings.

8.11.8 Vendor Driver CD

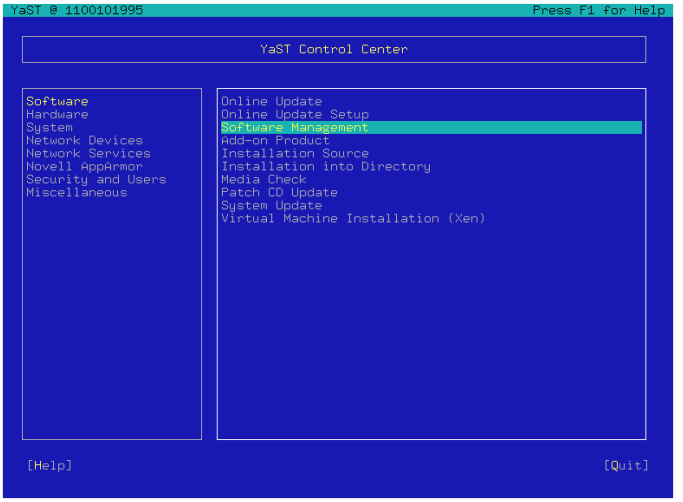
Install device drivers from a Linux driver CD that contains drivers for SUSE Linux Enterprise with *Miscellaneous* > *Vendor Driver CD*. When installing SUSE Linux Enterprise from scratch, use this YaST module to load the required drivers from the vendor CD after the installation.

8.12 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.

When YaST is started in text mode, the YaST Control Center appears first. See Figure 8.9, “Main Window of YaST in Text Mode” (page 185). 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 *Exit*.

Figure 8.9 *Main Window of YaST in Text Mode*



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.

8.12.1 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 8.12.2, "Restriction of Key Combinations" (page 187) 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 8.10 *The Software Installation Module*



8.12.2 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.

8.13 Managing YaST from the Command Line

When a task only needs to be done once, the graphical or ncurses interface is usually the best solution. If a task needs to be done repeatedly, it might be easier to use the YaST command line interface. Custom scripts can also use this interface for automating tasks.

View a list of all module names available on your system with `yast -l` or `yast --list`. To display the available options of a module, enter `yast` *`module_name`* `help`. If a module does not have a command line mode, a message informs you of this.

To display help for a module's command options, enter `yast` *`module_name command`* `help`. To set the option value, enter `yast` *`module_name command option=value`*.

Some modules do not support the command line mode because command line tools with the same functionality already exist. The modules concerned and the command line tools available are:

sw_single
: `sw_single` provides package management and system update functionality. Use `rug` instead of YaST in your scripts. Refer to Section 9.1, “Update from the Command Line with rug” (page 200).

online_update_setup
: `online_update_setup` configures automatic updating of your system. This can be configured with `cron`.

inst_suse_register
: With `inst_suse_register`, register your SUSE Linux Enterprise. For more information about the registration, see Section 8.3.4, “Registering SUSE Linux Enterprise” (page 140).

hwinfo
: `hwinfo` provides information about the hardware of your system. The command `hwinfo` does the same.

GenProf, LogProf, SD_AddProfile, SD_DeleteProfile, SD_EditProfile, SD_Report, and subdomain

These modules control or configure AppArmor. AppArmor has its own command line tools.

8.13.1 Managing Users

The YaST commands for user management, unlike traditional commands, considers the configured authentication method and default user management settings of your system when creating, modifying, or removing users. For example, you do not need create home directory or copy `skel` files during or after the user addition. If you enter the username and password, all other settings are made automatically in accordance with default configuration. The functionality provided by the command line is the same as in the graphical interface.

The YaST module `users` is used for user management. To display the command options, enter `yast users help`.

To add multiple users, create a `/tmp/users.txt` file with a list of users to add. Enter one username per line and use the following script:

Example 8.2 *Adding Multiple Users*

```
#!/bin/bash
#
# adds new user, the password is same as username
#

for i in `cat /tmp/users.txt`;
do
 yast users add username=$i password=$i
done
```

Similarly to adding, you can delete users defined in `/tmp/users.txt`:

Example 8.3 *Removing Multiple Users*

```
#!/bin/bash
#
# the home will be not deleted
# to delete homes, use option delete_home
#

for i in `cat /tmp/users.txt`;
do
yast users delete username=$i
done
```

8.13.2 Configuring the Network and Firewall

Network and firewall configuration commands are often wanted in scripts. Use `yast lan` for network configuration and `yast firewall`.

To display the YaST network card configuration options, enter `yast lan help`. To display the YaST firewall card configuration options, enter `yast firewall help`. The network and firewall configurations with YaST are persistent. After reboot, it is not necessary to execute scripts again.

To display a configuration summary for the network, use `yast lan list`. The first item in the output of Example 8.4, "Sample Output of `yast lan list`" (page 190) is a device ID. To get more information about the configuration of the device, use `yast lan show id=<number>`. In this example, the correct command is `yast lan show id=0`.

Example 8.4 *Sample Output of yast lan list*

```
0       Digital DECchip 21142/43, DHCP
```

The command line interface of the YaST firewall configuration is a fast and easy way to enable or disable services, ports, or protocols. To display allowed services, ports, and protocols, use `yast firewall services show`. For examples of how to enable a service or port, use `yast firewall services help`. To enable masquerading, enter `yast firewall masquerade enable`.

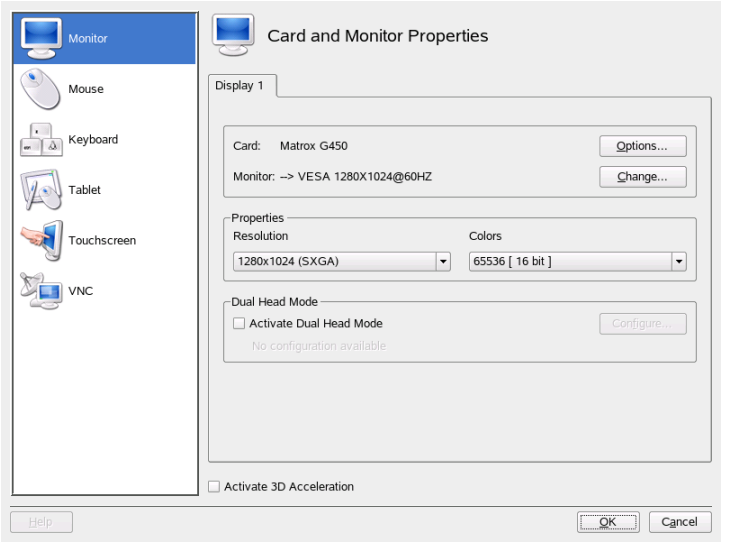
8.14 SaX2

Configure the graphical environment of your system with *Hardware* > *Graphics Card and Monitor*. This opens the SUSE Advanced X11 Configuration interface (SaX2), where you can configure devices such as your mouse, keyboard, or display devices. This interface can also accessed from the GNOME main menu with *Computer* > *More Applications* > *System* > *Sax2* or the KDE main menu with *System* > *Configuration* > *SaX2*.

8.14.1 Card and Monitor Properties

Adjust the settings for your graphics card and display device in *Card and Monitor Properties*. If you have more than one graphics card installed, each device is shown in a separate dialog reachable by a tab. At the top of the dialog, see the current settings for the selected graphics card and the monitor that is attached to it. If more than one screen can be connected to the card (dual head), the monitor on the primary output is shown. Normally, the card and display device are detected automatically by the system during installation. However, you can tune many parameters manually or even change the display device completely.

Figure 8.11 *Card and Monitor Properties*



TIP: Autodetecting New Display Hardware

If you change your display hardware after installation, use `sax2 -r` on the command line to cause SaX2 to detect your hardware. You must be `root` to run SaX2 from the command line.

Graphics Card

It is not possible to change the graphics card because only known models are supported and these are detected automatically. However, you can change many options that affect the behavior of the card. Normally, this should not be necessary because the system already has set them up appropriately during installation. If you are an expert and want to tweak some of the options, click *Options* next to the graphics card and select the option to change. To assign a value needed to a certain option, enter this value in the dialog that appears after selecting that option. Click *OK* to close the options dialog.

Monitor

To change the current settings for the monitor, click *Change* next to the monitor. A new dialog opens in which to adjust various monitor-specific settings. This dialog has several tabs for various aspects of monitor operation. Select the first tab to manually select the vendor and model of the display device in two lists. If your monitor is not listed, you can choose one of the VESA or LCD modes that suit your needs or, if you have a vendor driver disk or CD, click *Utility Disk* and follow the instructions on the screen to use it. Check *Activate DPMS* to use display power management signaling. *Display Size*, with the geometrical properties of the monitor, and *Sync Frequencies*, with the ranges for the horizontal and vertical sync frequencies of your monitor, are normally set up correctly by the system, but you can modify these values manually. After making all adjustments, click *OK* to close this dialog.

WARNING: Changing Monitor Frequencies

Although there are safety mechanisms, you should still be very careful when changing the allowed monitor frequencies manually. Incorrect values might destroy your monitor. You should always refer to the monitor's manual before changing frequencies.

Resolution and Color Depth

The resolution and color depth can be chosen directly from two lists in the middle of the dialog. The resolution you select here marks the highest resolution to use. All common resolutions down to 640x480 are also added to the configuration automatically. Depending on the graphical desktop used, you can switch to any of these later without the need for reconfiguration.

Dual Head

If you have a graphics card with two outputs installed in your computer, you can connect two screens to your system. Two screens that are attached to the same graphics card are referred to as *dual head*. SaX2 automatically detects multiple display devices in the system and prepares the configuration accordingly. To use the dual head mode of a graphics card, check *Activate Dual Head Mode* at the bottom of the dialog and click *Configure* to set the dual head options and the arrangement of the screens in the dual head dialog.

The tabs in the row at the top of the dialog each correspond to a graphics card in your system. Select the card to configure and set its multihead options in the dialog below. In the upper part of the multihead dialog, click *Change* to configure the additional screen. The possible options are the same as for the first screen. Choose the resolution to use for this screen from the list. Select one of three possible multihead modes.

Cloned Multihead
: In this mode, all monitors display the same contents. The mouse is only visible on the main screen.

Xinerama Multihead
: All screens combine to form a single large screen. Program windows can be positioned freely on all screens or scaled to a size that fills more than one monitor.

NOTE

Linux currently does not offer 3D support for Xinerama multihead environments. In this case, SaX2 deactivates the 3D support.

The arrangement of the dual head environment describes the sequence of the individual screens. By default, SaX2 configures a standard layout that follows the sequence of the

detected screens, arranging all screens in a row from left to right. In the *Arrangement* part of the dialog, determine the way the monitors are arranged by selecting one of the sequence buttons. Click *OK* to close the dialog.

TIP: Using a Beamer with Laptop Computers

To connect a beamer to a laptop computer, activate dual head mode. In this case, SaX2 configures the external output with a resolution of 1024x768 and a refresh rate of 60 Hz. These values suit most beamers very well.

Multihead

If you have more than one graphics card installed in your computer, you can connect more than one screen to your system. Two or more screens that are attached to different graphics cards are referred to as *multihead*. SaX2 automatically detects multiple graphics cards in the system and prepares the configuration accordingly. By default, SaX2 configures a standard layout that follows the sequence of the detected graphics cards, arranging all screens in a row from left to right. The additional *Arrangement* tab allows for changing this layout manually. Drag the icons representing the individual screens in the grid and click *OK* to close the dialog.

Testing the Configuration

Click *OK* in the main window after completing the configuration of your monitor and your graphics card, then test your settings. This ensures that your configuration is suitable for your devices. If the image is not steady, terminate the test immediately by pressing Ctrl+Alt+Backspace and reduce the refresh rate or the resolution and color depth.

NOTE

Regardless of whether you run a test, all modifications are only activated when you restart the X server.

8.14.2 Mouse Properties

Adjust the settings for your mouse in *Mouse Properties*. If you have more than one mouse with different drivers installed, each driver is shown in a separate tab. Multiple

devices operated by the same driver are shown as one mouse. Activate or deactivate the currently selected mouse with the check box at the top of the dialog. Below the check box, see the current settings for that mouse. Normally, the mouse is detected automatically, but you can change it manually if the automatic detection fails. Refer to the documentation for your mouse for a description of the model. Click *Change* to select the vendor and model from two lists then click *OK* to confirm your selection. In the options part of the dialog, set various options for operating your mouse.

Activate 3-Button Emulation
: If your mouse has only two buttons, a third button is emulated when you click both buttons simultaneously.

Activate Mouse Wheel
: Check this box to use a scroll wheel.

Invert X-Axis and *Invert Y-Axis*
: If one of these options is selected, the mouse pointer moves in the opposite direction. For touch pads, this feature is sometimes useful.

Emulate Wheel with Mouse Button
: If your mouse does not have a scroll wheel but you want to use similar functionality, you can assign an additional button for this. Select the button to use. While pressing this button, any movement of the mouse is translated into scroll wheel commands. This feature is especially useful with trackballs.

When you are satisfied with your settings, click *OK* to confirm your changes.

NOTE

Any changes you make here take effect only after you restart the X server.

8.14.3 Keyboard Properties

Use this dialog to adjust the settings for operating your keyboard in the graphical environment. In the upper part of the dialog, select the type, language layout, and variant. Use the test field at the bottom of the dialog to check if special characters are displayed correctly. Select additional layouts and variants to use from the list in the middle. Depending on the type of your desktop, these may be switched in the running system

without the need for reconfiguration. After you click *OK*, the changes are applied immediately.

8.14.4 Tablet Properties

Use this dialog to configure a graphics tablet attached to your system. Click the *Graphics Tablet* tab to select vendor and model from the lists. Currently, only a limited number of graphics tablets is supported. To activate the tablet, check *Activate This Tablet* at the top of the dialog.

In the *Port and Mode* dialog, configure the connection to the tablet. SaX2 enables the configuration of graphics tablets connected to the USB port or the serial port. If your tablet is connected to the serial port, verify the port. `/dev/ttyS0` refers to the first serial port. `/dev/ttyS1` refers to the second. Additional ports use similar notation. Choose appropriate *Options* from the list and select the *Primary Tablet Mode* suitable for your needs.

If your graphics tablet supports electronic pens, configure them in *Electronic Pens*. Add eraser and pen and set their properties after clicking *Properties*.

When you are satisfied with the settings, click *OK* to confirm your changes.

8.14.5 Touchscreen Properties

Use this dialog to configure touchscreens attached to your system. If you have more than one touchscreen installed, each device is shown in a separate dialog reachable by a tab. To activate the currently selected touchscreen, check *Assign a Touchscreen to Display* at the top of the dialog. Select vendor and model from the lists below and set an appropriate *Connection Port* at the bottom. You can configure touchscreens connected to the USB port or the serial port. If your touchscreen is connected to the serial port, verify the port. `/dev/ttyS0` refers to the first serial port. `/dev/ttyS1` refers to the second. Additional ports use similar notation. When you are satisfied with your settings, click *OK* to confirm your changes.

8.15 Troubleshooting

All error messages and alerts are logged in the directory `/var/log/YaST2`. The most important file for finding YaST problems is `y2log`.

8.16 For More Information

More information about YaST can be found on the following Web sites and directories:

- `/usr/share/doc/packages/yast2`—Local YaST development documentation
- `http://www.opensuse.org/YaST_Development`—The YaST project page in the openSUSE wiki
- `http://forge.novell.com/modules/xfmod/project/?yast`—Another YaST project page

Managing Software with ZENworks

9

SUSE Linux Enterprise is ready for integration into an environment administrated by Novell ZENworks Linux Management. It includes an open source ZENworks management agent, back-end daemon, and user space software management tools. Novell ZENworks package management tools use a ZENworks Linux Management server to download packages and updates. If no ZENworks Linux Management server is available in your local network, your system can get updates from the Novell Customer Center, which is described in Section 3.14.4, "Novell Customer Center Configuration" (page 40).

The back-end daemon for the Novell ZENworks Linux Management Agent is the ZENworks Management Daemon (ZMD). ZMD performs software management functions. The daemon is started automatically during boot.

Check the status of the daemon with `rczmd status`. To start the daemon, enter `rczmd start`. To restart it, use `rczmd restart`. Deactivate it with `rczmd stop`.

ZMD can also be started with special options to control its behavior. To have ZMD always start with some special options permanently, set `ZMD_OPTIONS` in `/etc/sysconfig/zmd` then run `SuSEconfig`. The available options are:

-n, --no-daemon
: Do not run the daemon in the background.

-m, --no-modules
: Do not load any modules.

-s, --no-services
Do not load initial services.

-r, --no-remote
Do not start remote services.

ZMD configuration is stored in `/etc/zmd/zmd.conf`. You can change the configuration manually or with `rug`. The URL for the ZENworks service that zmd uses at initial start-up and a registration key are stored in `/var/lib/zmd`. Updates are downloaded to the ZMD cache in `/var/cache/zmd`.

ZMD is the back-end only. The software management tasks are initiated through the command line tool `rug` or the graphical Software Updater applet.

9.1 Update from the Command Line with rug

`rug` uses the `zmd daemon` to install, update, and remove software according to the commands given. It can either install software from local files, or from servers. You may use one or more remote servers, known as services. Supported services are `mount` for local files, and `yum` or `ZENworks` for servers.

`rug` sorts software from services into catalogs (also known as channels), which correspond to groups of similar software. For example, one catalog might contain software from an update server as well as software from a third-party software vendor. You are able to subscribe to individual catalogs in order to control the display of available packages and prevent the accidental installation of unwanted software. Operations usually are performed only on software from catalogs to which you are subscribed.

9.1.1 Obtaining Information from rug

`rug` provides a wide range of useful information. It allows you to check the status of `zmd`, view registered services and catalogs, or see information about available patches.

If the `zmd` is not used for a certain period of time, it can be switched to sleep mode. To check the zmd status and reactivate the daemon, use `rug ping`. This command wakes `zmd` up and logs its status information.

To see your registered services, use `rug sl`, and to see which services are supported on your system, use `rug st`.

To check for new patches, use `rug pch`. To get information about a patch, enter `rug patch-info` *`patch`*.

9.1.2 Subscribing to rug Services

By default, a newly installed system is subscribed to several services. To add a new service, use `rug sa` *`URI service_name`*. Replace *`service_name`* with a meaningful and unique string that identifies the new service.

NOTE: Error on Accessing the Update Catalog

If you are not able to access the update catalog, this might be due to an expired subscription. Normally, SUSE Linux Enterprise comes with a one or three years subscription, during which you have access to the update catalog. This access will be denied once the subscription ends.

In case of an access denial to the update catalog you will see a warning message with a recommendation to visit the Novell Customer Center and check your subscription. The Novell Customer Center is available at `http://www.novell.com/center/`.

9.1.3 Installing and Removing Software with rug

To install a package from any subscribed catalogs, use `rug in` *`package_name`*. To install from a selected catalog only, use `-c` *`catalog name`*. Get more information about a package with `rug if` *`package_name`*.

To remove a package, use `rug rm package_name`. If other packages depend on this package, `rug` displays their names, versions, and types. Confirm if you want to remove the package anyway.

9.1.4 rug User Management

One the main advantages of `rug` is its user management. Normally, only `root` can update or install new packages. With `rug`, you can assign the right to update the system to other users and restrict them, for example, to only updating without the possibility to remove software. Privileges you can grant are:

install
: The user may install new software

lock
: The user may set package locks

remove
: The user may remove software

subscribe
: The user may change channel subscriptions

trusted
: The user is considered as trusted, so he is able to install packages without package signatures

upgrade
: The user may update software packages

view
: This allows the user to see which software is installed on the machine and which software is in available channels. The option is relevant only to remote users. Local users are normally permitted to view installed and available packages.

superuser
: Permits all rug commands except user management and settings, which must be done locally.

To give a user permission to update the system, use the `rug ua username upgrade` command. Replace *username* by the name of the user. To revoke the privileges of a user, use command `rug ud username`. To list users with their rights, use `rug ul`.

To change the current privileges of a user, use `rug ue username` and replace the *username* by the name of the desired user. You get a list with the rights of the selected user. The `edit` command is interactive. Use plus (+) or minus (-) to add or remove the user's privileges and press Enter. For example, to permit the user to delete software, enter `+remove`. To save and quit, press Enter at a blank prompt.

9.1.5 Scheduling Updates

Using `rug`, the system can be updated automatically (for example, by scripts). The simplest example is a fully automatic update. To do this, configure a cron job as `root` that executes `rug up -y`. The `up -y` option downloads and installs the patches from your catalogs without confirmation.

However, you may not want the patches to be installed automatically, but may want to retrieve them and select the patches for installation at a later time. To download patches only, use the `rug up -dy` command. The `up -dy` option downloads the patches from your catalogs without confirmation and saves them to the `rug` cache. The default location of the `rug` cache is `/var/cache/zmd`.

9.1.6 Configuring rug

`rug` allows you to customize its setup via a set of preferences. Some of them are pre-configured during installation. Use `rug get` command to get a list of the preferences available. To edit a preference, enter `rug set preference`. For example, adjust settings if you need to update your system through a proxy. Before downloading the updates, send your username and password to the proxy server. To do so, use the following commands:

```
rug set proxy-url url_path
rug set proxy-username name
rug set proxy-password password
```

Replace *url_path* by the name of your proxy server. Replace *name* by your username. Replace *password* by your password.

9.1.7 For More Information

For more information about updating from the command line, enter `rug --help` or see the `rug(1)` man page. The `--help` option is also available for all rug commands. If, for example, you need help for `rug update`, enter `rug update --help`.

9.2 Managing Packages with the ZEN Tools

The ZEN tools serve as graphical front-ends for the ZENworks Management Daemon (zmd), allowing you easily to install or remove software, apply security updates, and manage services and catalogs with just a few clicks.

9.2.1 Getting Permissions

Managing packages on a Linux system requires `root` privileges. The ZEN tools and `rug` have their own user management system that allows users to install software updates. When a user first invokes an action that requires special privileges in the ZEN tools, a prompt for the `root` password appears. When the password has been verified, The ZEN tools automatically add the user's account to the user management system with update permissions. To review or change these settings, use the `rug` user management commands (see Section 9.1.4, “rug User Management” (page 202) for information).

9.2.2 Obtaining and Installing Software Updates

Software Updater resides in the notification area (GNOME) or the system tray (KDE) of your panel as an icon depicting a globe. It changes color and appearance depending on the availability of a network link and new updates. Once a day, Software Updater automatically checks whether updates for your system are available (right-click the application icon and choose *Refresh* to force an immediate check). The Software Updater applet in the panel changes from a globe to an exclamation mark on an orange background when new updates are available.

NOTE: Error on Accessing the Update Catalog

If you are not able to access the update catalog, this might be due to an expired subscription. Normally, SUSE Linux Enterprise comes with a one or three years subscription, during which you have access to the update catalog. This access will be denied once the subscription ends.

In case of an access denial to the update catalog you will see a warning message with a recommendation to visit the Novell Customer Center and check your subscription. The Novell Customer Center is available at `http://www.novell.com/center/`.

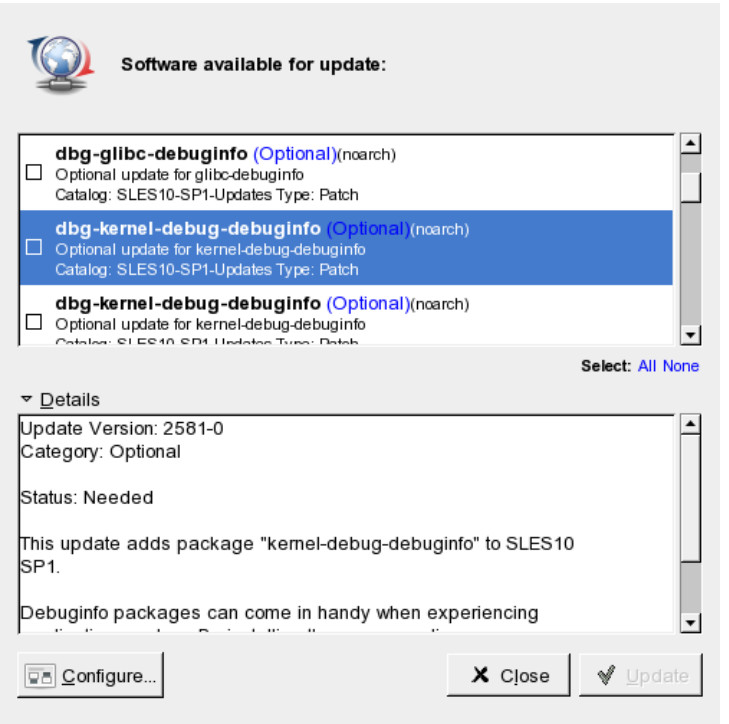
Left-click the panel icon to open the updater window. It displays a list of patches and new package versions available. Each entry has a short description and, if applicable, a category icon: Security patches are marked with a yellow shield. Optional patches are marked with a light blue circle. Recommended patches are not marked with an icon. Security patches are listed first, then recommended patches, optional patches, and finally new package versions. Use the links *All*, *Packages*, and *Patches* to filter the list of packages displayed.

NOTE: Packages versus Patches

Officially released updates from Novell show up as *Patches*. New package versions from other sources show up as *Packages*.

To get details about a certain entry, mark it with the mouse and click the *Details* link under the list window. To select an entry for installation, mark the entry's check box. Use the links *All* and *None* to select or deselect all patches. Clicking *Update* installs the selected programs.

Figure 9.1 *Selecting the Software Updates*



9.2.3 Installing Software

To install software packages, start *Install Software* from the menu or run `zen-installer`. The interface is almost identical to Software Updater (see Section 9.2.2, “Obtaining and Installing Software Updates” (page 204)). The only difference is a search panel you can use to search for packages or to filter the list. Mark the check box of packages that should be installed then press *Install* to start the package installation. Possible dependencies on other packages are automatically resolved by the installer.

9.2.4 Remove Software

Start *Remove Software* from the menu or run `zen-remover` to uninstall software packages. The list of packages can be narrowed with the links *Products* (uninstalls the

complete products), *Patterns* (see Section "Installing and Removing Patterns" (page 133) for details on patterns), *Packages*, and *Patches*. Mark the check box of a list entry that should be removed then press *Remove* to start the package uninstallation. If other packages depend on the ones marked by you, these are also removed. You must confirm the removal of additional packages. If you click *Cancel* in the confirmation dialog, no packages are uninstalled.

9.2.5 Configuring the Software Updater

To configure the ZEN tools, click *Configure* in the application window. A window with three tabs opens: *Services*, *Catalogs*, and *Preferences*.

Services and Catalogs

Services are basically sources that provide software packages and information about these packages. Each service can offer one or more catalogs.

The service tab lists all services available together with type and status information (if you cannot see the latter two, adjust the window size). Use *Remove Service* or *Add Service* to add or remove services. The following service types are available:

YUM
: An HTTP, HTTPS, or FTP server using the RPM-MD format for the package data.

ZYPP
: ZYPP services are the YaST installation sources added with *Software > Installation Source* in YaST. Use Software Updater or YaST to add installation sources. The source from which you initially installed (DVD or CD-ROM in most cases) is preconfigured. If you change or delete this source, replace it with another valid installation source (ZYPP service), because otherwise you cannot install new software.

NOTE: Terminology

The terms YaST installation source, YaST package repository, and ZYPP service are the same name for a source from which you can install software.

Mount
: With *Mount*, embed a directory mounted on your machine. This is useful, for example, in a network that regularly mirrors the Novell YUM server and exports its content to the local network. To add the directory, provide the full path to the directory in *Service URI*.

NU
: NU stands for Novell Update. Novell provides updates for SUSE Linux Enterprise exclusively as NU services. If you configured update during installation, the official Novell NU server is already present in the list.

  If you skipped the update configuration during installation, run `suse_register` on the command line or the YaST module *Software* > *Product Registration* as user `root`. The Novell Update server is automatically added to Software Updater.

RCE and ZENworks
: Opencarpet, Red Carpet Enterprise, or ZENworks services are only available if your company or organization has set up these services within your internal network. This may, for example, be the case if your organization is using third-party software for which updates are deployed on a single server.

After SUSE Linux Enterprise is installed, two services are preconfigured: your installation source (DVD, CD-ROM, or network resource) as a ZYPP service and a SUSE Linux Enterprise update server as a NU service, which was added during product registration. Normally there is no need to change these settings. If you do not see a NUYUM service, open a `root` shell and execute the command `suse_register`. A service is added automatically.

Catalogs

Services are able to provide packages for different pieces of software or for different software versions (typically RCE or ZENworks services do so). These are organized in different categories called *catalogs*. Subscribe or unsubscribe from a catalog by marking or unmarking the check box in front of it.

At the moment, the SUSE Linux services (YUM and ZYPP) do not provide different catalogs. Each service only has one catalog. If Software Updater was configured during installation or with `suse_register`, it subscribes to the YUM and ZYPP catalogs automatically. If you manually add a service, you must subscribe to its catalogs.

Preferences

On the *Preferences* tab, specify whether Software Updater should be launched at start-up or not. As user `root`, you can also modify the Software Updater settings. As an unprivileged user, you can only view the settings. Refer to the `rug` man page for an explanation of the settings.

9.3 For More Information

Find more information about ZENworks Linux Management and ZMD at `http://www.novell.com/products/zenworks/linuxmanagement/index.html`.

Updating SUSE Linux Enterprise

10

SUSE® Linux Enterprise provides the option of updating an existing system to the new version without completely reinstalling it. No new installation is needed. Old data, such as home directories and system configuration, is kept intact. During the life cycle of the product, you can apply Service Packs to increase system security and correct software defects. Install from a local CD or DVD drive or from a central network installation source.

10.1 Updating SUSE Linux Enterprise

Follow the steps outlined in this section, if you want to update from SUSE Linux Enterprise Server 9 to SUSE Linux Enterprise Server 10, for example. You can also follow these steps if you want to update from SUSE Linux Enterprise 10 SP1 to SUSE Linux Enterprise 10 SP2.

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 updating 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 and the software selected.

10.1.1 Preparations

Before updating, copy the old configuration files to a separate medium, such as tape device, 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 10.1, “List with df -h” (page 212), the root partition to write down is /dev/hda3 (mounted as /).

Example 10.1 *List with df -h*

```
Filesystem     Size  Used Avail Use% Mounted on
/dev/hda3       74G   22G   53G  29% /
tmpfs          506M     0  506M   0% /dev/shm
/dev/hda5      116G  5.8G  111G   5% /home
/dev/hda1       44G    4G   40G   9% /data
```

10.1.2 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. Check the following issues 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.

10.1.3 Updating with YaST

Following the preparation procedure outlined in Section 10.1.1, “Preparations” (page 211), you can now update your system:

1. Optionally, prepare an installation server. For background information, see Section 4.2.1, “Setting Up an Installation Server Using YaST” (page 56).

2. Boot the system as for the installation, described in Section 3.3, “System Start-Up for Installation” (page 18). In YaST, choose a language and select *Update* in the *Installation Mode* dialog. Do not select *New Installation*.

3. 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/hda3` was selected in the example in Section 10.1.1, “Preparations” (page 211)). YaST reads the old `fstab` on this partition to analyze and mount the file systems listed there.

4. 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.

   4a Click *Update Options* to update only software that is already installed (*Only Update Installed Packages*) or to add new software and features to the system according to selected patterns. It is advisable to accept the suggestion. You can adjustment it later with YaST.

   4b You also have the possibility to make backups (*Backup*) of various system components. Selecting backups slows down the update process. Use this option if you do not have a recent system backup.

5. Click *Accept* and confirm *Start Update* to start the software installation process.

At the end of the installation read the release notes and then click *Finish* to restart the computer and log in.

10.2 Installing Service Packs

Use Service Packs to update a SUSE Linux Enterprise installation. There are several different ways in which you can apply a Service Pack. You can either update the existing installation or start a whole new installation using the Service Pack media. Possible scenarios for updating the system and setting up a central network installation source are described here.

TIP: Installation Changes

Read the installation instructions on the Service Pack media for further changes.

10.2.1 Setting Up a Network Installation Source for Service Pack Media

As with the initial installation of SUSE Linux Enterprise, it is much more efficient having a central installation source on your network to serve all clients rather than installing all of them separately using a set of physical media.

Configuring a Network Installation Source on SUSE Linux Enterprise Using YaST

Basically, follow the procedure outlined in Section 4.2, “Setting Up the Server Holding the Installation Sources” (page 56). Just add another installation source called SLE-10-SP-*x*-*arch*, SLES-10-SP-*x*-*arch*, or SLED-10-SP-*x*-*arch* (where *x* is the number of the Service Pack and *arch* is the name of your hardware architecture) and make it available via NFS, HTTP, or FTP.

10.2.2 Installing a Service Pack

NOTE

To update an existing SUSE Linux Enterprise 10 system to a SUSE Linux Enterprise 10 Service Pack (SP), see Section 10.2.3, “Updating to a Service Pack” (page 217).

Installing a SUSE Linux Enterprise Service Pack is very similar to installing the original SUSE Linux Enterprise media. As with the original installation, you can choose to install from a local CD or DVD drive or from a central network installation source.

Installing from a Local CD or DVD Drive

Before starting a new installation of a SUSE Linux Enterprise SP, ensure that all of the Service Pack installation media (CDs or DVD) are available.

Procedure 10.1 *Booting from the Service Pack Medium*

1. Insert the first SUSE Linux Enterprise SP medium (CD 1 or DVD 1) and boot your machine. A boot screen similar to the original installation of SUSE Linux Enterprise 10 is displayed.

2. Select *Installation* and continue as outlined in the YaST installation instructions in Chapter 3, *Installation with YaST* (page 17).

Network Installation

Before starting a network installation of an SUSE Linux Enterprise SP, make sure that the following requirements are met:

- A network installation source set up according to Section 10.2.1, “Setting Up a Network Installation Source for Service Pack Media” (page 214).

- A working network connection both on the installation server and the target machine that includes a name service, DHCP (optional, but needed for PXE boot), and OpenSLP (optional).

- The SUSE Linux Enterprise SP CD 1 or DVD 1 to boot the target system *or* a target system set up for PXE boot according to Section 4.3.5, “Preparing the Target System for PXE Boot” (page 75).

Network Installation—Boot from CD or DVD

To perform a network installation using the SP CD or DVD as the boot medium, proceed as follows:

1. Insert the SUSE Linux Enterprise SP CD 1 or DVD 1 and boot your machine. A boot screen similar to the original installation of SUSE Linux Enterprise 10 is displayed.

2. Select *Installation* to boot the SP kernel then use F3 to select a type of network installation source (FTP, HTTP, NFS, or SMB).

3. Provide the appropriate path information or select *SLP* as the installation source.

4. Select the appropriate installation server from those offered or use the boot options prompt to provide the type of installation source and its actual location as in Section 3.3.4, “Installing from a Network Source without SLP” (page 20). YaST starts.

   Finish the installation as outlined in Chapter 3, *Installation with YaST* (page 17).

Network Installation—PXE Boot

To perform a network installation of a SUSE Linux Enterprise Service Pack via network, proceed as follows:

1. Adjust the setup of your DHCP server to provide the address information needed for PXE boot according to Section 4.3.5, “Preparing the Target System for PXE Boot” (page 75).

2. Set up a TFTP server to hold the boot image needed for PXE boot.

   Use the first CD or DVD of your SUSE Linux Enterprise Service Pack for this and otherwise follow the instructions in Section 4.3.2, “Setting Up a TFTP Server” (page 68).

3. Prepare PXE boot and Wake-on-LAN on the target machine.

4. Initiate the boot of the target system and use VNC to remotely connect to the installation routine running on this machine. See Section 4.5.1, “VNC Installation” (page 81) for more information.

5. Accept the license agreement then select a language, default desktop, and other installation settings.

6. Click *Yes, Install* to start the installation.

7 Continue as usual with the installation (entering a password for `root`, completing the network configuration, testing your Internet connection, activating the Online Update service, selecting the user authentication method, and entering a username and password).

For detailed instructions for installing SUSE Linux Enterprise, see Chapter 3, *Installation with YaST* (page 17).

10.2.3 Updating to a Service Pack

There are two preferred ways to update the system to the Service Pack (SP) feature level. One way is to boot from the SP medium. The alternative is to run YaST Online Update or zen-updater and select the *Update to Service Pack X* patch. By updating to the new feature level, additional features like new drivers or software enhancements are available to your system.

WARNING: Do not miss the *Update to Service Pack* patch

If you do not select the *Update to Service Pack* patch, your system will stay at the previous feature level and you will get bug fixes and security updates for a limited time only (for SUSE Linux Enterprise 10 SP2 this period is now extended to six months). Thus for ongoing system integrity it is recommended to switch to the new feature level as early as possible.

Other update ways are running `rug` commands manually, using the Patch CD (see Section 8.3.7, “Updating from a Patch CD” (page 144)), or making use of a locally installed SMT system.

NOTE

On s390 systems, the Patch CD update option is not available.

Booting from the SP Medium for the Update

Boot from the SP medium and choose *Update* as the installation mode in YaST. For more detailed information and finishing the update, see Section 10.1.3, “Updating with YaST” (page 213).

Starting with YaST Online Update

Before initiating the YaST Online Update to update to the SP feature level, make sure that the following requirements are met:

- The system must be online throughout the entire update process, because this process requires access to the Novell Customer Center.
- If your setup involves third party software or add-on software, test this procedure on another machine to make sure that the dependencies are not broken by the update.
- Make sure that the entire process is completed successfully. Otherwise the system becomes inconsistent.

You can only update to Service Pack 2, if Service Pack 1 is fully installed beforehand. If this is not already the case, update to Service Pack 1 first as outlined in Section “SUSE Linux Enterprise GA to SP1 and SP2” (page 223).

Figure 10.1 *Service Pack 1 Package Management Update*

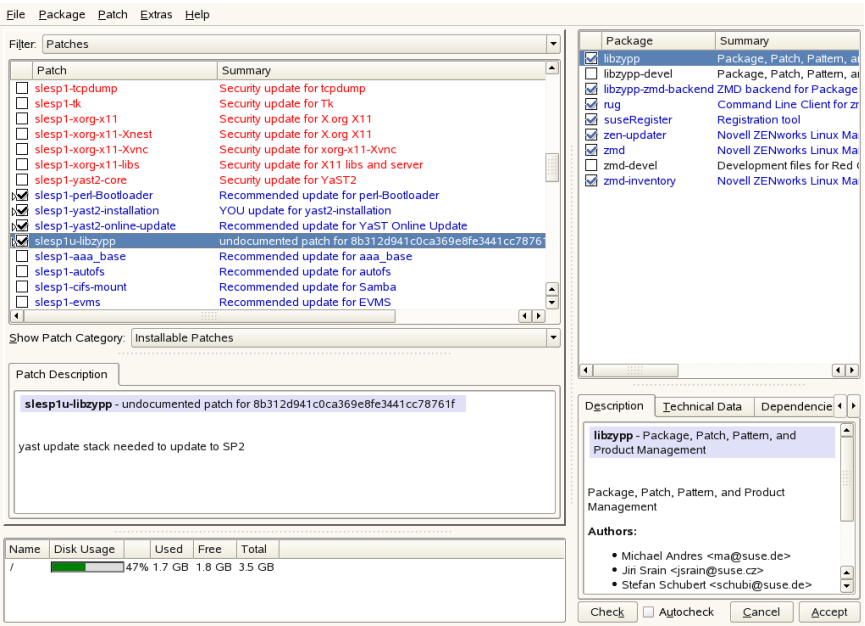
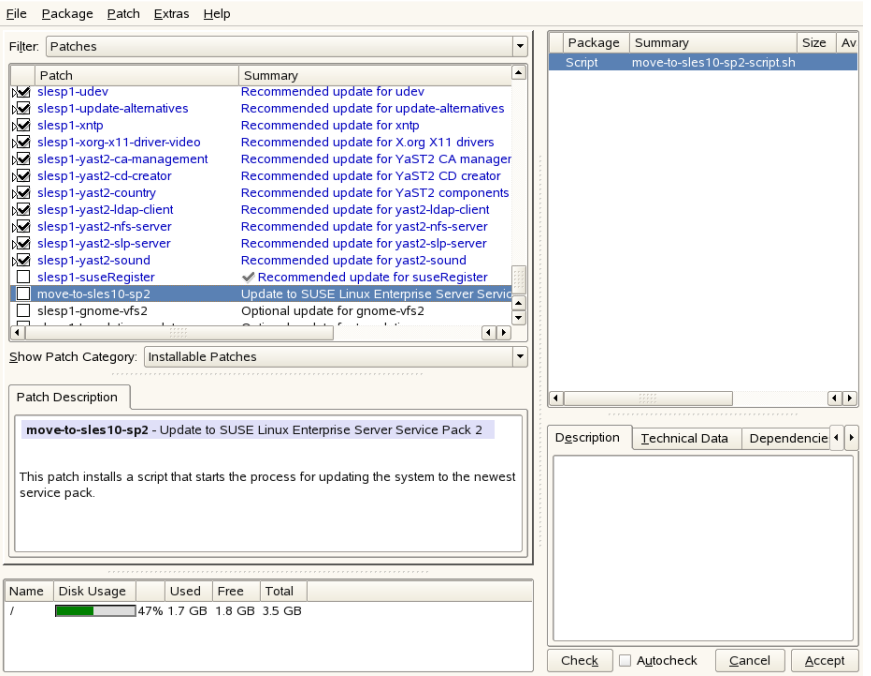


Figure 10.2 *Update to Service Pack 2*



NOTE

During update migration using YaST Online Update, the ZMD stack is updated and the ZMD daemon is restarted, too. Therefore, it is advisable to avoid using any other software management tools such as `rug`, `zen-updater`, `zen-installer` and `zen-remover`. It is recommended to quit the `zen-updater` during the migration.

1 In a running SUSE Linux Enterprise system, select *Computer* > *YaST* > *Software* > *Online Update*.

   If you are not logged in as `root`, enter the `root` password when prompted.

2 The *Online Update* dialog appears. Several patches are preselected. Scroll down the patch list and verify that the package management related patches and the

SUSE Linux Enterprise 10 SP2 maintenance stack update (`slesp1u-libzypp`) are actually preselected. Then press *Accept* to apply the selected updates.

3. The *Patch Download and Installation* dialog tracks the progress log. When *Total Progress* reaches *100%*, click *Close*. The *Online Update* will then restart automatically.

4. Once restarted, press *Accept* to apply all available updates together with a new kernel. When installed, you must reboot the system.

5. In the restarted *Online Update* you now should scroll down the patch list and select *Update to Service Pack 2* (`move-to-sles10-sp2`) as shown in Figure 10.2, "Update to Service Pack 2" (page 219). In the pop-up window, click *Accept* to confirm the start of the update procedure to the service pack feature level.

   The `move-to-sles10-sp2` patch is marked `optional`. If you do not select it, your system will stay at the SP1 feature level and you will get bug fixes and security updates only for a limited time (six month after the availibility of SP2).

6. The *Patch Download and Installation* dialog tracks the progress log of the migration patch installation. When *Total Progress* reaches *100%*, click *Finish*.

7. Start YaST Online Updates again. Apply the `product-sles10-sp2` and `slesp2o-sp2_online` patch to bring the system to the SP2 level. Both these patches are preselected because they are mandatory, if you installed `move-to-sles10-sp2` during the steps before.

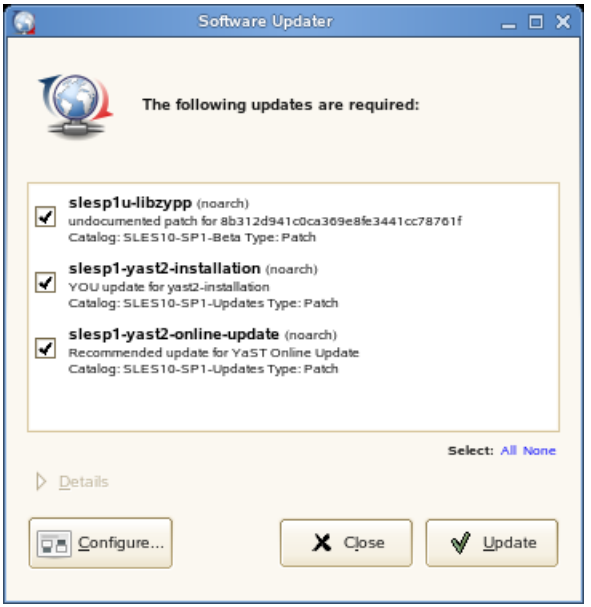
8. Click *Close* to finish the update to SUSE Linux Enterprise 10 SP2 and reboot.

Starting with zen-updater

For background information about ZENworks, see Chapter 9, *Managing Software with ZENworks* (page 199).

Before initiating the online update using zen-updater to progress to the SP feature level, make sure that the requirements are met as listed in Section "Starting with YaST Online Update" (page 218).

Figure 10.3 *Apply SLE10 SP2 Maintenance Stack Update*



1 In a running SUSE Linux Enterprise system, start the zen-updater by clicking the updater icon at the bottom.

TIP: Waking up ZMD

If you see the *ZMD not running* message, check in a terminal as `root` with `rczmd status` whether ZMD is alive. In case of trouble enter `rug restart --clean` to force a restart and cleanup of ZMD and its database.

If you are not logged in as `root`, enter the `root` password when prompted.

2 Apply all maintenance updates available for your system.

3 Apply the SLE10 SP2 maintenance stack update (`slesp1u-libzypp`). These items should be preselected and clicking *Update* should initiate this step. After resolving all dependencies click *Apply*. Confirm the message pop-up by clicking *Close* when finished.

4 In the restarted Software Updater, page down and select the optional `move-to-sles10-sp2` patch and apply it. If you do not select it, your system will stay at the SP1 feature level and you will get bug fixes and security updates only for a limited time (six month after the availibility of SP2).

5 In the Software Updater apply the `product-sles10-sp2` and `slesp2o-sp2_online` patch to bring the system to the SP2 level. Both these patches are mandatory, if you installed `move-to-sles10-sp2` during the steps before, and therefore are preselected.

6 Click *Close* to finish the update to SUSE Linux Enterprise 10 SP2 and reboot.

Using rug

For background information about `rug` command line tool, see Section 9.1, "Update from the Command Line with rug" (page 200). Use `rug`, if you need a scriptable solution for the update.

Before initiating the online update using `rug` to progress to the SP feature level, make sure that the requirements are met as listed in Section "Starting with YaST Online Update" (page 218).

This it the minimal command sequence needed to migrate the system to the SP2 patch level:

```
rug in -t patch slesp1u-libzypp && rug ping -a
rug in -t patch move-to-sles10-sp2 && rug ping -a
rug refresh && rug ping -a
rug up -t patch -g recommended && rug ping -a
reboot
```

NOTE

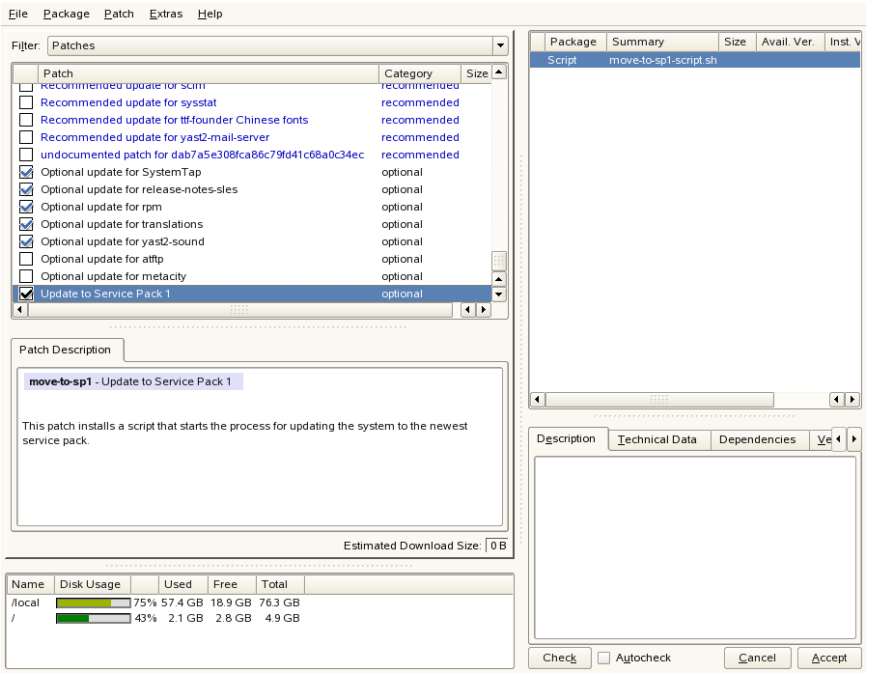
`rug ping -a` ensures, that the ZMD initialization is complete after the previous `rug` command.

SUSE Linux Enterprise GA to SP1 and SP2

NOTE

The following steps are only relevant, if your system is still running at the GA patch level.

Figure 10.4 *Update to Service Pack 1*



1 In a running SUSE Linux Enterprise system (GA), select *Computer* > *YaST* > *Software* > *Online Update*.

   If you are not logged in as `root`, enter the `root` password when prompted.

2 The *Online Update* dialog appears. Scroll down the patch list and select *Update to Service Pack 1* as shown in Figure 10.4, “Update to Service Pack 1” (page 223). In the pop-up window, click *Accept* to confirm the start of the update procedure to the service pack feature level.

3 The *Patch Download and Installation* dialog tracks the progress log of the migration patch installation. When *Total Progress* reaches *100%*, click *Finish*.

4 Run the online update a second time. Once done, in the *Patch Download and Installation* click *Close*. During this second run YaST installs the kernel and all the other software.

5 Click *Finish* when you see *Installation Finished* reported near the end of the progress log.

6 To finish the update, manually reboot the system to activate the new kernel.

Now SUSE Linux Enterprise runs at the SP1 patch level. Continue with Section "Starting with YaST Online Update" (page 218) to promote the system to the SP2 patch level.

10.3 Software Changes from Version 9 to Version 10

The individual aspects changed from version 9 to version 10 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.

NOTE: Software Changes from SLES 10 to SLES 10 SP 1

For a detailed list of software and configuration changes from SUSE Linux Enterprise Server 10 to SUSE Linux Enterprise Server 10 SP1, refer to the release notes of the service pack. View them in the installed system using the YaST release notes module.

10.3.1 Multiple Kernels

It is possible to install multiple kernels side by side. This feature is meant to allow administrators to upgrade from one kernel to another by installing the new kernel, verifying that the new kernel works as expected, then uninstalling the old kernel. While YaST does not yet support this feature, kernels can easily be installed and uninstalled from the shell using `rpm -i package.rpm`.

The default boot loader menus contain one kernel entry. Before installing multiple kernels, it is useful to add an entry for the extra kernels, so they can be selected easily. The kernel that was active before installing the new kernel can be accessed as `vmlinuz.previous` and `initrd.previous`. By creating a boot loader entry similar to the default entry and having this entry refer to `vmlinuz.previous` and `initrd.previous` instead of `vmlinuz` and `initrd`, the previously active kernel can be accessed. Alternatively, GRUB and LILO support wild card boot loader entries. Refer to the GRUB info pages (`info grub`) and to the `lilo.conf`(5) manual page for details.

10.3.2 Changes with Kernel Modules

The following kernel modules are no longer available:

- `km_fcdsl`—AVM Fritz!Card DSL
- `km_fritzcapi`—AVM FRITZ! ISDN Adapters

The following kernel module package was changed internally:

- `km_wlan`—Various drivers for wireless LAN cards. The `madwifi` driver for Atheros WLAN cards from `km_wlan` was removed.

For technical reasons, it was necessary to drop support for Ralink WLAN cards. The following modules were not part of the distribution and will not be added in the future:

- `ati-fglrx`—ATI FireGL Graphics Cards
- `nvidia-gfx`—NVIDIA gfx driver

- km_smartlink-softmodem—Smart Link Soft Modem

10.3.3 Console Number Change and Serial Devices

As of 2.6.10, serial devices on ia64 are named based on the order of ACPI and PCI enumeration. The first device in the ACPI name space (if any) becomes /dev/ttyS0, the second becomes /dev/ttyS1, etc., and PCI devices are named sequentially starting after the ACPI devices.

On HP systems, you must reconfigure the EFI console then you can drop the console parameter from the kernel boot command. As a work-around, you can try console=ttyS1... as a boot parameter instead of console=ttyS0....

Find details in /usr/src/linux/Documentation/ia64/serial.txt, which is part of the kernel-source software package.

10.3.4 LD_ASSUME_KERNEL Environment Variable

Do not set the LD_ASSUME_KERNEL environment variable any longer. In the past, it was possible to use it to enforce LinuxThreads support, which was dropped. If you set LD_ASSUME_KERNEL=2.4.x in SUSE Linux Enterprise 10, everything breaks because ld.so looks for glibc and related tools in a path that does not exist.

10.3.5 Stricter tar Syntax

The tar usage syntax is stricter now. The tar options must come before the file or directory specifications. Appending options, like --atime-preserve or --numeric-owner, after the file or directory specification makes tar fail. Check your backup scripts. Commands such as the following no longer work:

```
tar czf etc.tar.gz /etc --atime-preserve
```

See the tar info pages for more information.

10.3.6 Apache 2 Replaced with Apache 2.2

The Apache Web server (version 2) has been replaced with version 2.2. For Apache version 2.2, Chapter 40, *The Apache HTTP Server* (page 745) 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`.

10.3.7 Kerberos for Network Authentication

`Kerberos` is the default for network authentication instead of `heimdal`. Converting an existing heimdal configuration automatically is not possible. During a system update, backup copies of configuration files are created as shown in Table 10.1, "Backup Files" (page 227).

Table 10.1 *Backup Files*

Old File	Backup File
`/etc/krb5.conf`	`/etc/krb5.conf.heimdal`
`/etc/krb5.keytab`	`/etc/krb5.keytab.heimdal`

The client configuration (`/etc/krb5.conf`) is very similar to the one of heimdal. If nothing special was configured, it is enough to replace the parameter `kpasswd_server` with `admin_server`.

It is not possible to copy the server-related (kdc and kadmind) data. After the system update, the old heimdal database is still available under `/var/heimdal`. MIT kerberos maintains the database under `/var/lib/kerberos/krb5kdc`. For more information, see Chapter 45, *Network Authentication—Kerberos* (page 841) and Chapter 46, *Installing and Administering Kerberos* (page 849).

10.3.8 Hotplug Events Handled by the udev Daemon

Hotplug events are now completely handled by the udev daemon (`udevd`). The event multiplexer system in `/etc/hotplug.d` and `/etc/dev.d` is no longer used. 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.

10.3.9 Firewall Activation During the Installation

To increase security, the enclosed firewall solution SuSEFirewall2 is activated at the end of the installation in the proposal dialog. This means that all ports are closed initially and can be opened in the proposal dialog if necessary. By default, you cannot log in from remote systems. It also interferes with network browsing and multicast applications, such as SLP, Samba ("Network Neighborhood"), and some games. You can fine-tune the firewall settings using YaST.

If network access is required during the installation or configuration of a service, the respective YaST module opens the needed TCP and UDP ports of all internal and external interfaces. If this is not desired, close the ports in the YaST module or specify other detailed firewall settings.

10.3.10 KDE and IPv6 Support

By default, IPv6 support is not enabled for KDE. You can enable it using the `/etc/sysconfig` editor of YaST. The reason for disabling this feature is that IPv6 addresses are not properly supported by all Internet service providers and, as a consequence, this would lead to error messages while browsing the Web and delays while displaying Web pages.

10.3.11 Online Update and Delta Packages

Online Update now supports a special kind of RPM package that only stores the binary difference from a given base package. This technique significantly reduces the package size and download time at the expense of higher CPU load for reassembling the final package. See `/usr/share/doc/packages/deltarpm/README` for technical details.

10.3.12 Print System Configuration

At the end of the installation (proposal dialog), the ports needed for the print system must be open in the firewall configuration. Port 631/TCP and port 631/UDP are needed for CUPS and should not be closed for normal operation. Port 515/TCP (for the old LPD protocol) and the ports used by Samba must also be open for printing via LPD or SMB.

10.3.13 Change to X.Org

The change from XFree86 to X.Org is facilitated by compatibility links that enable access to important files and commands with the old names.

Table 10.2 *Commands*

XFree86	X.Org
`XFree86`	`Xorg`
`xf86config`	`xorgconfig`
`xf86cfg`	`xorgcfg`

Table 10.3 *Log Files in /var/log*

XFree86	X.Org
`XFree86.0.log`	`Xorg.0.log`

XFree86	X.Org
XFree86.0.log.old	Xorg.0.log.old

In the course of the change to X.Org, the packages were renamed from XFree86* to xorg-x11*.

10.3.14 X.Org Configuration File

The configuration tool SaX2 writes the X.Org configuration settings into /etc/X11/xorg.conf. During an installation from scratch, no compatibility link from XF86Config to xorg.conf is created.

10.3.15 XView and OpenLook Support Dropped

The packages xview, xview-devel, xview-devel-examples, olvwm, and xtoolpl were dropped. In the past, only the XView (OpenLook) base system was provided. The XView libraries are no longer provided after the system update. Even more important, OLVWM (OpenLook Virtual Window Manager) is no longer available.

10.3.16 Terminal Emulators for X11

A number of terminal emulators were removed because they are either no longer maintained or do not work in the default environment, especially by not supporting UTF-8. SUSE Linux Enterprise Server offers standard terminals, such as xterm, the KDE and GNOME terminals, and mlterm (Multilingual Terminal Emulator for X), which might be a replacement for aterm and eterm.

10.3.17 OpenOffice.org (OOo)

Directories

OOo is now installed in `/usr/lib/ooo-2.0` instead of `/opt/OpenOffice.org`. The default directory for user settings is now `~/.ooo-2.0` instead of `~/OpenOffice.org1.1`.

Wrapper

There are some new wrappers for starting the OOo components. The new names are shown in Table 10.4, "Wrapper" (page 231).

Table 10.4 *Wrapper*

Old	New
`/usr/X11R6/bin/OOo-calc`	`/usr/bin/oocalc`
`/usr/X11R6/bin/OOo-draw`	`/usr/bin/oodraw`
`/usr/X11R6/bin/OOo-impress`	`/usr/bin/ooimpress`
`/usr/X11R6/bin/OOo-math`	`/usr/bin/oomath`
`/usr/X11R6/bin/OOo-padmin`	`/usr/sbin/oopadmin`
`/usr/X11R6/bin/OOo-setup`	–
`/usr/X11R6/bin/OOo-template`	`/usr/bin/oofromtemplate`
`/usr/X11R6/bin/OOo-web`	`/usr/bin/ooweb`
`/usr/X11R6/bin/OOo-writer`	`/usr/bin/oowriter`
`/usr/X11R6/bin/OOo`	`/usr/bin/ooffice`
`/usr/X11R6/bin/OOo-wrapper`	`/usr/bin/ooo-wrapper`

The wrapper now supports the option `--icons-set` for switching between KDE and GNOME icons. The following options are no longer supported: `--default-configuration`, `--gui`, `--java-path`, `--skip-check`, `--lang` (the language is now determined by means of locales), `--messages-in-window`, and `--quiet`.

KDE and GNOME Support

KDE and GNOME extensions are available in the `OpenOffice_org-kde` and `OpenOffice_org-gnome` packages.

10.3.18 Sound Mixer kmix

The sound mixer kmix is preset as the default. For high-end hardware, there are other mixers, like QAMix. KAMix, envy24control (only ICE1712), or hdspmixer (only RME Hammerfall).

10.3.19 DVD Burning

In the past, a patch was applied to the `cdrecord` binary from the `cdrecord` package to support burning DVDs. Instead, a new binary `cdrecord-dvd` is installed that has this patch.

The `growisofs` program from the `dvd+rw-tools` package can now burn all DVD media (DVD+R, DVD-R, DVD+RW, DVD-RW, DVD+RL). Try using that one instead of the patched `cdrecord-dvd`.

10.3.20 Starting Manual Installation at the Kernel Prompt

The *Manual Installation* mode is gone from the boot loader screen. You can still get linuxrc into manual mode using `manual=1` at the boot prompt. Normally this is not necessary because you can set installation options at the kernel prompt directly, such as `textmode=1` or a URL as the installation source.

10.3.21 JFS: Not Supported Anymore

Due to technical problems with JFS, it is no longer supported. The kernel file system driver is still there, but YaST does not offer partitioning with JFS.

10.3.22 AIDE as a Tripwire Replacement

As an intrusion detection system, use AIDE (package name `aide`), which is released under the GPL. Tripwire is no longer available on SUSE Linux.

10.3.23 PAM Configuration

New Configuration Files (containing comments for more information)

`common-auth`
: Default PAM configuration for auth section

`common-account`
: Default PAM configuration for account section

`common-password`
: Default PAM configuration for password changing

`common-session`
: Default PAM configuration for session management

You should include these default configuration files from within your application-specific configuration file, because it is easier to modify and maintain one file instead of the approximately forty files that used to exist on the system. If you install an application later, it inherits the already applied changes and the administrator is not required to remember to adjust the configuration.

The changes are simple. If you have the following configuration file (which should be the default for most applications):

```
#%PAM-1.0
auth     required       pam_unix2.so
account  required       pam_unix2.so
password required       pam_pwcheck.so
```

```
password required       pam_unix2.so    use_first_pass use_authtok
#password required      pam_make.so     /var/yp
session required        pam_unix2.so
```

you can change it to:

```
#%PAM-1.0
auth     include        common-auth
account  include        common-account
password include        common-password
session  include        common-session
```

10.3.24 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`.

10.3.25 Changes in the powersave Package

The configuration files in `/etc/sysconfig/powersave` have changed:

Table 10.5 *Split Configuration Files in /etc/sysconfig/powersave*

Old	Now Split Into
`/etc/sysconfig/powersave/common`	`common`
	`cpufreq`
	`events`
	`battery`
	`sleep`
	`thermal`

/etc/powersave.conf has become obsolete. Existing variables have been moved to the files listed in Table 10.5, “Split Configuration Files in /etc/sysconfig/powersave” (page 234). If you changed the “event” variables in /etc/powersave.conf, these must now be adapted in /etc/sysconfig/powersave/events.

The names of sleep states have changed from:

- suspend (ACPI S4, APM suspend)
- standby (ACPI S3, APM standby)

To:

- suspend to disk (ACPI S4, APM suspend)
- suspend to ram (ACPI S3, APM suspend)
- standby (ACPI S1, APM standby)

10.3.26 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 28.5.1, “Configuring the powersave Package” (page 520).

10.3.27 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.

10.3.28 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`.

10.3.29 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`

10.3.30 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).

10.3.31 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.

10.3.32 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 need 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`*.

Part II. Administration

11 OpenWBEM

Novell® has embraced the open standard strategies of Web-Based Enterprise Management (WBEM) proposed by the Distributed Management Task Force (DMTF) [http://www.dmtf.org/home]. Implementing these strategies can substantially reduce the level of complexity associated with managing disparate systems in your network.

The following information describes a few of the components proposed by the DMTF standards. Understanding what these are and how they relate to each other can help you understand what OpenWBEM is and how you most effectively use it in your network.

- Web-Based Enterprise Management (WBEM) is a set of management and Internet standard technologies developed to unify the management of enterprise computing environments. WBEM provides the ability for the industry to deliver a well-integrated set of standards-based management tools leveraging the emerging Web technologies. The DMTF has developed a core set of standards that make up WBEM:
  - A data model: the Common Information Model (CIM) standard
  - An encoding specification: CIM-XML Encoding Specification
  - A transport mechanism: CIM operations over HTTP
- The Common Information Model (CIM) is a conceptual information model that describes management and is not bound to a particular implementation. This allows for the interchange of management information between management systems and applications. This can be either agent-to-manager or manager-to-manager commu-

nications that provide for distributed system management. There are two parts to CIM: the CIM Specification and the CIM Schema.

The CIM Specification describes the language, naming, and meta schema. The meta schema is a formal definition of the model. It defines the terms used to express the model and their usage and semantics. The elements of the meta schema are Classes, Properties, and Methods. The meta schema also supports Indications and Associations as types of Classes, and References as types of Properties.

The CIM Schema provides the actual model descriptions. The CIM Schema supplies a set of classes with properties and associations that provide a well understood conceptual framework within which it is possible to organize the available information about the managed environment.

- The Common Information Model Object Manager (CIMOM) is a CIM object manager or, more specifically, an application that manages objects according to the CIM standard.
- CIMOM providers are software that performs specific tasks within the CIMOM that are requested by client applications. Each provider instruments one or more aspects of the CIMOM's schema.

SUSE® Linux Enterprise Server contains the open source CIMOM from the OpenWBEM project [http://openwbem.org].

The Web-Based Enterprise Management software selection includes a set of packages that contain basic Novell providers, including some sample providers, and a base set of accompanying Novell schemas.

As Novell moves forward with OpenWBEM and development of specific providers, it will provide tools that offer the following important features:

- Efficient monitoring of network systems
- Recording of alterations within existing management configurations
- Hardware inventory and asset management

Understanding how the OpenWBEM CIMOM is set up and how to configure it can help you monitor and manage disparate systems in your network with more confidence and ease.

11.1 Setting Up OpenWBEM

To set up OpenWBEM, select the Web-Based Enterprise Management software selection or pattern in YaST when you install SUSE Linux Enterprise Server or select it as a component to install on a server that is already running SUSE Linux Enterprise Server. This software selection includes the following packages:

cim-schema, Common Information Model (CIM) Schema:
: This package contains the Common Information Model (CIM). CIM is a model for describing overall management information in a network or enterprise environment. CIM consists of a specification and a schema. The specification defines the details for integration with other management models. The schema provides the actual model descriptions.

openwbem, Web Based Enterprise Management (WBEM) Implementation:
: This package contains an implementation of OpenWBEM. OpenWBEM is a set of software components that help facilitate the deployment of the Distributed Management Task Force (DMTF) CIM and WBEM technologies. If you are not familiar with the DMTF and its technologies, you can visit the DMTF Web site [http://www.dmtf.org].

openwbem-base-providers:
: This package contains a Novell Linux instrumentation of base operating system components such as computer, system, operating system, and processes for the OpenWBEM CIMOM.

openwbem-smash-providers:
: This package contains a Novell Linux instrumentation of the Systems Management Architecture for Server Hardware (SMASH) providers for the OpenWBEM CIMOM.

yast2-cim, YaST2 - CIM Bindings:
: This package adds CIM bindings to YaST2 (YaST2 is the Graphical User Interface of the SUSE System Tools Manager). These bindings provide a client interface to the Common Information Model Object Manager (CIMOM).

This section includes the following information:

- Section 11.1.1, “Starting, Stopping, or Checking Status for owcimomd” (page 244)

- Section 11.1.2, “Ensuring Secure Access” (page 244)
- Section 11.1.3, “Setting Up Logging” (page 247)

11.1.1 Starting, Stopping, or Checking Status for owcimomd

When Web-Based Enterprise Management software is installed, the daemon, owcimomd, is started by default. The following table explains how to start, stop, and check status for owcimomd.

Table 11.1 *Commands for Managing owcimomd*

Task	Linux Command
Start owcimomd	As root in a console shell, enter `rcowcimomd start`.
Stop owcimomd	As root in a console shell, enter `rcowcimomd stop`.
Check owcimomd status	As root in a console shell, enter `rcowcimomd status`.

11.1.2 Ensuring Secure Access

The default setup of OpenWBEM is relatively secure. However, you might want to review the following to ensure access to OpenWBEM components is as secure as desired for your organization.

- Section “Certificates” (page 245)
- Section “Ports” (page 245)
- Section “Authentication” (page 247)

Certificates

Secure Socket Layers (SSL) transports require a certificate for secure communications to occur. When OES is installed, OpenWBEM has a self-signed certificate generated for it.

If desired, you can replace the path for the default certificate with a path to a commercial certificate that you have purchased or with a different certificate that you have generated in the `http_server.SSL_cert = path_filename` setting in the `/etc/openwbem/openwbem.conf` file.

The default generated certificate is in the following location:

```
/etc/openwbem/servercert.pem
```

If you want to generate a new certificate, use the following command. Running this command replaces the current certificate, so Novell recommends making a copy of the old certificate before generating a new one.

As root in a console shell, enter

```
sh/etc/openwbem/owgencert
```

If you want to change the certificate that OpenWBEM uses, see Section 11.2.2, “Changing the Certificate Configuration” (page 256).

Ports

OpenWBEM is configured by default to accept all communications through a secure port, 5989. The following table explains the port communication setup and recommended configuration.

Table 11.2 *Port Communication Setup and Recommended Configurations*

Port	Type	Notes and Recommendations
5989	Secure	The secure port that OpenWBEM communications use via HTTPS services. This is the default configuration.

Port	Type	Notes and Recommendations
		With this setting, all communications between the CIMOM and client applications are encrypted when sent over the Internet between servers and workstations. Users must authenticate through the client application to view this information. Novell recommends that you maintain this setting in the configuration file. In order for the OpenWBEM CIMOM to communicate with the necessary applications, this port must be open in routers and firewalls if they are present between the client application and the nodes being monitored.
5988	Unsecure	The unsecure port that OpenWBEM communications use via HTTP services. This setting is disabled by default. With this setting, all communications between the CIMOM and client applications are open for review when sent over the Internet between servers and workstations by anyone without any authentication. Novell recommends that you use this setting only when attempting to debug a problem with the CIMOM. As soon as the problem is resolved, set the non-secure port option back to Disabled. In order for the OpenWBEM CIMOM to communicate with the necessary applications that require non-secure access, this port must be open in routers and firewalls if they are present between the client application and the nodes being monitored.

If you want to change the default port assignments, see Section 11.2.3, “Changing the Port Configuration” (page 257).

Authentication

The following authentication settings are set and enabled as the default for OpenWBEM in SUSE Linux Enterprise Server.

You can change any of the default settings. See Section 11.2.1, “Changing the Authentication Configuration” (page 249).

- `http_server.allow_local_authentication = true`
- `http_server.ssl_client_verification = disabled`
- `http_server.use_digest = false`
- `owcimomd.allow_anonymous = false`
- `owcimomd.allowed_users = root`
- `owcimomd.authentication_module = /usr/lib/openwbem/authentication/libpamauthentication.so`

The OpenWBEM CIMOM is PAM enabled by default; therefore the local root user can authenticate to the OpenWBEM CIMOM with local root user credentials.

11.1.3 Setting Up Logging

You can change any of the default settings. For more information, see Section 11.2.4, “Changing the Default Logging Configuration” (page 258).

By default, logging for OpenWBEM is set up as follows.

- `log.main.components = *`
- `log.main.level = ERROR`
- `log.main.type = syslog`

This means that owcimomd logging is set up to go to the `/var/log/messages` file or to other files depending on the configuration of `syslogd`. It logs all errors for all components (owcimomd).

11.2 Changing the OpenWBEM CIMOM Configuration

When OpenWBEM CIMOM (owcimomd) starts, it reads it run-time configuration from the `openwbem.conf` file. The `openwbem.conf`file is located in the `/etc/openwbem` directory.

Any setting that has the options commented out with a semicolon (;) or pound sign (#) uses the default setting.

When making changes to this file, you can use any text editor that saves the file in a format that is native to the platform you are using.

You can change any of the settings in the `openwbem.conf` file. This section discusses the following configuration settings:

- Section 11.2.1, “Changing the Authentication Configuration” (page 249)
- Section 11.2.2, “Changing the Certificate Configuration” (page 256)
- Section 11.2.3, “Changing the Port Configuration” (page 257)
- Section 11.2.4, “Changing the Default Logging Configuration” (page 258)
- Section 11.2.5, “Configuring Debug Logging” (page 266)
- Section 11.2.6, “Configuring Additional Logs” (page 268)

11.2.1 Changing the Authentication Configuration

When changing the Authentication configuration, there are several things that you can control:

- Who can access the CIMOM
- What authentication module is used

See the following settings:

- Section “http_server.allow_local_authentication ” (page 249)
- Section “http_server.digest_password_file ” (page 250)
- Section “http_server.ssl_client_verification ” (page 251)
- Section “http_server.ssl_trust_store ” (page 252)
- Section “http_server.use_digest” (page 252)
- Section “owcimomd.ACL_superuser” (page 253)
- Section “owcimomd.allow_anonymous” (page 253)
- Section “owcimomd.allowed_users” (page 254)
- Section “owcimomd.authentication_module” (page 255)
- Section “simple_auth.password_file” (page 255)

http_server.allow_local_authentication

Purpose

Directs the http_server to allow local authentication without supplying a password, relying on local system file permissions.

You can use this setting with the Basic or Digest settings.

Syntax

```
http_server.allow_local_authentication = option
```

Option	Description
true	Enables local authentication. This is the default setting.
false	Disables local authentication.

Example

```
http_server.allow_local_authentication = true
```

http_server.digest_password_file

Purpose

Specifies a location for the password file. This is required if the http_server.use_digest setting is enabled.

Syntax

```
http_server.digest_password_file = path_filename
```

The following is the default path and filename for the digest password file:

```
/etc/openwbem/digest_auth.passwd
```

Example

```
http_server.digest_password_file =
/etc/openwbem/digest_auth.passwd
```

http_server.ssl_client_verification

Purpose

Determines whether the server should attempt to authenticate clients with SSL Client Certificate verification.

This setting is disabled by default.

Syntax:

```
http_server.ssl_client_verification = option
```

Option	Description
autoupdate	Specifies the same functionality as the *Optional* option; however, previously unknown client certificates that pass HTTP authentication are added to a trust store so that subsequent client connections with the same certificate do not require HTTP authentication.
disabled	Disables client certificate checking. This is the default setting.
optional	Allows a trusted certificate to be authenticated (no HTTP authentication is necessary). Also allows an untrusted certificate to pass the SSL handshake if the client passes the HTTP authentication.
required	Requires a trusted certificate for the SSL handshake to succeed.

Example

```
http_server.ssl_client_verification = disabled
```

http_server.ssl_trust_store

Purpose

Specifies a directory containing the OpenSSL trust store.

Syntax

`http_server.ssl_trust_store = path`

The following is the default path for the trust store file.

`/etc/openwbem/truststore`

Example

`http_server.ssl_trust_store = /etc/openwbem/truststore`

http_server.use_digest

Purpose

Directs the HTTP server to use Digest authentication, which bypasses the Basic authentication mechanism. To use digest, you must set up the digest password file using `owdigestgenpass`.

Digest doesn't use the authentication module specified by the owcimomd.authentication_module configuration setting.

Syntax

`http_server.use_digest = option`

Option	Description
false	Enables the Basic authentication mechanism. This is the default setting.

Option	Description
true	Disables the Basic authentication mechanism.

Example

```
http_server.use_digest = false
```

owcimomd.ACL_superuser

Purpose

Specifies the username of the user that has access to all Common Information Model (CIM) data in all namespaces maintained by the owcimomd. This user can be used to administer the `/root/security` name space, which is where all ACL user rights are stored.

ACL processing is not enabled until the `OpenWBEM_Acl1.0.mof` file has been imported.

Syntax

```
owcimomd.ACL_superuser = username
```

Example

```
owcimomd.ACL_superuser = root
```

owcimomd.allow_anonymous

Purpose

Enables or disables anonymous logins to owcimomd.

Syntax

```
owcimomd.allow_anonymous = option
```

Option	Description
false	Requires login with a username and password to access owcimomd data. This is the default and recommended setting.
true	Allows anonymous logins to owcimomd. This disables authentication. No username or password is required to access owcimomd data.

Example

```
owcimomd.allowed_anonymous = false
```

owcimomd.allowed_users

Purpose

Specifies a list of users who are allowed to access owcimomd data.

Syntax

```
owcimomd.allowed_users = option
```

Option	Description
username	Specifies one or more users who are allowed to access the owcimomd data. Separate each username with a space. The root user is the default setting.
*	Allows all users to authenticate (for example, if you choose to control access with ACLs instead).

Option	Description
	This option is enforced for all authentication methods unless owcimomd.allow_anonymous is set to `true`.

Example

```
owcimomd.allowed_users = bcwhitely jkcarey jlanderson
```

owcimomd.authentication_module

Purpose

Specifies the authentication module that is used by owcimomd. This setting should be an absolute path to the shared library containing the authentication module.

Syntax

```
owcimomd.authentication_module = path_filename
```

The following is the default path and filename for the authentication modules:

```
/usr/lib/openwbem/authentication/libpamauthentication.so
```

Example

```
owcimomd.authentication_module =
/usr/lib/openwbem/authentication/libpamauthentication.so
```

simple_auth.password_file

Purpose

Specifies the path to the password file when the simple authentication module is used.

This setting is disabled by default.

Syntax

```
simple_auth.password_file = path_filename
```

Example

```
simple_auth.password_file =
/etc/openwbem/simple_auth.passwd
```

11.2.2 Changing the Certificate Configuration

The http_server.SSL_cert and the http_server.SSL_key settings specify the location of the file or files that contains the host's private key and the certificate that is used by OpenSSL for HTTPS communications.

The `.pem` files are located in the following default location:

```
/etc/openwbem/servercert.pem
```

```
/etc/openwbem/serverkey.pem
```

Syntax

```
http_server.SSL_cert = path_filename
```

or

```
http_server.SSL_key = path_filename
```

NOTE

Both the key and certificate can be in the same file. In this case, the values of http_server.SSL_cert and http_server.SSL_key would be the same.

Examples

```
http_server.SSL_cert = /etc/openwbem/servercert.pem

http_server.SSL_key = /etc/openwbem/servercert.pem

http_server.SSL_key = /etc/openwbem/serverkey.pem
```

11.2.3 Changing the Port Configuration

The http_server.http_port and server.https_port settings specify the port number that owcimomd listens on for all HTTP and HTTPS communications.

Syntax

```
http_server.http_port = option
```

or

```
http_server.https_port = option
```

Option	Description
Specific_port_number	Specify the specific port for HTTP or HTTPS communications. For HTTP, the default port is 5988. For HTTPS, the default port is 5989.
-1	Disables HTTP or HTTPS connections (for example, if you only want to support HTTPS connections).
0	Dynamically assigns a port number at runtime.

Example

These settings disable the HTTP port and enable port 5989 for HTTPS communications:

```
http_server.http_port = -1

http_server.https_port = 5989
```

11.2.4 Changing the Default Logging Configuration

The following log settings in the `owcimomd.conf` file let you specify where and how much logging occurs, the type of errors logged, and the log size, filename, and format:

- Section “log.main.categories” (page 259)
- Section “log.main.components” (page 260)
- Section “log.main.format” (page 260)
- Section “log.main.level” (page 263)
- Section “log.main.location” (page 264)
- Section “log.main.max_backup_index” (page 264)
- Section “log.main.max_file_size” (page 265)
- Section “log.main.type” (page 266)

If you want to set up debug logging, see Section 11.2.5, “Configuring Debug Logging” (page 266).

If you want to set up additional logs, see Section 11.2.6, “Configuring Additional Logs” (page 268).

log.main.categories

Purpose

Specifies the categories the log outputs.

Syntax

```
log.main.categories = option
```

Option	Description
category_name	Specifies the categories to be logged using a space delimited list. The categories used in owcimomd are: • DEBUG • ERROR • FATAL • INFO For more information about these options, see Section "log.main.level" (page 263). If specified in this option, the predefined categories are not treated as levels, but as independent categories. No default is available; and if a category is not set, no categories are logged and the log.main.level setting is used.
*	All categories are logged. This is the default setting.

Example

```
log.main.categories = FATAL ERROR INFO
```

log.main.components

Purpose

Specifies the components that the log outputs.

Syntax

```
log.main.components = option
```

Option	Description
component_name	Specifies the components to be logged (such as owcimomd) using a space-delimited list. Providers can use their own components.
*	Specifies that all components are logged. This is the default setting.

Example

```
log.main.components = owcimomd nssd
```

log.main.format

Purpose

Specifies the format (text mixed with printf() style conversion specifiers) of the log messages.

Syntax

```
log.main.format = conversion_specifier
```

Option	Specifies
%%	%
%c	Component (such as owcimomd)
%d	Date Can be followed by a date format specifier enclosed between braces. For example, %d{%H:%M:%S} or %d{%d %b %Y %H:%M:%S}. If no date format specifier is given, then ISO 8601 format is assumed. The only addition is %Q, which is the number of milliseconds. For more information about the date format specifiers, see the documentation for the strftime() function found in the <ctime> header.
%e	Message as XML CDATA. This includes the "<![CDATA[" and ending "]]>"
%F	Filename
%l	Filename and line number. For example, file.cpp(100)
%L	Line number
%M	Method name where the logging request was issued (only works on C++ compilers which support __PRETTY_FUNCTION__ or C99's __func__).
%m	Message

Option	Specifies
%n	Platform-dependent line separator character (\n) or characters (\r\n).
%p	Category, also known as level or priority.
%r	Number of milliseconds elapsed between the start of the application and the creation of the logging event.
%t	Thread ID
\n	New line
\t	Tab
\r	Line feed
\\	\
\x<hexDigits>	Character represented in hexadecimal

It is possible to change the minimum field width, the maximum field width, and justification. The optional format modifier is placed between the percent sign (%) and the conversion character. The first optional format modifier is the left justification flag, which is the minus (-) character. The optional minimum field width modifier follows, which is an integer that represents the minimum number of characters to output. If the data item requires fewer characters, it is padded with spaces on either the left or the right, according to the justification flag. If the data item is larger than the minimum field width, the field is expanded to accommodate the data.

The maximum field width modifier is designated by a period (.) followed by a decimal constant. If the data item is longer than the maximum field, then the extra characters are removed from the beginning of the data item (by default) or from the end (if the left justification flag was specified).

Examples

Log4j TTCC layout:

```
"%r [%t] %-5p %c - %m"
```

Similar to TTCC but with some fixed-size fields:

```
"%-6r [%15.15t] %-5p %30.30c - %m"
```

XML output conforming to log4j.dtd 1.2, which can be processed by Chainsaw (if used, this must be on one line; it is split up here for readability):

```
"<log4j:event logger="%c" timestamp="%d{%s%Q}" level="%p"
thread="%t"> <log4j:message>%e</log4j:message>
<log4j:locationInfo class="" method="" file="%F"
line="%L"/></log4j:event>"
```

The following is the default:

```
log.main.format = [%t]%m
```

log.main.level

Purpose

Specifies the level the log outputs. If set, the log outputs all predefined categories at and above the specified level.

Syntax

```
log.main.level = option
```

Option	Description
DEBUG	Logs all Debug, Info, Error, and Fatal error messages.
ERROR	Logs all Error and Fatal error messages. This is the default setting.
FATAL	Logs only Fatal error messages.

Option	Description
INFO	Logs all Info, Error, and Fatal error messages.

Example

```
log.main. level = ERROR
```

log.main.location

Purpose

Specifies the location of the log file owcimomd uses when the log.main.type setting option specifies that logging is sent to a file.

Syntax

```
log.main.location = path_filename
```

Example

```
log.main.location = /system/cimom/var/owcimomd.log
```

log.main.max_backup_index

Purpose

Specifies the amount of backup logs that are kept before the oldest is erased.

Syntax

```
log.main.backup_index = option
```

Option	Description
unsigned_integer_above_0	Specifies the number of backup logs kept.

Option	Description
	The default setting is 1 log file.
0	No backup logs are made and the log is truncated when it reaches the maximum file size.

Example

```
log.main.max_backup_index = 1
```

log.main.max_file_size

Purpose

Specifies the maximum size (in KB) that the owcimomd log can grow to.

Syntax

```
log.main.max_file_size = option
```

Option	Description
unsigned_integer_in_KB	Limits the log to a certain size in KB.
0	Lets the log grow to an unlimited size. This is the default setting.

Example

```
log.main.max_file_size = 0
```

log.main.type

Purpose

Specifies the type of main log owcimomd uses.

Syntax

`log.main.type = option`

Option	Description
file	Sends all messages to a file that is identified in the log.main.location configuration setting.
null	Disables logging.
syslog	Sends all messages to the syslog interface. This is the default setting.

Example

```
log.main.type = syslog
```

11.2.5 Configuring Debug Logging

If owcimomd is run in debug mode, then the debug log is active with the following settings:

- `log.debug.categories = *`
- `log.debug.components = *`
- `log.debug.format = [%t] %m`
- `log.debug.level = *`

- `log.debug.type = stderr`

Debug Log with Color

If you want a color version of the debug log, use the following ASCII escape codes:

```
log.debug.format =
\x1b[1;37;40m[\x1b[1;31;40m%-.6t\x1b[1;37;40m]\x1b[1;32;40m
%m\x1b[0;37;40m
```

If you want to use additional colors, use the following codes with the `log.debug.format` command:

Table 11.3 *Additional Color Codes for the log.debug.format Command*

Color	Codes
red	\x1b[1;31;40m
dark red	\x1b[0;31;40m
green	\x1b[1;32;40m
dark green	\x1b[0;32;40m
yellow	\x1b[1;33;40m
dark yellow	\x1b[0;33;40m
blue	\x1b[1;34;40m
dark blue	\x1b[0;34;40m
purple	\x1b[1;35;40m
dark purple	\x1b[0;35;40m
cyan	\x1b[1;36;40m

Color	Codes
dark cyan	\x1b[0;36;40m
white	\x1b[1;37;40m
dark white	\x1b[0;37;40m
gray	\x1b[0;37;40m
reset color	\x1b[0;37;40m

11.2.6 Configuring Additional Logs

If you want to create additional logs, list the log names under this setting:

`owcimomd.additional_logs = logname`

Separate multiple lognames spaces.

Syntax

`owcimomd.additional_logs = logname`

For each log, the following settings apply:

- `log.log_name.categories`
- `log.log_name.components`
- `log.log_name.format`
- `log.log_name.level`
- `log.log_name.location`
- `log.log_name.max_backup_index`

- `log.log_name.max_file_size`

Example

```
owcimomd.additional_logs = errorlog1 errorlog2 errorlog3
```

11.3 For More Information

For more information about OpenWBEM, see the following information:

- Documents in `usr/share/doc/packages/openwbem` on the local server filesystem:
  - `readme`
  - `openwbem-faq.html`
- A Novell Cool Solutions Article: An Introduction to WBEM and OpenWBEM in SUSE Linux [http://www.novell.com/coolsolutions/feature/14625.html]
- OpenWBEM Web site [http://www.openwbem.org]
- DMTF Web site [http://www.dmtf.org]

Mass Storage over IP Networks—iSCSI

12

One of the central tasks in computer centers and when operating servers is providing hard disk capacity for server systems. Fiber channel is often used for this purpose in the mainframe sector. So far, UNIX computers and the majority of servers are not connected to central storage solutions.

linux-iSCSI provides an easy and reasonably inexpensive solution for connecting Linux computers to central storage systems. In principle, iSCSI represents a transfer of SCSI commands on the IP level. If a program starts an inquiry for such a device, the operating system produces the necessary SCSI commands. These are then embedded in IP packages and encrypted as necessary by software that is commonly known as an *iSCSI initiator*. The packages are then transferred to the corresponding iSCSI remote station, also called *iSCSI target*.

Many storage solutions provide access over iSCSI, but it is also possible to run a Linux server that provides an iSCSI target. In this case, it is important to set up the Linux server optimized for file system services. The iSCSI target just accesses block devices in Linux. Therefore it is possible to use RAID solutions to increase disk space as well as a lot of memory to improve data caching. For more information about RAID, also see Section 7.2, “Soft RAID Configuration” (page 123).

12.1 Setting Up an iSCSI Target

SUSE® Linux Enterprise Server comes with an open source iSCSI target solution that evolved from the Ardis iSCSI target. A basic setup can be done with YaST, but to take full advantage of iSCSI, a manual setup is required.

12.1.1 Creating iSCSI Targets with YaST

The iSCSI target configuration exports existing block devices or file system images to iSCSI initiators. First create the needed block devices with YaST or create file system images. For an overview of partitioning, see Section 8.5.7, “Using the YaST Partitioner” (page 155). File system images must be created manually. For example, if you want to create the image `/var/lib/xen/images/xen-0` with the size 4GB, first make sure that the directory is there then create the image itself:

```
mkdir -p /var/lib/xen/images
dd if=/dev/zero of=/var/lib/xen/images/xen-0 seek=1M bs=4096 count=1
```

To configure the iSCSI target, run the *iSCSI Target* module in YaST. The configuration is split into three tabs. In the *Service* tab, select the start mode and the firewall settings. If you want to access the iSCSI target from a remote machine, select *Open Port in Firewall*. If a `iSNS` server should manage the discovery and access control, activate *iSNS Access Control* and enter the IP address of your iSNS Server. Note, that you cannot use even valid hostnames, but must use the IP address. For more about `iSNS`, read Chapter 13, *iSNS for Linux Overview* (page 283).

The *Global* tab provides settings for the iSCSI server. The authentication set here is used for the discovery of services, not for accessing the targets. If you do not want to restrict the access to the discovery, use *No Authentication*.

If authentication is needed, there are two possibilities to consider. One is that an initiator must prove that it has the permissions to run a discovery on the iSCSI target. This is done with *Incoming Authentication*. The other is that the iSCSI target must prove to the initiator that it is the expected target. Therefore, the iSCSI target can also provide a username and password. This is done with *Outgoing Authentication*. Find more information about authentication in RFC 3720 (see `http://www.ietf.org/rfc/rfc3720.txt`).

The targets are defined in the *Targets* tab. Use *Add* to create a new iSCSI target. The first dialog asks for information about the device to export.

Target
: The *Target* line has a fixed syntax that looks like the following:

  ```
  iqn.yyyy-mm.<reversed domain name>
  ```

It always starts with iqn. yyyy-mm is the format of the date when this target is activated. Find more about naming conventions in RFC 3722 (see http://www.ietf.org/rfc/rfc3722.txt).

Identifier
: The *Identifier* is freely selectable. It should follow some scheme to make the whole system more structured.

LUN
: It is possible to assign several LUNs to a target. To do this, select a target in the *Targets* tab then click *Edit*. There, add new LUNs to an existing target.

Path
: Add the path to the block device or file system image to export.

The next menu configures the access restrictions of the target. The configuration is very similar to the configuration of the discovery authentication. In this case, at least an incoming authentication should be setup.

Next finishes the configuration of the new target, and brings you back to the overview page of the *Target* tab. Activate your changes by clicking on *Finish*.

12.1.2 Configuring an iSCSI Target Manually

Configure an iSCSI target in `/etc/ietd.conf`. All parameters in this file before the first *Target* declaration are global for the file. Authentication information in this portion has a special meaning—it is not global, but is used for the discovery of the iSCSI target.

If you have access to an `iSNS` server, the first thing to configure is telling the target about this server. Note, that the address of the `iSNS` server must always be given as IP address. A normal domain name is not sufficient. The configuration for this functionality looks like the following:

```
iSNSServer 192.168.1.111
iSNSAccessControl no
```

This configuration makes the `iSCSI` target register itself with the `iSNS` server, which in turn provides the discovery for initiators. For more about `iSNS` read Chapter 13,

iSNS for Linux Overview (page 283). Note that the access control for the `iSNS` discovery is not supported. Just keep `iSNSAccessControl no`.

All direct iSCSI authentication may be done in two directions. The iSCSI target can require the iSCSI initiator to authenticate with the `IncomingUser`, which can be added multiple times. The iSCSI initiator may also require the iSCSI target to authenticate. Use `OutgoingUser` for this. Both have the same syntax:

```
IncomingUser <username> <password>
OutgoingUser <username> <password>
```

The authentication is followed by one or several target definitions. For each target, add a `Target` section. This section always starts with a `Target` identifier followed by definitions of logical unit numbers:

```
Target iqn.yyyy-mm.<reversed domain name>[:identifier]
          Lun 0 Path=/dev/mapper/system-v3
          Lun 1 Path=/dev/hda4
          Lun 2 Path=/var/lib/xen/images/xen-1,Type=fileio
```

In the `Target` line, yyyy-mm is the date when this target is activated, and identifier is freely selectable. Find more about naming conventions in RFC 3722 (see `http://www.ietf.org/rfc/rfc3722.txt`). Three different block devices are exported in this example. The first one is a logical volume (see also Section 7.1, “LVM Configuration” (page 115)), the second is an IDE partition, and the third is an image available in the local file system. All these look like block devices to an iSCSI initiator.

Before activating the iSCSI target, add at least one `IncomingUser` after the `Lun` definitions. It does the authentication for the use of this target.

To activate all your changes, restart the iscsitarget daemon with `rcopen-iscsi restart`. Check your configuration in the `/proc` file system:

```
cat /proc/net/iet/volume
tid:1 name:iqn.2006-02.com.example.iserv:systems
        lun:0 state:0 iotype:fileio path:/dev/mapper/system-v3
        lun:1 state:0 iotype:fileio path:/dev/hda4
        lun:2 state:0 iotype:fileio path:/var/lib/xen/images/xen-1
```

There are many more options that control the behavior of the iSCSI target. Find them in the manual page of `ietd.conf`.

Active sessions are also displayed in the `/proc` file system. For each connected initiator, an extra entry is added to `/proc/net/iet/session`:

```
cat /proc/net/iet/session
tid:1 name:iqn.2006-02.com.example.iserv:system-v3
   sid:562949957419520
initiator:iqn.2005-11.de.suse:cn=rome.example.com,01.9ff842f5645
      cid:0 ip:192.168.178.42 state:active hd:none dd:none
   sid:281474980708864 initiator:iqn.2006-02.de.suse:01.6f7259c88b70
      cid:0 ip:192.168.178.72 state:active hd:none dd:none
```

12.1.3 Configuring Online Targets with ietadm

When changes to the iSCSI target configuration are necessary, you always must restart the target to activate changes that are done in the configuration file. Unfortunately, all active sessions are interrupted in this process. To maintain an undisturbed operation, the changes should be done in the main configuration file `/etc/ietd.conf`, but also made manually to the current configuration with the administration utility `ietadm`.

To create a new iSCSI target with a LUN, first update your configuration file. The additional entry could be:

```
Target iqn.2006-02.com.example.iserv:system2
          Lun 0 Path=/dev/mapper/system-swap2
          IncomingUser joe secret
```

To set up this configuration manually, proceed as follows:

1. Create a new target with the command `ietadm --op new --tid=2 --params Name=iqn.2006-02.com.example.iserv:system2`.
2. Add a logical unit with `ietadm --op new --tid=2 --lun=0 --params Path=/dev/mapper/system-swap2`.
3. Set the username and password combination on this target with `ietadm --op new --tid=2 --user --params=IncomingUser=joe,Password=secret`.
4. Check the configuration with `cat /proc/net/iet/volume`.

It is also possible to delete active connections. First, check all active connections with the command `cat /proc/net/iet/session`. This may look like:

```
cat /proc/net/iet/session
tid:1 name:iqn.2006-03.com.example.iserv:system
        sid:281474980708864 initiator:iqn.1996-04.com.example:01.82725735af5
                cid:0 ip:192.168.178.72 state:active hd:none dd:none
```

To delete the session with the session ID 281474980708864, use the command `ietadm --op delete --tid=1 --sid=281474980708864 --cid=0`. Be aware that this makes the device unaccessible on the client system and processes accessing this device are likely to hang.

ietadm can also be used to change various configuration parameters. Obtain a list of the global variables with `ietadm --op show --tid=1 --sid=0`. The output looks like:

```
InitialR2T=Yes
ImmediateData=Yes
MaxConnections=1
MaxRecvDataSegmentLength=8192
MaxXmitDataSegmentLength=8192
MaxBurstLength=262144
FirstBurstLength=65536
DefaultTime2Wait=2
DefaultTime2Retain=20
MaxOutstandingR2T=1
DataPDUInOrder=Yes
DataSequenceInOrder=Yes
ErrorRecoveryLevel=0
HeaderDigest=None
DataDigest=None
OFMarker=No
IFMarker=No
OFMarkInt=Reject
IFMarkInt=Reject
```

All of these parameters may be changed easily. For example, if you want to change the maximum number of connections to two, use `ietadm --op update --tid=1 --params=MaxConnections=2`. In the file `/etc/ietd.conf`, the associated line should look like `MaxConnections 2`.

WARNING: Update ietd.conf According to Changes with ietadm

The changes that you make with the command `ietadm` are not permanent for the system. These changes are lost at the next reboot if they are not added

to the configuration file `/etc/ietd.conf`. Depending on the usage of iSCSI in your network, this may lead to severe problems.

There are several more options available for the command `ietadm`. Find an overview with `ietadm -h`. The abbreviations there are target ID (tid), session ID (sid), and connection ID (cid). They can also be found in `/proc/net/iet/session`.

12.2 Configuring iSCSI Initiator

iSCSI initiator, also called client, can be used to connect to any iSCSI target. This is not restricted to the iSCSI target solution explained above. The configuration of iSCSI initiator involves two major steps—the discovery of available iSCSI targets and the setup of an iSCSI session. Both can be done with YaST.

12.2.1 Using YaST for the iSCSI Initiator Configuration

The configuration is divided into three tabs. The *Service* tab may be used to enable the iSCSI initiator at boot time. It also offers to set a unique *Initiator Name* and an `iSNS` server to use for the discovery. The default port for `iSNS` is `3205`. The *Connected Targets* tab gives an overview of the currently connected iSCSI targets. Like the *Discovered Targets* tab, it gives the option to add new targets to the system. *Discovered Targets* is the tab to start with. It provides the possibility of discovering iSCSI targets in the network.

1. Use *Discovery* to open the discovery dialog.
2. Enter the IP address and change the port if necessary.
3. If necessary, add the *Incoming* or *Outgoing* authentication.
4. Use *Next* to start the discovery.

After a successful discovery, use *Login* to activate the target. You will be asked for authentication information to use the selected iSCSI target. *Next* finishes the configuration. If everything went well, the target now appears in *Connected Targets*.

The virtual iSCSI device is now available. Find the actual device with `lsscsi`:

```
lsscsi
[1:0:0:0]   disk    IET      VIRTUAL-DISK     0     /dev/sda
```

12.2.2 Setting Up the iSCSI Initiator Manually

Both the discovery and the configuration of iSCSI connections require a running iscsid. When running the discovery the first time, the internal database of the iSCSI initiator is created in the directory `/var/lib/open-iscsi`.

If your discovery is password protected, provide the authentication information to iscsid. Because the internal database does not exist when doing the first discovery, it cannot be used at this time. Instead, the configuration file `/etc/iscsid.conf` must be edited to provide the information. To add your password information for the discovery, add the following lines to the end of `/etc/iscsid.conf`:

```
discovery.sendtargets.auth.authmethod = CHAP
discovery.sendtargets.auth.username = <username>
discovery.sendtargets.auth.password = <password>
```

The discovery stores all received values in an internal persistent database. In addition, it displays all detected targets. Run this discovery with the command `iscsiadm -m discovery --type=st --portal=<targetip>`. The output should look like:

```
149.44.171.99:3260,1 iqn.2006-02.com.example.iserv:systems
```

To discover the available targets on a `iSNS` server, use the command `iscsiadm --mode discovery --type isns --portal <targetip>`

For each target defined on the iSCSI target, one line appears. Learn how to obtain more information about the stored data in Section 12.2.4, “The iSCSI Client Databases” (page 279).

The special `--login` option of `iscsiadm` creates all needed devices:

```
iscsiadm -m node -n iqn.2006-02.com.example.iserv:systems --login
```

The newly generated devices show up in the output of `lsscsi` and can now be accessed by mount.

12.2.3 Configuring LVM Autoassembly on iSCSI Devices

LVM startup is supported by udev, so that any LVM volume groups will be activated automatically via udev once all required physical volumes have been detected.

LVM autoassembly in udev makes use of the udev helper program `collect`. This program takes as the first argument an abstract `ID` to be checked, followed by a list of component IDs. Once this program has been called with each of the component IDs as the first argument it'll return 0.

So for autoassembly the physical volume UUIDs for the given volume group is registered as argument list for `collect`. udev (or rather vol_id) is capable of detecting the physical volume UUID on a device and hence it can be passed as the first argument to `collect`.

Once `collect` has been called with all physical volume UUIDs (i.e. udev has received events for all component devices) the next rule triggers which just calls `vgchange -a y <vgname>` and the volume group will be activated.

How to configure

Use the script `/usr/share/doc/packages/lvm2/lvm-vg-to-udev-rules.sh`. It takes as argument the volume group you want to start automatically. This script will generate the required udev rules. Now restart iSCSI to activate the volume groups. If you want to have the array started automatically on boot you have to switch the iSCSI component devices to `automatic`, so that the initiator will log into the target automatically on boot.

12.2.4 The iSCSI Client Databases

All information that was discovered by the iSCSI initiator is stored in two database files that reside in `/var/lib/open-iscsi`. There is one database for the discovery

of targets and one for the discovered nodes. When accessing a database, you first must select if you want to get your data from the discovery or from the node database. Do this with the `-m discovery` and `-m node` parameters of `iscsiadm`. Using `iscsiadm` just with one of these parameters gives an overview of the stored records:

```
iscsiadm -m discovery
149.44.171.99:3260,1 iqn.2006-02.com.example.iserv:systems
```

The target name in this example is `iqn.2006-02.com.example.iserv:systems`. This name is needed for all actions that relate to this special data set. To examine the content of the data record with the ID `iqn.2006-02.com.example.iserv:systems`, use the following command:

```
iscsiadm -m node --targetname iqn.2006-02.com.example.iserv:systems
node.name = iqn.2006-02.com.example.iserv:systems
node.transport_name = tcp
node.tpgt = 1
node.active_conn = 1
node.startup = manual
node.session.initial_cmdsn = 0
node.session.reopen_max = 32
node.session.auth.authmethod = CHAP
node.session.auth.username = joe
node.session.auth.password = ********
node.session.auth.username_in = <empty>
node.session.auth.password_in = <empty>
node.session.timeo.replacement_timeout = 0
node.session.err_timeo.abort_timeout = 10
node.session.err_timeo.reset_timeout = 30
node.session.iscsi.InitialR2T = No
node.session.iscsi.ImmediateData = Yes
....
```

To edit the value of one of these variables, use the command `iscsiadm` with the `update` operation. For example, if you want iscsid to log in to the iSCSI target when it initializes, set the variable `node.startup` to the value `automatic`:

```
iscsiadm -m node -n iqn.2006-02.com.example.iserv:systems --op=update
--name=node.startup --value=automatic
```

Remove obsolete data sets with the operation `delete`. If the target `iqn.2006-02.com.example.iserv:systems` is no longer a valid record, delete this record with the command `iscsiadm -m node -n iqn.2006-02.com.example.iserv:systems --op=delete`. Use this

option with caution because it deletes the record without any additional confirmation prompt.

To get a list of all discovered targets, run the command `iscsiadm -m node`.

12.2.5 For More Information

The iSCSI protocol has been available for several years. There are many reviews and additional documentation comparing iSCSI with SAN solutions, doing performance benchmarks, or just describing hardware solutions. Important pages for more information about open-iscsi are:

- http://www.open-iscsi.org/
- http://www.open-iscsi.org/cgi-bin/wiki.pl
- http://www.novell.com/coolsolutions/appnote/15394.html

There is also some online documentation available. See the manual pages of `iscsiadm`, `iscsid`, `ietd.conf`, and `ietd` and the example configuration file `/etc/iscsid.conf`.

13 iSNS for Linux Overview

Storage area networks (SANs) can contain many disk drives that are dispersed across complex networks. This can make device discovery and device ownership difficult. iSCSI initiators must be able to identify storage resources in the SAN and determine whether they have access to them.

Internet Storage Name Service (iSNS) is a standards-based service that is available with SUSE Linux Enterprise Server (SLES) 10 Support Pack 2. iSNS facilitates the automated discovery, management, and configuration of iSCSI devices on a TCP/IP network. iSNS provides intelligent storage discovery and management services comparable to those found in Fibre Channel networks.

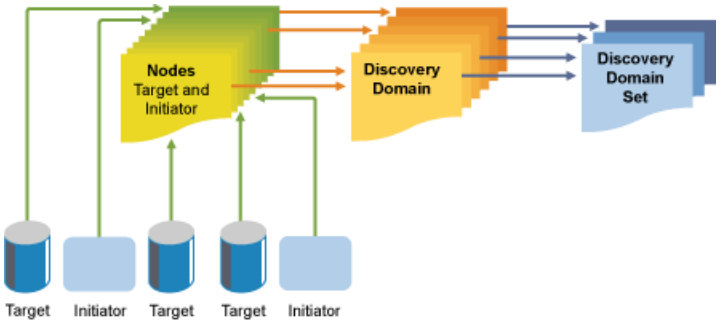
13.1 How iSNS Works

For an iSCSI initiator to discover iSCSI targets, it needs to identify which devices in the network are storage resources and what IP addresses it needs to access them. A query to an iSNS server returns a list of iSCSI targets and the IP addresses that the initiator has permission to access.

Using iSNS, you create iSNS discovery domains and discovery domain sets. You then group or organize iSCSI targets and initiators into discovery domains and group the discovery domains into discovery domain sets. By dividing storage nodes into domains, you can limit the discovery process of each host to the most appropriate subset of targets registered with iSNS, which allows the storage network to scale by reducing the number of unnecessary discoveries and by limiting the amount of time each host spends estab-

lishing discovery relationships. This lets you control and simplify the number of targets and initiators that must be discovered.

Figure 13.1 *iSNS Discovery Domains and Discovery Domain Sets*



Both, iSCSI targets and iSCSI initiators use iSNS clients to initiate transactions with iSNS servers using the iSNS protocol. They then register device attribute information in a common discovery domain, download information about other registered clients, and receive asynchronous notification of events that occur in their discovery domain.

iSNS servers respond to iSNS protocol queries and requests made by iSNS clients using the iSNS protocol. iSNS servers initiate iSNS protocol state change notifications and store properly authenticated information submitted by a registration request in an iSNS database.

Some of the benefits provided by iSNS for Linux include:

- Provides an information facility for registration, discovery, and management of networked storage assets.
- Integrates with the DNS infrastructure.
- Consolidates registration, discovery, and management of iSCSI storage.
- Simplifies storage management implementations.
- Improves scalability compared to other discovery methods.

An example of the benefits iSNS provides can be better understood through the following scenario:

Suppose you have a company that has 100 iSCSI initiators and 100 iSCSI targets. Depending on your configuration, all iSCSI initiators could potentially try to discover and connect to any of the 100 iSCSI targets. This could create a discovery and connection nightmare. By grouping initiators and targets into discovery domains, you can prevent iSCSI initiators in one department from discovering the iSCSI targets in another department. The result is that the iSCSI initiators in a specific department only discover those iSCSI targets that are part of the department's Discovery Domain.

13.2 iSNS for Linux Installation and Setup

iSNS for Linux is included with SLES 10 SP2, but is not installed or configured by default. You must install the iSNS package modules (isns and yast2-isns modules) and configure or set up iSNS to use it.

NOTE

iSNS can be installed on the same server as an iSCSI target or initiator. Using an iSCSI target and initiator on the same machine is not supported.

To install iSNS for Linux:

1. Start YaST and select *Software Management*.
2. In the *search* field, enter `isns`.
3. Select both the isns and yast2-isns packages, then click *Accept*.

13.3 Setting Up iSNS

iSNS must be started at the server. You can do this by entering `rcisns start` or `/etc/init.d/isns start` at the server console of the server where you install it. You can also use the stop, status, and restart options with iSNS.

iSNS can also be configured to start automatically each time the server is rebooted. To do this

1. Start YaST and under *Network Services*, select *iSNS Server*.

2. With the *Service* tab selected, specify the IP address of your iSNS server, then click *Save Address*.

3. In the Service Start section of the screen, select *When Booting*.

   You can also choose to start the iSNS server manually. You must then use the `rcisns start` command to start the service each time the server is restarted.

13.3.1 Creating iSNS Discovery Domains

In order for iSCSI initiators and targets to use the iSNS service, they must belong to a discovery domain. A default discovery domain named *default DD* is automatically created when you install the iSNS service. The existing iSCSI targets and initiators that have been configured to use iSNS are automatically added to the default discovery domain.

To create a new discovery domain:

1. Start YaST and under *Network Services*, select *iSNS Server*.

2. Click the *Discovery Domains* tab, then click the *Create Discovery Domain* button.

   You can also select an existing discovery domain and click the *Delete* button to remove that discovery domain.

3. Specify the name of the discovery domain you are creating, then click *OK*.

13.3.2 Creating iSNS Discovery Domain Sets

Discovery domains must belong to a discovery domain set. You can create a discovery domain and add nodes to that discovery domain, but it is not active and the iSNS service does not function unless you add the discovery domain to a discovery domain set. A default discovery domain set named *default DDS* is automatically created when you

install iSNS and the default discovery domain is automatically added to that domain set.

To create a discovery domain set:

1. Start YaST and under *Network Services*, select *iSNS Server*.
2. Click the *Discovery Domains Sets* tab, then click the *Create Discovery Domain Set* button.

   You can also select an existing discovery domain set and click the *Delete* button to remove that discovery domain set.
3. Specify the name of the discovery domain set you are creating, then click *OK*.

13.3.3 Adding iSCSI Nodes to a Discovery Domain

1. Start YaST and under *Network Services*, select *iSNS Server*.
2. Click the *iSCSI Nodes* tab and ensure the iSCSI targets and initiators that you want to use the iSNS service are listed.

   If an iSCSI target or initiator is not listed, you might need to restart the iSCSI service on the node. You can do this by running the `rcopen-iscsi restart` command to restart an initiator or the `rciscsitarget restart` command to restart a target.

   You can select an iSCSI node and click the *Delete* button to remove that node from the iSNS database. This is useful if you are no longer using an iSCSI node or have renamed it.

   The iSCSI node will be automatically added to the list (iSNS database) again when you restart the iSCSI service or reboot the server unless you remove or comment out the iSNS portion of the iSCSI configuration file.
3. Click the *Discovery Domains* tab, select the desired discovery domain, then click the *Display Members* button.

4 Click *Add existing iSCSI Node*, select the node you want to add to the domain, then click *Add Node*.

5 Repeat the last step for as many nodes as you want to add to the discovery domain, then click *Done* when you are finished adding nodes.

An iSCSI node can belong to more than one discovery domain.

13.3.4 Adding Discovery Domains to a Discovery Domain Set

1 Start YaST and under *Network Services*, select *iSNS Server*.

2 Click the *Discovery Domains Set* tab.

3 Select *Create Discovery Domain Set* to add a new set to the list of discovery domain sets.

4 Choose a discovery domain set to modify.

5 Click *Add Discovery Domain*, select the discovery domain you want to add to the discovery domain set, then click *Add Discovery Domain*.

6 Repeat the last step for as many discovery domains as you want to add to the discovery domain set, then click *Done*.

A discovery domain can belong to more than one discovery domain set.

13.4 For More Information

The `linuxisns` project is hosted at `http://sourceforge.net/projects/linuxisns/`. The Mailinglist for this project is found at `http://sourceforge.net/mailarchive/forum.php?forum_name=linuxisns-discussion`. General information about `iSNS` was written down in `rfc4171`. See also `http://www.ietf.org/rfc/rfc4171`.

14 Oracle Cluster File System 2

Oracle Cluster File System 2 (OCFS2) is a general-purpose journaling file system that is fully integrated in the Linux 2.6 kernel and later. OCFS2 allows you to store application binary files, data files, and databases on devices in a SAN. All nodes in a cluster have concurrent read and write access to the file system. A distributed lock manager helps prevent file access conflicts. OCFS2 supports up to 32,000 subdirectories and millions of files in each directory. The O2CB cluster service (a driver) runs on each node to manage the cluster.

OCFS2 was added to SUSE Linux Enterprise Server 9 to support Oracle Real Application Cluster (RAC) databases and its application files, Oracle Home. In SUSE Linux Enterprise Server 10 and later, OCFS2 can be used for any of the following storage solutions:

- Oracle RAC and other databases
- General applications and workloads
- XEN image store in a cluster

  XEN virtual machines and virtual servers can be stored on OCFS2 volumes that are mounted by cluster servers to provide quick and easy portability of XEN virtual machines between servers.
- LAMP (Linux, Apache, MySQL, and PHP | PERL | Python) stacks

In addition, it is fully integrated with Heartbeat 2.

As a high-performance, symmetric, parallel cluster file system, OCFS2 supports the following functions:

- An application's files are available to all nodes in the cluster. Users simply install it once on an OCFS2 volume in the cluster.
- All nodes can concurrently read and write directly to storage via the standard file system interface, enabling easy management of applications that run across a cluster.
- File access is coordinated through the Distributed Lock Manager (DLM).

  DLM control is good for most cases, but an application's design might limit scalability if it contends with the DLM to coordinate file access.
- Storage backup functionality is available on all back-end storage. An image of the shared application files can be easily created, which can help provide effective disaster recovery.

OCFS2 also provides the following capabilities:

- Metadata caching
- Metadata journaling
- Cross-node file data consistency
- A GTK GUI-based administration via the `ocfs2console` utility
- Operation as a shared-root file system
- Support for multiple-block sizes (each volume can have a different block size) up to 4 KB, for a maximum volume size of 16 TB
- Support for up to 255 cluster nodes
- Context-dependent symbolic link (CDSL) support for node-specific local files
- Asynchronous and direct I/O support for database files for improved database performance

14.1 O2CB Cluster Service

The O2CB cluster service is a set of modules and in-memory file systems that are required to manage OCFS2 services and volumes. You can enable these modules to be loaded and mounted during system boot. For instructions, see Section 14.6.2, "Configuring OCFS2 Services" (page 296).

Table 14.1 *O2CB Cluster Service Stack*

Service	Description
Node Manager (NM)	Keeps track of all the nodes in the `/etc/ocfs2/cluster.conf` file
Heartbeat (HB)	Issues up/down notifications when nodes join or leave the cluster
TCP	Handles communications between the nodes with the TCP protocol
Distributed Lock Manager (DLM)	Keeps track of all locks and their owners and status
CONFIGFS	User space configuration file system. For details, see Section 14.3, "In-Memory File Systems" (page 292)
DLMFS	User space interface to the kernel space DLM. For details, see Section 14.3, "In-Memory File Systems" (page 292)

14.2 Disk Heartbeat

OCFS2 requires the nodes to be alive on the network. The O2CB cluster service sends regular keepalive packages to ensure that they are alive. It uses a private connection between nodes instead of the LAN to avoid network delays that might be interpreted as a node disappearing and thus, lead to a node's self-fencing.

The OC2B cluster service communicates the node status via a disk heartbeat. The heartbeat system file resides on the Storage Area Network (SAN), where it is available to all nodes in the cluster. The block assignments in the file correspond sequentially to each node's slot assignment.

Each node reads the file and writes to its assigned block in the file at two-second intervals. Changes to a node's time stamp indicates the node is alive. A node is dead if it does not write to the heartbeat file for a specified number of sequential intervals, called the heartbeat threshold. Even if only a single node is alive, the O2CB cluster service must perform this check, because another node could be added dynamically at any time.

You can modify the disk heartbeat threshold in the `/etc/sysconfig/o2cb` file, using the `O2CB_HEARTBEAT_THRESHOLD` parameter. The wait time is calculated as follows:

```
(O2CB_HEARTBEAT_THRESHOLD value – 1) * 2 = threshold in seconds
```

For example, if the `O2CB_HEARTBEAT_THRESHOLD` value is set at the default value of 7, the wait time is 12 seconds ((7 - 1) * 2 = 12).

14.3 In-Memory File Systems

OCFS2 uses two in-memory file systems for communications:

Table 14.2 *In-Memory File Systems Used by OCFS2*

In-Memory File System	Description	Mount Point
configfs	Communicates the list of nodes in the cluster to the in-kernel node manager, and communicates the resource used for the heartbeat to the in-kernel heartbeat thread	`/config`
ocfs2_dlmfs	Communicates locking and unlocking for cluster-wide locks on resources to the in-kernel distributed lock manager that keeps track of all locks and their owners and status	`/dlm`

14.4 Management Utilities and Commands

OCFS2 stores node-specific parameter files on the node. The cluster configuration file (`/etc/ocfs2/cluster.conf`) resides on each node assigned to the cluster.

The `ocfs2console` utility is a GTK GUI-based interface for managing the configuration of the OCFS2 services in the cluster. Use this utility to set up and save the `/etc/ocfs2/cluster.conf` file to all member nodes of the cluster. In addition, you can use it to format, tune, mount, and umount OCFS2 volumes.

IMPORTANT

The file browser column in the ocfs2console utility is prohibitively slow and inconsistent across the cluster. We recommend that you use the `ls(1)` command to list files instead.

Additional OCFS2 utilities are described in the following table. For information about syntax for these commands, see their man pages.

Table 14.3 *OCFS2 Utilities*

OCFS2 Utility	Description
debugfs.ocfs2	Examines the state of the OCFS file system for the purpose of debugging.
fsck.ocfs2	Checks the file system for errors and optionally repairs errors.
mkfs.ocfs2	Creates an OCFS2 file system on a device, usually a partition on a shared physical or logical disk. This tool requires the O2CB cluster service to be up.
mounted.ocfs2	Detects and lists all OCFS2 volumes on a clustered system. Detects and lists all nodes on the system that have mounted an OCFS2 device or lists all OCFS2 devices.

OCFS2 Utility	Description
ocfs2cdsl	Creates a context-dependent symbolic link (CDSL) for a specified filename (file or directory) for a node. A CDSL filename has its own image for a specific node, but has a common name in the OCFS2.
tune.ocfs2	Changes OCFS2 file system parameters, including the volume label, number of node slots, journal size for all node slots, and volume size.

Use the following commands to manage O2CB services. For more information about the `o2cb` command syntax, see its man page.

Table 14.4 *O2CB Commands*

Command	Description
/etc/init.d/o2cb status	Reports whether the o2cb services are loaded and mounted
/etc/init.d/o2cb load	Loads the O2CB modules and in-memory file systems
/etc/init.d/o2cb online *ocfs2*	The cluster named ocfs2 gets online At least one node in the cluster must be active for the cluster to be online.
/etc/init.d/o2cb offline *ocfs2*	The cluster named ocfs2 gets offline
/etc/init.d/o2cb unload	Unloads the O2CB modules and in-memory file systems
/etc/init.d/o2cb start *ocfs2*	If the cluster is set up to load on boot, starts the cluster named ocfs2 by loading `o2cb` and onlining the cluster At least one node in the cluster must be active for the cluster to be online.

Command	Description
/etc/init.d/o2cb stop *ocfs2*	If the cluster is set up to load on boot, stops the cluster named ocfs2 by offlining the cluster and unloading the O2CB modules and in-memory file systems

14.5 OCFS2 Packages

The OCFS2 kernel module (`ocfs2`) is installed automatically in SUSE Linux Enterprise Server 10 and later. To use OCFS2, use YaST (or the command line if you prefer) to install the `ocfs2-tools` and `ocfs2console` packages on each node in the cluster.

1. Log in as the `root` user, then open the YaST Control Center.
2. Select *Software* > *Software Management*.
3. In the *Search* field, enter `ocfs2`.

   The software packages `ocfs2-tools` and `ocfs2console` should be listed in the right panel. If they are selected, the packages are already installed.
4. If you need to install the packages, select them, then click *Install* and follow the on-screen instructions.

14.6 Creating an OCFS2 Volume

Follow the procedures in this section to configure your system to use OCFS2 and to create OCFS2 volumes.

14.6.1 Prerequisites

Before you begin, do the following:

- Initialize, carve, or configure RAIDs (Redundant Array of Independent Disks) on the SAN disks, as needed, to prepare the devices you plan to use for your OCFS2 volumes. Leave the devices as free space.

  We recommend that you store application files and data files on different OCFS2 volumes, but it is only mandatory to do so if your application volumes and data volumes have different requirements for mounting. For example, the Oracle RAC database volume requires the `datavolume` and `nointr` mounting options, but the Oracle Home volume should never use these options.

- Make sure that the `ocfs2console`, and `ocfs2-tools` packages are installed. Use YaST or command line methods to install them if they are not. For YaST instructions, see Section 14.5, “OCFS2 Packages” (page 295).

14.6.2 Configuring OCFS2 Services

Before you can create OCFS2 volumes, you must configure OCFS2 services. In the following procedure, you generate the `/etc/ocfs2/cluster.conf` file, save the `cluster.conf` file on all nodes, and create and start the O2CB cluster service (`o2cb`).

Follow the procedure in this section for one node in the cluster.

1. Open a terminal window and log in as the `root` user.
2. If the `o2cb` cluster service is not already enabled, enter `chkconfig --add o2cb`.

   When you add a new service, `chkconfig` ensures that the service has either a start or a kill entry in every run level.
3. If the `ocfs2` service is not already enabled, enter `chkconfig --add ocfs2`.
4. Configure the `o2cb` cluster service driver to load on boot.

   4a. Enter `/etc/init.d/o2cb configure`

   4b. At the `Load O2CB driver on boot (y/n) [n]` prompt, enter `y` (yes) to enable load on boot.

4c At the `Cluster to start on boot (Enter "none" to clear) [ocfs2]` prompt, enter `none`. This choice presumes that you are setting up OCFS2 for the first time or resetting the service. You specify a cluster name in the next step when you set up the `/etc/ocfs2/cluster.conf` file.

5 Use the `ocfs2console` utility to set up and save the `/etc/ocfs2/cluster.conf` file to all member nodes of the cluster.

This file should be the same on all the nodes in the cluster. Use the following steps to set up the first node. Later, you can use the `ocfs2console` to add new nodes to the cluster dynamically and to propagate the modified `cluster.conf` file to all nodes.

However, if you change other settings, such as the cluster name and IP address, you must restart the cluster for the changes to take effect, as described in Step 6 (page 298).

5a Open the `ocfs2console` GUI by entering `ocfs2console`.

5b In the `ocfs2console`, select *Cluster* > *Cluster Nodes*.

If `cluster.conf` is not present, the console will create one with a default cluster name of `ocfs2`. Modify the cluster name as desired.

5c In the Node Configuration dialog box, click *Add* to open the Add Node dialog box.

5d In the Add Node dialog box, specify the unique name of your primary node, a unique IP address (such as *`192.168.1.1`*), and the port number (optional, default is 7777), then click *OK*.

The `ocfs2console` console assigns node slot numbers sequentially from 0 to 254.

5e In the Node Configuration dialog box, click *Apply*, then click *Close* to dismiss the Add Node dialog box.

5f Click *Cluster* > *Propagate Configuration* to save the `cluster.conf` file to all nodes.

6 If you need to restart the OCFS2 cluster for the changes to take effect, enter the following lines, waiting in between for the process to return a status of *OK*.

```
/etc/init.d/o2cb stop
/etc/init.d/o2cb start
```

14.6.3 Creating an OCFS2 Volume

Creating an OCFS2 file system and adding new nodes to the cluster should be performed on only one of the nodes in the cluster.

1 Open a terminal window and log in as the `root` user.

2 If the O2CB cluster service is offline, start it by entering the following command then wait for the process to return a status of *OK*.

```
/etc/init.d/o2cb online ocfs2
```

Replace *`ocfs2`* with the actual cluster name of your OCFS2 cluster.

The OCFS2 cluster must be online, because the format operation must first ensure that the volume is not mounted on any node in the cluster.

3 Create and format the volume using one of the following methods:

- In EVMSGUI, go to the Volumes page, select *Make a file system* > *OCFS2*, then specify the configuration settings.
- Use the `mkfs.ocfs2` utility. For information about the syntax for this command, refer to the `mkfs.ocfs2` man page.
- In the `ocfs2console`, click *Tasks* > *Format*, select a device in the Available Devices list that you want to use for your OCFS2 volume, specify the configuration settings for the volume, then click *OK* to format the volume.

See the following table for recommended settings.

OCFS2 Parameter	Description and Recommendation
Volume label	A descriptive name for the volume to make it uniquely identifiable when it is mounted on different nodes. Use the `tunefs.ocfs2` utility to modify the label as needed.
Cluster size	Cluster size is the smallest unit of space allocated to a file to hold the data. Options are 4, 8, 16, 32, 64, 128, 256, 512, and 1024 KB. Cluster size cannot be modified after the volume is formatted. Oracle recommends a cluster size of 128 KB or larger for database volumes. Oracle also recommends a cluster size of 32 or 64 KB for Oracle Home.
Number of node slots	The maximum number of nodes that can concurrently mount a volume. On mounting, OCFS2 creates separate system files, such as the journals, for each of the nodes. Nodes that access the volume can be a combination of little-endian architectures (such as x86, x86-64, and ia64) and big-endian architectures (such as ppc64 and s390x). Node-specific files are referred to as local files. A node slot number is appended to the local file. For example: journal:0000 belongs to whatever node is assigned to slot number 0. Set each volume's maximum number of node slots when you create it, according to how many nodes that you expect to concurrently mount the volume. Use the `tunefs.ocfs2` utility to increase the number of node slots as needed; the value cannot be decreased.
Block size	The smallest unit of space addressable by the file system. Specify the block size when you create the volume.

OCFS2 Parameter	Description and Recommendation
	Options are 512 bytes (not recommended), 1 KB, 2 KB, or 4 KB (recommended for most volumes). Block size cannot be modified after the volume is formatted.

14.7 Mounting an OCFS2 Volume

1. Open a terminal window and log in as the `root` user.
2. If the O2CB cluster service is offline, start it by entering the following command, then wait for the process to return a status of *OK*.

   ```
   /etc/init.d/o2cb online ocfs2
   ```

   Replace *ocfs2* with the actual cluster name of your OCFS2 cluster.

   The OCFS2 cluster must be online, because the format operation must ensure that the volume is not mounted on any node in the cluster.
3. Use one of the following methods to mount the volume.

   - In the `ocfs2console`, select a device in the Available Devices list and click *Mount*. Optionally, specify the directory mount point and mount options and click *OK*.
   - Mount the volume from the command line, using the `mount` command.
   - Mount the volume from the `/etc/fstab` file on system boot.

   Mounting an OCFS2 volume takes about 5 seconds, depending on how long it takes for the heartbeat thread to stabilize. On a successful mount, the device list in the `ocfs2console` shows the mount point along with the device.

TIP: Adding New Nodes

When new nodes try to connect to the cluster, they are not allowed to join because the nodes have not added them to their connection list. To solve this issue, manually go to each node and issue the following command to update the respective connection list:

```
o2cb_ctl -H -n ocfs2 -t cluster -a online=yes
```

For information about mounting an OCFS2 volume using any of these methods, see the *OCFS2 User Guide [http://oss.oracle.com/projects/ocfs2/documentation/]* on the OCFS2 project at Oracle [http://oss.oracle.com/projects/ocfs2/].

When running Oracle RAC, make sure to use the `datavolume` and `nointr` mounting options for OCFS2 volumes that contain the Voting diskfile (CRS), Cluster registry (OCR), Data files, Redo logs, Archive logs, and Control files. Do not use these options when mounting the Oracle Home volume.

Option	Description
`datavolume`	Ensures that the Oracle processes open the files with the o_direct flag.
`nointr`	No interruptions. Ensures the IO is not interrupted by signals.

14.8 Additional Information

For information about using OCFS2, see the *OCFS2 User Guide [http://oss.oracle.com/projects/ocfs2/documentation/]* on the OCFS2 project at Oracle [http://oss.oracle.com/projects/ocfs2/].

15 Access Control Lists in Linux

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/`.

15.1 Traditional File Permissions

The basics of traditional Linux file permissions are explained in Section 18.2, “Users and Access Permissions” (page 361). More advanced features are the setuid, setgid, and sticky bit.

15.1.1 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`.

15.1.2 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.

15.1.3 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
```

15.2 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.

15.3 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.

15.4 Handling ACLs

Table 15.1, “ACL Entry Types” (page 307) 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 15.2, “Masking Access Permissions” (page 307) 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 15.1 *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 15.2 *Masking Access Permissions*

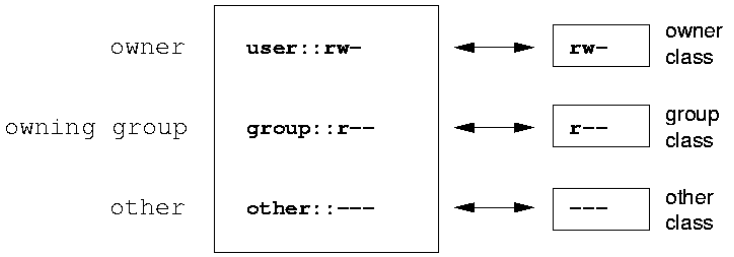
Entry Type	Text Form	Permissions
named user	`user:geeko:r-x`	`r-x`
mask	`mask::rw-`	`rw-`
	effective permissions:	`r--`

15.4.1 ACL Entries and File Mode Permission Bits

Figure 15.1, “Minimum ACL: ACL Entries Compared to Permission Bits” (page 308) and Figure 15.2, “Extended ACL: ACL Entries Compared to Permission Bits” (page 308) 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

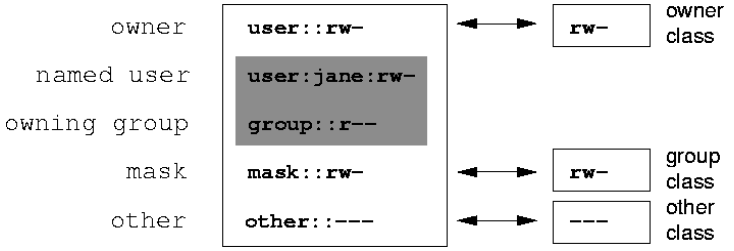
ACL entry owner. *Other class* permissions are mapped to the respective ACL entry. However, the mapping of the *group class* permissions is different in the two cases.

Figure 15.1 *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 15.1, “Minimum ACL: ACL Entries Compared to Permission Bits” (page 308). In the case of an extended ACL—with mask—the group class permissions are mapped to the mask entry. This is shown in Figure 15.2, “Extended ACL: ACL Entries Compared to Permission Bits” (page 308).

Figure 15.2 *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.

15.4.2 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 15.2, “Masking Access Permissions” (page 307)). 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`.

15.4.3 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 311)). 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.

15.4.4 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".

15.5 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.

15.6 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)`.

16 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`.

16.1 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.

16.2 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`.

16.3 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`.

16.4 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.

16.5 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 16.1, "The Most Important RPM Query Options" (page 320).

Table 16.1 *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 16.1, “rpm -q -i wget” (page 321).

Example 16.1 *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 16.2, “Script to Search for Packages” (page 321). Pass the partial filename to the script shown as a parameter when running it.

Example 16.2 *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 16.2 *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.

16.6 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.

16.7 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.

16.8 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 Section 8.3.1, “Installing and Removing Software” (page 132)).

17 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.

17.1 Debugging

17.1.1 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
```

17.1.2 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
```

17.1.3 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`.

17.2 Files and File Systems

17.2.1 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)
```

17.2.2 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
```

17.2.3 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
```

17.2.4 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
```

17.3 Hardware Information

17.3.1 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`.

17.3.2 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
```

17.3.3 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.

17.4 Networking

17.4.1 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
```

17.5 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]
```

17.5.1 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:

17.6 Processes

17.6.1 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
```

17.6.2 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
```

17.6.3 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:

17.6.4 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.

17.7 System Information

17.7.1 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.

17.7.2 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.

17.7.3 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.

17.7.4 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`.

17.7.5 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
```

17.7.6 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
```

17.7.7 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 17.6.4, “Processes: `top`” (page 341).

```
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
```

17.8 User Information

17.8.1 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.

17.9 Time and Date

17.9.1 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
```

18 Working with the Shell

When booting your Linux system, you are usually directed to a graphical user interface that guides you through the login process and the following interactions with the system. Although graphical user interfaces have become very important and user-friendly, using them is not the only way to communicate with your system. You can also use a text-oriented communication like a command line interpreter, usually called the shell, where you can enter commands. Because Linux provides options to start shell windows from the graphical user interface, you can easily use both methods.

In administration, shell-based applications are especially important for controlling computers over slow network links or if you want to perform tasks as `root` on the command line. For Linux "newbies" it might be rather unusual to enter commands in a shell, but you will soon realize that the shell is not only for administrators—in fact, using the shell is often the quickest and easiest way to perform some daily tasks.

There are several shells for UNIX or Linux. The default shell in SUSE® Linux Enterprise is Bash (GNU Bourne-Again Shell).

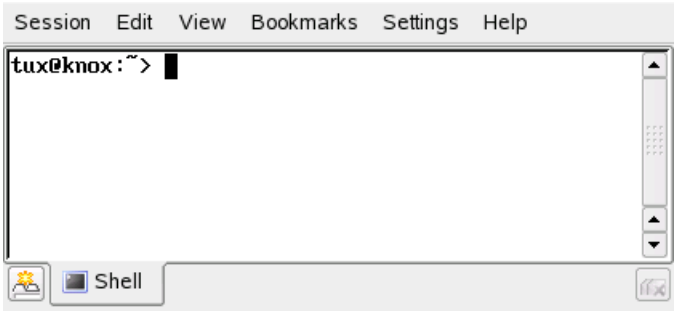
This chapter deals with a couple of basics you need to know for using the shell. This includes the following topics: how to enter commands, the directory structure of Linux, how to work with files and directories and how to use some basic functions, the user and permission concept of Linux, an overview of important shell commands, and a short introduction to the vi editor, which is a default editor always available in Unix and Linux systems.

18.1 Getting Started with the Bash Shell

In Linux, you can use the command line parallel to the graphical user interface and easily switch between them. To start a terminal window from the graphical user interface in KDE, click the Konsole icon in the panel. In GNOME, click the GNOME Terminal icon in the panel.

The Konsole or the GNOME Terminal window appears, showing the prompt on the first line like in Figure 18.1, “Example of a Bash Terminal Window” (page 350). 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 is after this prompt, you can send commands directly to your computer system.

Figure 18.1 *Example of a Bash Terminal Window*



18.1.1 Entering Commands

A command consists of several elements. The first element is always the actual command, followed by parameters or options. You can type a command and edit it by using ←, →, <—, Del, and Space. You can also add options or correct typing errors. The command is executed when you press Enter.

IMPORTANT: No News Is Good News

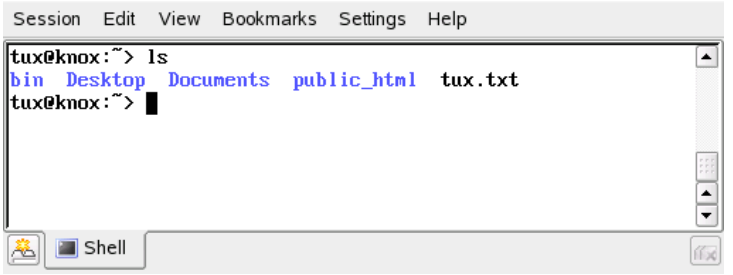
The shell is not verbose: in contrast to some graphical user interfaces, it usually does not provide confirmation messages when commands have been executed. Messages only appear in case of problems or errors.

Also keep this in mind for commands to delete objects. Before entering a command like `rm` for removing a file, you should know if you really want to get rid of the object: it will be deleted irretrievably, without enquiry.

Using Commands without Options

Look at the structure of commands using a simple example: the `ls` command, used to list the contents of a directory. The command can be used with or without options. Entering the plain `ls` command shows the contents of the current directory:

Figure 18.2 *The ls Command*



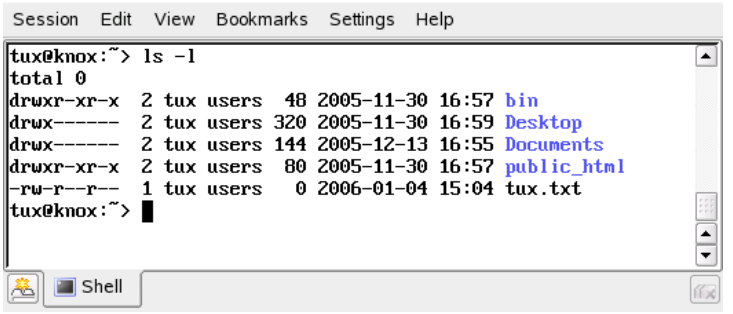
Unlike in other operating systems, files in Linux may have a file extension, such as `.txt`, but do not need to have one. This makes it difficult to differentiate between files and folders in this output of the `ls`. By default, the colors can give you a hint: directories are usually shown in blue, files in black.

Using Commands with Options

A better way to get more details about the contents of a directory is using the `ls` command with a string of options. Options modify the way a command works so that you can get it to do specific tasks. Options are separated from the command with a blank

and are prefixed with a hyphen. The `ls -l` command shows the contents of the same directory in full detail (long listing format):

Figure 18.3 *The ls -l Command*



On the left of each object name, information about the object is shown in several columns. The most important are the following: The first column shows the file type of the object (in this example, `d` for directory or – for normal files). The next nine columns show the user permissions for the object. Columns 11 and 12 show the name of the file owner and the group (in this case, `tux` and `users`). Find information about user permissions and the user concept of Linux in Section 18.2, “Users and Access Permissions” (page 361). The next column shows the file size in bytes. Then date and time of the last change are displayed. The last column shows the object name.

If you want to see even more, you can combine two options for the `ls` command and enter `ls -la`. The shell now also shows hidden files in the directory, indicated by a dot in front (for example, `.hiddenfile`).

Getting Help

Nobody is expected to know all options of all commands by heart. If you remember the command name but are not sure about the options, you can enter the command followed by a blank and `--help`. This `--help` option exists for many commands. Entering `ls --help` displays all the options for the `ls` command.

18.1.2 Linux Directory Structure

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 knowlegde of the default directory structure in a Linux system. You can think of directories as electronic folders in which files, programs, and subdirectories are stored. The top level directory in the hierarchy is the root directory, referred to as `/`. This is the place from which all other directories can be accessed.

Figure 18.4 shows the standard directory tree in Linux, with the home directories of the example users `yxz`, `linux`, and `tux`. The `/home` directory contains the directories in which the individual users can store their personal files.

NOTE: Home Directory in a Network Environment

If you are working in a network environment, your home directory may not be called `/home`. It can be mapped to any directory in the file system.

The following list provides a brief description of the standard directories in Linux.

Figure 18.4 *Excerpt from a Standard Directory Tree*

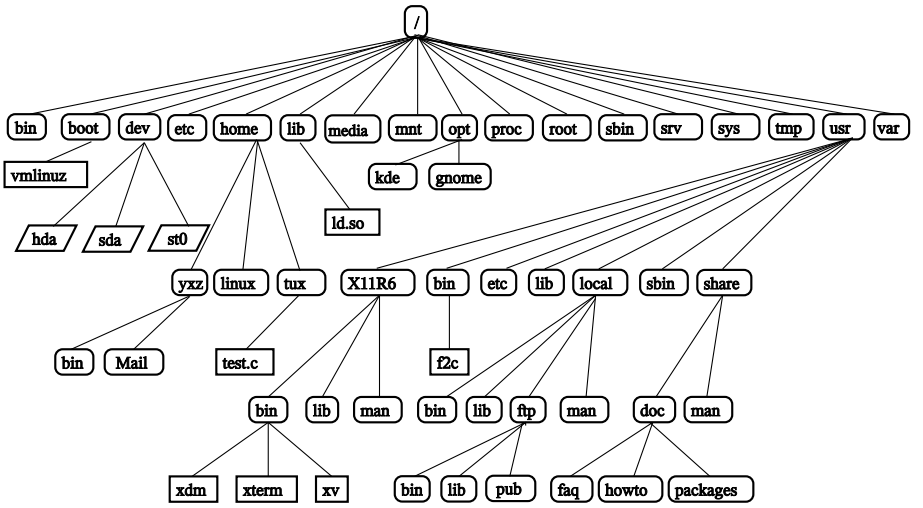


Table 18.1 *Overview of a Standard Directory Tree*

`/`	Root directory, starting point of the directory tree
`/home`	Personal directories of users
`/dev`	Device files that represent hardware components
`/etc`	Important files for system configuration
`/etc/init.d`	Boot scripts
`/bin`, `/sbin`	Programs needed early in the boot process (`/bin`) and for the administrator (`/sbin`)
`/usr`, `/usr/local`	All application programs and local, distribution-independent extensions (`/usr/local`)
`/usr/bin`, `/usr/sbin`	Generally accessible programs (`/usr/bin`) and reserved for the system administrator (`/usr/sbin`)
`/usr/share/doc`	Various documentation files
`/tmp`, `/var/tmp`	Temporary files (do not save files in this directory unless you do not need them)
`/opt`	Optional software, larger add-on program packages (such as KDE, GNOME, and Netscape)
`/proc`	Process file system
`/sys`	System file system where all device information for the kernel is gathered
`/var/log`	System log files

18.1.3 Working with Directories and Files

To address a certain file or directory, you must specify the path leading to that directory or file. There are two ways to specify a path:

- The entire (absolute) path from the root directory to the respective file
- A path starting from the current directory (relative path)

Absolute paths always start with a slash. Relative paths do not have a slash at the beginning.

NOTE: Linux Is Case-Sensitive

Linux distinguishes between uppercase and lowercase in the file system. For example, entering `test.txt` or `Test.txt` makes a difference in Linux. Keep this in mind when entering filenames or paths.

To change directories, use the `cd` command.

- To switch to your home directory, enter `cd`.
- Refer to the current directory with a dot (`.`). This is mainly useful for other commands (`cp`, `mv`, …).
- The next higher level in the tree is represented by two dots (`..`). For example, to switch to the parent directory of your current directory, enter `cd ..`.

Examples of Addressing a File

The `cd` commands in Section 18.1.3, "Working with Directories and Files" (page 355) used relative paths. You can use also absolute paths. For example, suppose you want to copy a file from your home directory to a subdirectory of `/tmp`:

1. First, from your home directory create a subdirectory in `/tmp`:

   1a. If your current directory is not your home directory, enter `cd ~` to switch to it. From anywhere in the file system, you can reach your home directory by entering `cd ~`.

1b In your home directory, enter `mkdir /tmp/test`. `mkdir` stands for "make directory". This command creates a new directory named `test` in the `/tmp` directory. In this case, use an absolute path to create the directory.

1c To check what happened, now enter `ls -l /tmp`. The new directory `test` should appear in the list of contents of the `/tmp` directory.

2 Now create a new file in your home directory and copy it to the `/tmp/test` directory by using a relative path.

2a Enter `touch myfile.txt`. The `touch` command with the `myfile.txt` option creates a new, empty file named `myfile.txt` in your current directory.

2b Check this by entering `ls -l`. The new file should appear in the list of contents.

2c Enter `cp myfile.txt ../tmp/test`. This copies `myfile.txt` to the directory `/tmp/test` without changing the name of the file.

2d Check this by entering `ls -l /tmp/test`. The file `myfile.txt` should appear in the list of contents for `/tmp/test`.

To list the contents of home directories of other users, enter `ls ~`*`username`* . In the example directory tree in Figure 18.4, "Excerpt from a Standard Directory Tree" (page 353), one of the sample users is `tux`. In this case, `ls ~tux` would list the contents of the home directory of `tux`.

NOTE: Handling Blanks in Filenames or Directory Names

If a filename contains a space, either escape the space using a back slash (`\`) in front of the blank or enclose the filename in single or double quotes. Otherwise Bash interprets a filename like `My Documents` as the names of two files or directories. The difference between single and double quotes is that variable expansion takes place within double quotes. Single quotes ensure that the shell sees the quoted string literally.

18.1.4 Useful Features of the Shell

Entering commands in Bash can include a lot of typing. In the following, get to know some features of the Bash that can make your work a lot easier and save a lot of typing.

History and Completion

By default, Bash "remembers" commands you have entered. This feature is called *history*. To repeat a command that has been entered before, press ↑ until the desired command appears at the prompt. Press ↓ to move forward through the list of previously entered commands. Use Ctrl + R to search in the history.

You can edit the selected command, for example, changing the name of a file, before you execute the command by pressing Enter. To edit the command line, just move the cursor to the desired position using the arrow keys and start typing.

Completing a filename or directory name to its full length after typing its first letters is another helpful feature of Bash. To do so, type the first letters then press →|. If the filename or path can be uniquely identified, it is completed at once and the cursor moves to the end of the filename. You can then enter the next option of the command, if necessary. If the filename or path cannot be uniquely identified (because there are several filenames starting with the same letters), the filename or path is only completed up to the point where again several options are possible. You can then obtain a list of them by pressing →| a second time. After this, you can enter the next letters of the file or path then try completion again by pressing →|. When completing filenames and paths with the help of →|, you can simultaneously check whether the file or path you want to enter really exists (and you can be sure of getting the spelling right).

Wild Cards

Another convenience offered by the shell is wild cards for pathname expansion. Wild cards are characters that can stand for other characters. There are three different types of these in Bash:

`?`
: Matches exactly one arbitrary character

`*`
: Matches any number of characters

`[set]`

Matches one of the characters from the group specified inside the square brackets, which is represented here by the string *set*. As part of *set* you can also specify character classes using the syntax *[:class:]*, where a class is one of `alnum`, `alpha`, `ascii`, etc.

Using `!` or `^` at the beginning of the group ([!*set*]) matches one character other than those identified by *set*.

Assuming that your `test` directory contains the files `Testfile`, `Testfile1`, `Testfile2`, and `datafile`.

- The command `ls Testfile?` lists the files `Testfile1` and `Testfile2`.
- The command `ls Testfile?` lists the files `Testfile1` and `Testfile2`.
- With `ls Test*`, the list also includes `Testfile`.
- The command `ls *fil*` shows all the sample files.
- Use the `set` wild card to address all sample files whose last character is a number: `ls Testfile[1-9]` or, using classes, `ls Testfile[[:digit:]]`.

Of the four types of wild cards, the most inclusive one is the asterisk. It could be used to copy all files contained in one directory to another one or to delete all files with one command. The command `rm *fil*`, for instance, would delete all files in the current directory whose name includes the string *fil*.

Viewing Files with Less and More

Linux includes two small programs for viewing text files directly in the shell: `less` and `more`. Rather than starting an editor to read a file like `Readme.txt`, simply enter `less Readme.txt` to display the text in the console window. Use Space to scroll down one page. Use Page Up and Page Down to move forward or backward in the text. To exit less, press Q.

Instead of less, you can also use the older program more. However, it is less convenient because it does not allow you to scroll backwards.

The program less got its name from the the precept that *less is more* and can also be used to view the output of commands in a convenient way. To see how this works, read Section "Redirection and Pipes" (page 359).

Redirection and Pipes

Normally, the standard output in the shell is your screen or the console window and the standard input is the keyboard. However, the shell provides functions by which you can redirect the input or the output to another object, such as a file or another command. With the help of the symbols > and <, for example, you can forward the output of a command to a file (output redirection) or use a file as input for a command (input redirection). For example, if you want to write the output of a command such as `ls` to a file, enter `ls -l > file.txt`. This creates a file named `file.txt` that contains the list of contents of your current directory as generated by the `ls` command. However, if a file named `file.txt` already exists, this command overwrites the existing file. To prevent this, use >>. Entering `ls -l >> file.txt` simply appends the output of the `ls` command to an already existing file named `file.txt`. If the file does not exist, it is created.

Sometimes it is also useful to use a file as the input for a command. For example, with the `tr` command, you can replace characters redirected from a file and write the result to the standard output, your screen. Suppose you want to replace all characters `t` of your `file.txt` from the example above with `x` and print this to your screen. Do so by entering `tr t x < file.txt`.

Just like the standard output, the standard error output is sent to the console. To redirect the standard error output to a file named `errors`, append `2> errors` to the corresponding command. Both standard output and standard error are saved to one file named `alloutput` if you append `>& alloutput`.

Using *pipelines* or *pipes* is also a sort redirection, although the use of the pipe is not constrained to files. With a pipe (`|`), you can combine several commands, using the output of one command as input for the next command. For example, to view the contents or your current directory in `less`, enter `ls | less`. This only makes sense if the normal output with `ls` would be too lengthy. For instance, if you view the contents of the `dev` directory with `ls /dev`, you only see a small portion in the window. View the entire list with `ls /dev | less`.

18.1.5 Archives and Data Compression

Now that you have already created a number of files and directories, consider the subject of archives and data compression. Suppose you want to have the entire `test` directory packed in one file that you can save on a USB stick as a backup copy or send by e-mail. To do so, use the command `tar` (for *tape archiver*). With `tar --help`, view all the options for the `tar` command. The most important of these options are explained here:

-c
: (for create) Create a new archive.

-t
: (for table) Display the contents of an archive.

-x
: (for extract) Unpack the archive.

-v
: (for verbose) Show all files on screen while creating the archive.

-f
: (for file) Choose a filename for the archive file. When creating an archive, this option must always be given as the last one.

To pack the `test` directory with all its files and subdirectories into an archive named `testarchive.tar`, do the following:

1. Open a shell.
2. Use `cd` to your home directory where the `test` directory is located.
3. Enter `tar -cvf testarchive.tar test`. The `-c` option creates the archive, making it a file as directed by `-f`. The `-v` option lists the files as they are processed.
4. View the contents of the archive file with `tar -tf testarchive.tar`.

The `test` directory with all its files and directories has remained unchanged on your hard disk. To unpack the archive, enter `tar -xvf testarchive.tar`, but do not try this yet.

For file compression, the obvious choice is `gzip` or, for a even better compression ratio, `bzip2`. Just enter `gzip testarchive.tar` (or `bzip2 testarchive.tar`, but `gzip` is used in this example). With `ls`, now see that the file `testarchive.tar` is no longer there and that the file `testarchive.tar.gz` has been created instead. This file is much smaller and therefore much better suited for transfer via e-mail or storage on a USB stick.

Now, unpack this file in the `test2` directory created earlier. To do so, enter `cp testarchive.tar.gz test2` to copy the file to that directory. Change to the directory with `cd test2`. A compressed archive with the `.tar.gz` extension can be unzipped with the `gunzip` command. Enter `gunzip testarchive.tar.gz`, which results in the file `testarchive.tar`, which then needs to be extracted or *untarred* with `tar -xvf testarchive.tar`. You can also unzip and extract a compressed archive in one step with `tar -xvf testarchive.tar.gz` (adding the `-z` option is no longer required). With `ls`, you can see that a new `test` directory has been created with the same contents as your `test` directory in your home directory.

18.1.6 Cleaning Up

After this crash course, you should be familiar with the basics of the Linux shell or command line. You may want to clean up your home directory by deleting the various test files and directories using the `rm` and `rmdir` commands. In Section 18.3, "Important Linux Commands" (page 365), find a list of the most important commands and a brief description of their functions.

18.2 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.

18.2.1 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 18.1, "Sample Output Showing File Permissions" (page 362).

Example 18.1 *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 18.2 *Sample Output Showing Directory Permissions*

```
drwxrwxr-x 1 tux project3 35 Jun 21 15:15  ProjectData
```

In Example 18.2, "Sample Output Showing Directory Permissions" (page 363), 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 362), 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`).

18.2.2 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 18.2, "Sample Output Showing Directory Permissions" (page 363) 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 18.2, “Sample Output Showing Directory Permissions” (page 363) 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 18.1, “Sample Output Showing File Permissions” (page 362) 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.

18.3 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`.

18.3.1 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 lo-

cated. 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.

18.3.2 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 22.1.6, “The free Command” (page 430) 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 the 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.

18.3.3 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.

18.4 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.

18.4.1 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.

18.4.2 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 18.2, "Simple Commands of the vi Editor" (page 378) This list is far from complete. More complete lists are available in the documentation found in Section 18.4.3, "For More Information" (page 379)

Table 18.2 *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

18.4.3 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. SUSE Linux Enterprise 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, http://www.linuxgazette.com/node/view/9039, 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 III. System

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

19

SUSE Linux Enterprise® is available for several 64-bit platforms. This does not necessarily mean that all the applications included have already been ported to 64-bit platforms. SUSE Linux Enterprise 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 SUSE Linux Enterprise 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.

NOTE: 31-Bit Applications on IBM System z:

s390 on IBM System z uses a 31-bit environment. References to 32-bit applications in the following also apply to 31-bit applications.

SUSE Linux Enterprise for the 64-bit platforms ia64, ppc64, s390x, and x86_64 is designed so that existing 32-bit applications run in the 64-bit environment "out-of-the-box." The corresponding 32-bit platforms are x86 for ia64, ppc for ppc64, s390 for s390x, and x86 for x86_64. 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. The current ppc64 system runs most applications in 32-bit mode, but you can run 64-bit applications.

19.1 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.

The same approach is used for the 64-bit platforms ppc64, s390x, and x86_64: 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`, `/usr/lib`, and `/usr/X11R6/lib` are now found under `/lib64`, `/usr/lib64`, and `/usr/X11R6/lib64`. This means that there is space for the 32-bit libraries under `/lib`, `/usr/lib` and `/usr/X11R6/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. For example, the X11 fonts are still found in the usual location under `/usr/X11R6/lib/X11/fonts`. This scheme conforms to LSB (Linux Standards Base) and FHS (File System Hierarchy Standard).

► **ipf:** The 64-bit libraries for ia64 are located in the standard `lib` directories. In such cases, there is neither a `lib64` directory or a `lib32` directory. ia64 executes the 32-bit x86 code under an emulation. A set of basic libraries is installed in `/emul/ia32-linux/lib` and `/emul/ia32-linux/usr/X11R6/lib`. ◄

19.2 Software Development

All 64-bit architectures support the development of 64-bit objects. The level of support for 32-bit compiling depends on the architecture. These are the various implementation options for the tool chain from GCC (GNU Compiler Collection) and binutils, which include the assembler `as` and the linker `ld`:

Biarch Compiler
: Both 32-bit and 64-bit objects can be generated with a biarch development tool chain. The compilation of 64-bit objects is the default on almost all platforms. 32-bit objects can be generated if special flags are used. This special flag is `-m32` for GCC (`-m31` for generating s390 binaries). The flags for the binutils are architecture-dependent, but GCC transfers the correct flags to linkers and assemblers. A biarch development tool chain currently exists for amd64 (supports development for x86 and amd64 instructions), for s390x, and for ppc64. 32-bit objects are normally created on the ppc64 platform. The `-m64` flag must be used to generate 64-bit objects.

No Support
: SUSE Linux Enterprise does not support the direct development of 32-bit software on all platforms. To develop applications for x86 under ia64, use the corresponding 32-bit version of SUSE Linux Enterprise.

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 SUSE Linux Enterprise environment is designed according to this principle. In the case of manually updated libraries, resolve these issues yourself.

19.3 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` or `rpmname-x86` (for ia64) if the second architecture is a 32-bit architecture or `rpmname-64bit` if the second architecture is a 64-bit architecture. 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` or `rpmname-devel-64bit`.

For example, to compile a program that uses `libaio` on a system whose second architecture is a 32-bit architecture (x86_64 or s390x), you need the following RPMs:

libaio-32bit
: 32-bit runtime package

libaio-devel-32bit
: Headers and libraries for 32-bit development

libaio
: 64-bit runtime package

libaio-devel
: 64-bit development headers and libraries

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. Examples for s390x with s390 as the second architecture or ppc64 with ppc as the second architecture would be similar. This example does not apply to ia64 where you do not build 32-bit packages.

TIP

When using s390 as second architecture, you have to use `-m31` instead of `-m32`, because this is a 31 bit system.

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/X11R6/lib/
   ```

Not all of these variables are needed for every program. Adapt them to the respective program.

An example `configure` call to compile a native 32-bit application on x86_64, ppc64, or s390x could appear as follows:

```
CC="gcc -m32"            \
LDFLAGS="-L/usr/lib;"  \
        .configure       \
          --prefix=/usr  \
          --libdir=/usr/lib
make
make install
```

19.4 Kernel Specifications

The 64-bit kernels for x86_64, ppc64, and s390x 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 on non-ppc64 platforms as 64-bit programs to function properly. On IBM System z, not all ioctls are available in the 32-bit kernel ABI.

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

20

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.

20.1 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. If the BIOS supports network booting, it is also possible to configure a boot server that provides the boot loader. On x86 systems, PXE boot is needed. Other architectures commonly use the BOOTP protocol to get 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 21, *The Boot Loader* (page 405). For a network boot, the BIOS acts as the boot loader. It gets the image to start from the boot server then starts the system. This is completely independent of local hard disks.

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. If special hardware drivers are needed before the mass storage can be accessed, they must be in initramfs. For more information about initramfs, refer to Section 20.1.1, "initramfs" (page 390). If the system does not have a local hard disk, initramfs must provide the root file system to the kernel. This can be done with the help of a network block device like iSCSI or SAN, but it is also possible to use NFS as the root device.

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 20.1.2, "init on initramfs" (page 391). Find more information about udev in Chapter 24, *Dynamic Kernel Device Management with udev* (page 463).

5. **init** init handles the actual booting of the system through several different levels providing different functionality. init is described in Section 20.2, "The init Process" (page 393).

20.1.1 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 the `initramfs` file. 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 SUSE Linux Enterprise®, 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.

20.1.2 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 (especially your hard drive). To access the root file system, the kernel needs to load the proper file system drivers.

Providing Block Special Files
: For each loaded module, the kernel generates device events. udev handles these events and generates the required block 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 7.2, "Soft RAID Configuration" (page 123). Find information about LVM in Section 7.1, "LVM Configuration" (page 115). Find information about EVMS and special storage settings in *Storage Administration Guide*.

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.

  If the file system resides on a networked block device like iSCSI or SAN, connection to the storage server is also set up by the initramfs.

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 20.1.1, "initramfs" (page 390), 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.

20.2 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 20.2.1, "Runlevels" (page 393)). The file also specifies which services and daemons are available in each of the levels. Depending on the entries in `/etc/inittab`, several scripts are run by init. For reasons of clarity, these scripts, called *init scripts*, all reside in the directory `/etc/init.d` (see Section 20.2.2, "Init Scripts" (page 396)).

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.

20.2.1 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 20.1, "Available Runlevels" (page 394). As an alternative, the runlevel can be specified at boot time (by adding the runlevel number 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 the single number 3 to the boot prompt.

Table 20.1 *Available Runlevels*

Runlevel	Description
0	System halt
S or 1	Single user mode
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 SUSE Linux Enterprise 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 26, *The X Window System* (page 483), 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 checks the current runlevel (`runlevel`) 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 run the scripts with the `stop` parameter, because there are some dependencies to consider.

4. The last things to start are the start scripts of the new runlevel. In this example, these are in `/etc/init.d/rc5.d` and begin with an `S`. Again, the number that follows the `S` determines the sequence in which the scripts are started.

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`.

20.2.2 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). For IBM System z systems, this is the case only during the boot process or if an immediate system shutdown is initiated (power failure or via “signal quiesce”). 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 exe-

cuted 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 20.2, “Possible init Script Options” (page 397). Scripts that are run directly by init do not have these links. They are run independently from the runlevel when needed.

Table 20.2 *Possible init Script Options*

Option	Description
`start`	Start service.
`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, making block special files available) 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 20.2.1, "Runlevels" (page 393).

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 20.1, "A Minimal INIT INFO Block" (page 399).

Example 20.1 *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 20.2.3, "Configuring System Services (Runlevel) with YaST" (page 400).

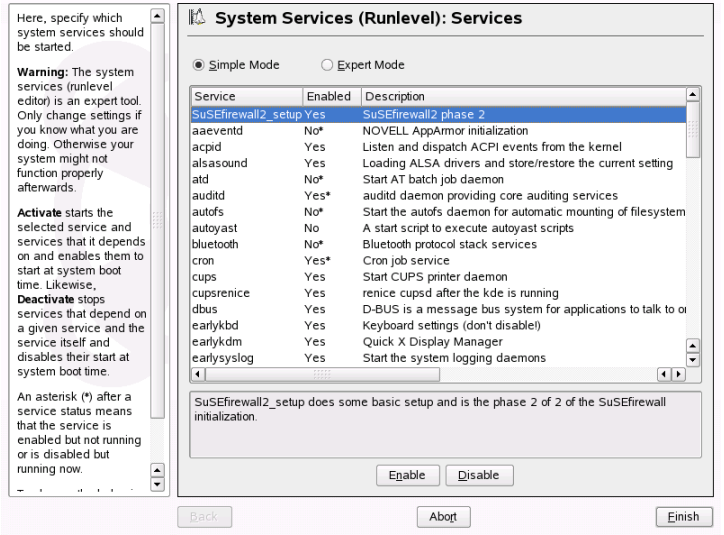
If a script already present in `/etc/init.d/` should be integrated into the existing runlevel scheme, create the links in the runlevel directories right away with insserv or by enabling the corresponding service in the runlevel editor of YaST. Your changes are applied during the next reboot—the new service 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.

20.2.3 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.

Figure 20.1 *System Services (Runlevel)*



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 SUSE Linux Enterprise 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 20.1, “Available Runlevels” (page 394)) 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.

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.

20.3 System Configuration via /etc/sysconfig

The main configuration of SUSE Linux Enterprise 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.

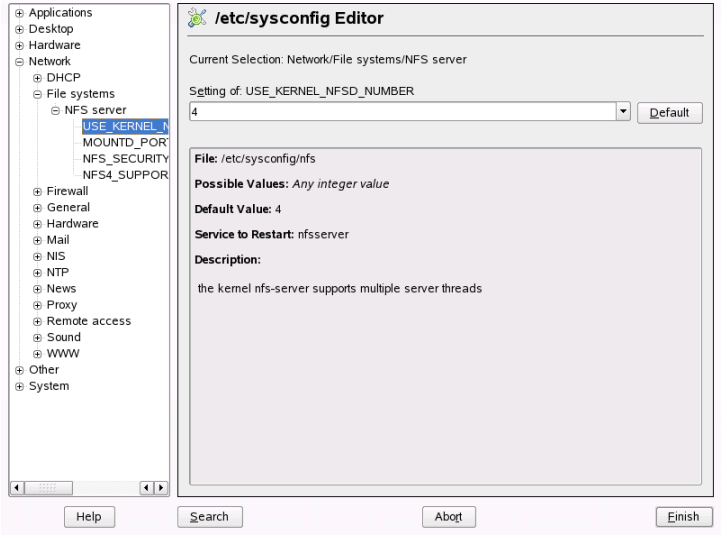
20.3.1 Changing the System Configuration Using the YaST sysconfig Editor

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 20.2 *System Configuration Using the sysconfig Editor*



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.

20.3.2 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 `init 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 `init` *`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.

---

21 The Boot Loader

This chapter describes how to configure GRUB, the boot loader used in SUSE Linux Enterprise®. 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 20, *Booting and Configuring a Linux System* (page 389). A boot loader represents the interface between machine (BIOS) and the operating system (SUSE Linux Enterprise). 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 “Partition Types” (page 156)). 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`).

21.1 Selecting a Boot Loader

By default, the boot loader GRUB is used in SUSE Linux Enterprise. However, in some cases and for special hardware and software constellations, LILO may be necessary. If you update from an older SUSE Linux Enterprise 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`.

21.2 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 412) 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 412)). 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.

21.2.1 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 21.3, “Configuring the Boot Loader with YaST” (page 416).

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 409). 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 410).

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/hda1`.

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 21.2.2, "The File device.map" (page 413).

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/hda5`, a root partition under `/dev/hda7`, and a Windows installation under `/dev/hda1`.

```
gfxmenu (hd0,4)/message
color white/blue black/light-gray
default 0
timeout 8

title linux
   kernel (hd0,4)/vmlinuz root=/dev/hda7 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/hda7 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/hda5` 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 SUSE Linux Enterprise. 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/hda7/`), 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 412).

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 51.1, “US Keyboard Layout” (page 920) 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/hda3 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.

21.2.2 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/hda
(hd1)   /dev/sda
```

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.

IMPORTANT: SATA Disks

Depending on the controller, SATA disks are either recognized as IDE (/dev/hd*x*) or SCSI (/dev/sd*x*) devices.

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
```

21.2.3 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.

21.2.4 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/hda7 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:

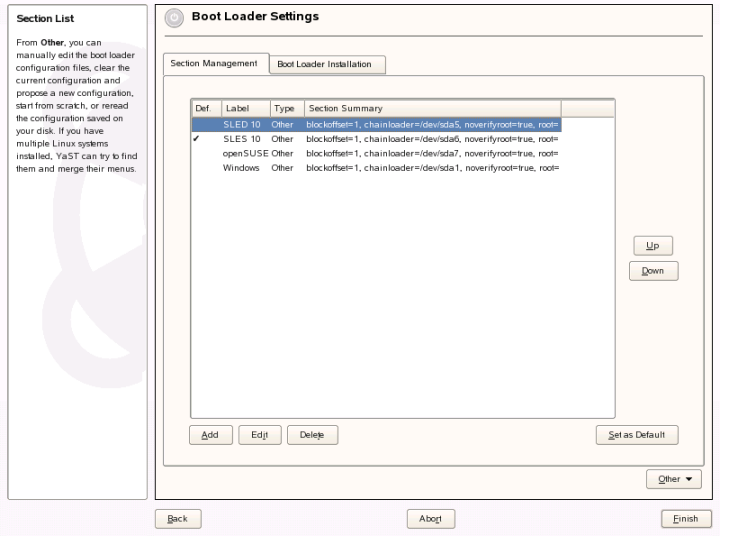
```
Error 32: Must be authenticated
```

   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.

21.3 Configuring the Boot Loader with YaST

The easiest way to configure the boot loader in your SUSE Linux Enterprise system is to use the YaST module. In the YaST Control Center, select *System* > *Boot Loader*. As in Figure 21.1, “Boot Loader Settings” (page 416), this shows the current boot loader configuration of your system and allows you to make changes.

Figure 21.1 *Boot Loader Settings*



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 21.2, “Booting with GRUB” (page 406) 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 21.2, “Booting with GRUB” (page 406) 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*.

21.3.1 Boot Loader Type

Set the boot loader type in *Boot Loader Installation*. The default boot loader in SUSE Linux Enterprise is GRUB. To use LILO, proceed as follows:

Procedure 21.1 *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 SUSE Linux Enterprise.

   Read Configuration Saved on Disk
   : Load your own `/etc/lilo.conf`. This action is not available during the installation of SUSE Linux Enterprise.

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.

21.3.2 Boot Loader Location

To change the location of the boot loader, follow these steps:

Procedure 21.2 *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.

21.3.3 Default System

To change the system that is booted by default, proceed as follows:

Procedure 21.3 *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.

21.3.4 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 21.4 *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.

21.3.5 Security Settings

Using this YaST module, you can also set a password to protect booting. This gives you an additional level of security.

Procedure 21.5 *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.

21.4 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*.

21.5 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 21.6 *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.

21.6 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 21, *The Boot Loader* (page 405) 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.

21.7 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 Knowledge base at `http://support.novell.com/`. 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 `/dev/hda` as `hd0` and `/dev/sda` 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`.

21.8 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 “GRUB” in the Technical Information Search at `http://www.novell.com/support` to get information about special issues.

22 Special System Features

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).

22.1 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.

22.1.1 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
```

Then copy personal adjustments back from the *.old files.

22.1.2 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 22.1, "Entry in /etc/crontab" (page 426), root is entered. Package-specific tables, located in /etc/cron.d, have the same format. See the cron man page (man cron).

Example 22.1 *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 22.2, "/etc/crontab: Remove Time Stamp Files" (page 427), which removes the `hourly` one before every full hour, the `daily` one once a day at 2:14 a.m., etc.).

Example 22.2 */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`.

22.1.3 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 22.3 *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.

22.1.4 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.

22.1.5 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 22.1, "`ulimit`: Setting Resources for the User" (page 429).

Table 22.1 *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 22.4 *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`.

IMPORTANT

Not all shells support `ulimit` directives. PAM (for instance, `pam_limits`) offers comprehensive adjustment possibilities if you depend on encompassing settings for these restrictions.

22.1.6 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`.

22.1.7 The /etc/resolv.conf File

Domain name resolution is handled through the file `/etc/resolv.conf`. Refer to Chapter 33, *The Domain Name System* (page 615).

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.

22.1.8 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.

22.1.9 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 SUSE® Linux Enterprise, 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.

22.2 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.

22.3 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.

22.4 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 20.3.1, “Changing the System Configuration Using the YaST sysconfig Editor” (page 402)). 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>]]
```

22.4.1 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.

22.4.2 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
```

22.4.3 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.

22.4.4 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`.

23 Printer Operation

SUSE Linux Enterprise® supports printing with many types of printers, including remote network printers. Printers can be configured with YaST or manually. 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 23.9, “Troubleshooting” (page 455).

CUPS is the standard print system in SUSE Linux Enterprise. 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 SUSE Linux Enterprise 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 23.9.1, “Printers without Standard Printer Language Support” (page 455) 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 SUSE Linux Enterprise version was released. Thus, the databases may not necessarily indicate the correct status, but only provide an approximation.

23.1 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.

23.2 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`.

► **zseries:** Printers and similar devices provided by the z/VM that you can connect locally with the IBM System z mainframes are not supported by CUPS or LPRng. On

these platforms, printing is only possible over the network. The cabling for network printers must be installed according to the instructions of the printer manufacturer. ◄

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.

23.3 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 SUSE Linux Enterprise, 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 23.8.3, "PPD Files in Various Packages" (page 452) and Section 23.9.2, "No Suitable PPD File Available for a PostScript Printer" (page 456).

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" (page 446)). 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 SUSE Linux Enterprise 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.

23.4 Setting Up a Printer

YaST can be used to configure a local printer that is directly connected to your machine (normally with USB or parallel port) or to set up printing over the network. It is also possible to add PPD (PostScript Printer Description) files for your printer with YaST.

23.4.1 Configuring Local Printers

If an unconfigured local printer is detected, YaST starts automatically to configure it. YaST can configure the printer automatically if the parallel or USB port can be set up automatically and the connected printer can be detected. The printer model must also be listed in the database used during the automatic hardware detection.

If the printer model is unknown or cannot be automatically detected, configure it manually. There are two possible reasons why a printer is not automatically detected:

- The printer does not identify itself correctly. This may apply to very old devices. Try to configure your printer as described in Section "Configuring Manually" (page 443).
- If the manual configuration does not work, communication between printer and computer is not possible. Check the cable and the plugs to make sure that the printer is properly connected. If this is the case, the problem may not be printer-related, but rather a USB or parallel port–related problem.

Configuring Manually

To manually configure the printer, select *Hardware* > *Printer* in the YaST control center. This opens the main *Printer Configuration* window, where the detected devices are listed in the upper part. The lower part lists any queues configured so far (refer to Section 23.1, "The Workflow of the Printing System" (page 441) for more information about print queues). If no printer was detected, both parts of the configuration window are empty. Use *Edit* to change the configuration of a listed printer or *Add* to set up a printer not automatically detected. Editing an existing configuration uses the same dialogs as in Procedure 23.1, "Adding a Local Printer Manually" (page 444).

In *Printer Configuration*, you can also *Delete* an existing entry. Clicking *Other* opens a list with advanced options. Select *Restart Detection*, to manually start the automatic

printer detection. If more than one printer is connected to the machine or more than one queue is configured for a printer, you can mark the active entry as the default. *CUPS Expert Settings* and *Change IPP Listen* are advanced configuration options— refer to Chapter 23, *Printer Operation* (page 439) for details.

Procedure 23.1 *Adding a Local Printer Manually*

TIP: YaST Print Test

To make sure that everything works correctly, the crucial configuration steps can be checked with the print test function of YaST. The test page also provides important information about the configuration tested. If the output is garbled, for example, with several pages almost empty, you can stop the printer by first removing all paper then stopping the test from YaST.

---

1. Start YaST and choose *Hardware* > *Printer* to open the *Printer Configuration* dialog.

2. Click *Add* to open the *Printer Type* window.

3. Choose *Directly Connected Printers*.

4. Select the port to which the printer is connected (usually USB or parallel port) and choose the device in the next configuration screen. It is recommended to *Test the Printer Connection* at this point. If problems occur, select the correct device or choose *Back* to return to the previous dialog.

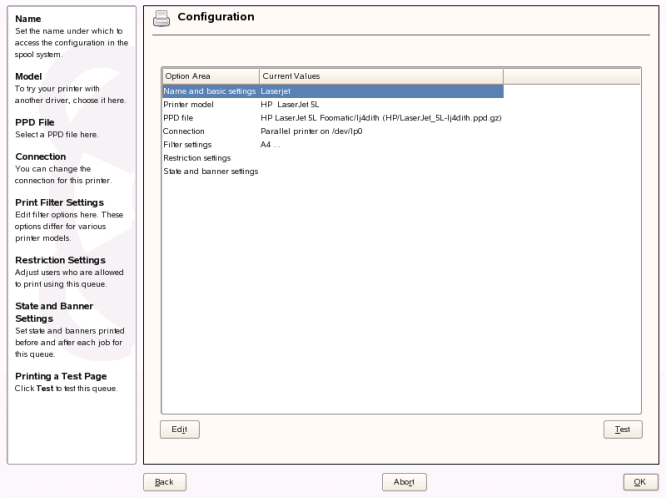
5. In *Queue Name*, set up a print queue. Specifying a *Name for Printing* is mandatory. It is recommended to choose a recognizable name—with this name, you can later identify the printer in the printing dialogs of applications. Use *Printer Description* and *Printer Location* to further describe the printer. This is optional, but useful if you have more than one printer connected to the machine or if you set up a print server. *Do Local Filtering* should be checked—it is needed for local printers.

6. In *Printer Model*, specify the printer by *Manufacturer* and *Model*. If your printer is not listed, you can try *UNKNOWN MANUFACTURER* from the manufacturer list and select an appropriate standard language (the set of commands controlling the printer) from the model list (refer to your printer's documentation to find out

which language your printer understands). If this does not work, refer to Section "Adding PPD Files with YaST" (page 446) for another possible solution.

7 The *Configuration* screen lists a summary of the printer setup. This dialog is also shown when editing an existing printer configuration from the start screen of this YaST module.

Figure 23.1 *Printer Configuration Summary*



The summary contains the following entries, which you can also modify with *Edit*:

- *Name and basic settings*, *Printer Model*, and *Connection* let you change entries made while following this procedure.
- Refer to Section "Choosing an Alternative PPD File with YaST" (page 446) for details on *PPD file*.
- With *Filter settings* fine-tune the printer setup. Configure options like *Page Size*, *Color Mode*, and *Resolution* here.
- By default, every user is able to use the printer. With *Restriction settings*, list users that are forbidden to use the printer or list users that are allowed to use it.

- With *State and banner settings* you can, for example, deactivate the printer by changing its state and specify whether a page with a *Starting Banner* or *Ending Banner* is printed before or after each job (the default is not to print them).

Adding PPD Files with YaST

If your printer does not show up in the *Printer Model* dialog, a PPD (PostScript Printer Description) file for your model is missing (see Section 23.3, “Installing the Software” (page 442) for more information about PPD files). With *Add PPD File to Database*, add a PPD file from the local file system or an FTP or HTTP server.

Get PPD files directly from your printer vendor or from the driver CD of the printer (see Section 23.9.2, “No Suitable PPD File Available for a PostScript Printer” (page 456) for details). An alternative source for PPD files is `http://www.linuxprinting.org/`, the “Linux Printing Database”. When downloading PPD files from linuxprinting.org, keep in mind that it always shows the latest Linux support status, which is not necessarily met by SUSE Linux Enterprise.

Choosing an Alternative PPD File with YaST

For many printer models, several PPD files are available. When configuring the printer, YaST defaults to the one marked `recommended` as a general rule. To get a list of PPD files available for a printer, select *PPD file* in *Configuration* then click *Edit*. See Figure 23.1, “Printer Configuration Summary” (page 445).

Normally it should not be necessary to change the PPD file—the PPD file chosen by YaST should produce the best results. However, if you want a color printer to print only in black and white, for example, it is most convenient to use a PPD file that does not support color printing. If you experience performance problems with a PostScript printer when printing graphics, it may help to switch from a PostScript PPD file to a PCL PPD file (provided your printer understands PCL).

23.4.2 Configuring Network Printers with YaST

Network printers are not detected automatically. They must be configured manually using the YaST printer module. Depending on your network setup, you can print to a print server (CUPS, LPD, SMB, or IPX) or directly to a network printer (preferably via TCP). Ask your network administrator for details on configuring a network printer in your environment.

Procedure 23.2 *Configuring a Network Printer with YaST*

1. Start YaST and choose *Hardware > Printer* to open the *Printer Configuration* dialog.
2. Click *Add* to open the *Printer Type* window.
3. Choose *Network Printers* to open a dialog in which to specify further details that should be provided by your network administrator.

23.5 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
```

23.5.1 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`.

23.6 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.

23.7 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`*.

23.8 Special Features in SUSE Linux Enterprise

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

23.8.1 CUPS and Firewall

After having performed a default installation of SUSE Linux Enterprise, 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 43.4, “SuSEfirewall2” (page 828).

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.

23.8.2 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 452) are vital preconditions for this feature, because otherwise the security would not be sufficient for an automatic activation of `cupsd`.

23.8.3 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.

23.9 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.

23.9.1 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.

23.9.2 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.

23.9.3 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 23.1, "`/etc/modprobe.conf`: Interrupt Mode for the First Parallel Port" (page 457). 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 23.1 */etc/modprobe.conf: Interrupt Mode for the First Parallel Port*

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

23.9.4 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 23.2, “Error Message from `lpd`” (page 458), the problem is caused by the remote `lpd`.

Example 23.2 *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 23.3, “Broadcast from the CUPS Network Server” (page 458).

Example 23.3 *Broadcast from the CUPS Network Server*

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

► **zseries:** Take into account that IBM System z ethernet devices do not receive broadcasts by default. ◄

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 23.5, “Network Printers” (page 447).

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.

23.9.5 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.

23.9.6 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`.

23.9.7 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 for-

warded 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
```

23.9.8 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.

23.9.9 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.

Dynamic Kernel Device Management with udev

24

Since version 2.6, the kernel is capable of adding or removing 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.

24.1 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.

24.2 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.

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.

24.3 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.

24.4 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 for 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 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.

24.5 Debugging udev Events

The program `udevmonitor` can be used to visualize the driver core events and the timing of the udev event processes.

```
UEVENT[1132632714.285362] add@/devices/pci0000:00/0000:00:1d.1/usb2/2-2
UEVENT[1132632714.288166] add@/devices/pci0000:00/0000:00:1d.1/usb2/2-2/2-2:1.0
UEVENT[1132632714.309485] add@/class/input/input6
UEVENT[1132632714.309511] add@/class/input/input6/mouse2
UEVENT[1132632714.309524] add@/class/usb_device/usbdev2.12
```

```
UDEV  [1132632714.348966] add@/devices/pci0000:00/0000:00:1d.1/usb2/2-2
UDEV  [1132632714.420947] add@/devices/pci0000:00/0000:00:1d.1/usb2/2-2/2-2:1.0
UDEV  [1132632714.427298] add@/class/input/input6
UDEV  [1132632714.434223] add@/class/usb_device/usbdev2.12
UDEV  [1132632714.439934] add@/class/input/input6/mouse2
```

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:

```
UDEV  [1132633002.937243] add@/class/input/input7
UDEV_LOG=3
ACTION=add
DEVPATH=/class/input/input7
SUBSYSTEM=input
SEQNUM=1043
PHYSDEVPATH=/devices/pci0000:00/0000:00:1d.1/usb2/2-2/2-2:1.0
PHYSDEVBUS=usb
PHYSDEVDRIVER=usbhid
PRODUCT=3/46d/c03e/2000
NAME="Logitech USB-PS/2 Optical Mouse"
PHYS="usb-0000:00:1d.1-2/input0"
UNIQ=""
EV=7
KEY=70000 0 0 0 0 0 0 0 0
REL=103
```

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`*.

24.6 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. The rule syntax and the provided keys to match or import data are described in the udev man page.

24.7 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
```

24.8 The Replaced hotplug Package

The formerly used hotplug package is entirely replaced by udev and the udev-related kernel infrastructure. The following parts of the former hotplug infrastructure have been made obsolete or had their functionality taken over by udev:

`/etc/hotplug/*.agent`
: No longer needed or moved to `/lib/udev`

`/etc/hotplug/*.rc`
: Replaced by the `/sys/*/uevent` trigger

`/etc/hotplug/blacklist`
: Replaced by the `blacklist` option in `modprobe.conf`

`/etc/dev.d/*`
: Replaced by the udev rule `RUN` key

`/etc/hotplug.d/*`
: Replaced by the udev rule `RUN` key

`/sbin/hotplug`
: Replaced by udevd listening to netlink; only used in the initial RAM file system until the root file system can be mounted, then it is disabled

`/dev/*`
: Replaced by dynamic udev and static content in `/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

24.9 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.

25 File Systems in Linux

SUSE Linux Enterprise® 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. To meet the requirements of high-performance clustering scenarios, SUSE Linux Enterprise Server includes OCFS2 (Oracle Cluster File System 2).

25.1 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.

25.2 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.

25.2.1 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 25.2.3, "Ext3" (page 474). The default mode is `data=ordered`, which ensures both data and metadata integrity, but uses journaling only for metadata.

25.2.2 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.

25.2.3 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.

25.2.4 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.

25.2.5 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.

25.2.6 Oracle Cluster File System 2

OCFS2 is a journaling file system that has been tailor-made for clustering setups. In contrast to a standard single-node file system like Ext3, OCFS2 is capable of managing several nodes. OCFS2 allows spreading a file system across shared storage, such as a SAN or multipath setup.

Every node in an OCFS2 setup has concurrent read and write access to all data. This requires OCFS2 to be cluster-aware, meaning that OCFS2 must include a means to determine of which nodes the cluster consists and whether these nodes are actually alive and available. To compute a cluster's membership, OCFS2 includes a node manager (NM). To monitor the availability of the nodes in a cluster, OCFS2 includes a simple heartbeat implementation. To avoid chaos arising from various nodes directly accessing the file system, OCFS2 also contains a lock manager, DLM (distributed lock manager). Communication between the nodes is handled via a TCP-based messaging system.

Major features and benefits of OCFS2 include:

- Metadata caching and journaling
- Asynchronous and direct I/O support for database files for improved database performance
- Support for multiple block sizes (where each volume can have a different block size) up to 4 KB, for a maximum volume size of 16 TB
- Cross-node file data consistency
- Support for up to 255 cluster nodes

For more in-depth information about OCFS2, refer to Chapter 14, *Oracle Cluster File System 2* (page 289).

25.3 Some Other Supported File Systems

Table 25.1, “File System Types in Linux” (page 478) 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 25.1 *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.

25.4 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 25.2, "Maximum Sizes of File Systems (On-Disk Format)" (page 479) offers an overview of the current limitations of Linux files and file systems.

Table 25.2 *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^{44}-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 25.2, “Maximum Sizes of File Systems (On-Disk Format)” (page 479) 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.

25.5 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://chichkin_i.zelnet.ru/namesys/
- 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.

26 The X Window System

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 SUSE Linux Enterprise®.

TIP: IBM System z: Configuring the Graphical User Interface

IBM System z do not have any input and output devices supported by X.Org. Therefore, none of the configuration procedures described in this section apply. More relevant information for IBM System z can be found in Section 8.6, "Network Devices" (page 164).

26.1 Manually Configuring the X Window System

By default, the X Window System is configured with the SaX2 interface, described in Section 8.14, "SaX2" (page 191). 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 SUSE Linux Enterprise 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.

IMPORTANT: Using X -configure

Use `X -configure` to configure your X setup if previous tries with SUSE Linux Enterprise's SaX2 have failed. If your setup involves proprietary binary-only drivers, `X -configure` cannot work.

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 26.1, “Sections in /etc/X11/xorg.conf” (page 485).

Table 26.1 *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 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.

Type	Meaning
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.

26.1.1 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 26.1, “Screen Section of the File /etc/X11/xorg.conf” (page 487).

Example 26.1 *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.

26.1.2 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/X11R6/lib/modules/drivers` or `/usr/X11R6/lib64/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.

26.1.3 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/X11R6/lib/X11/doc/` (the package `xorg-x11-doc` must be installed).

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.

26.2 Installing and Configuring Fonts

The installation of additional fonts in SUSE Linux Enterprise is very easy. Simply copy the fonts to any directory located in the X11 font path (see Section 26.2.1, “X11 Core Fonts” (page 491)). The installation directory should be a subdirectory of the directories configured in `/etc/fonts/fonts.conf` (see Section 26.2.2, “Xft” (page 492)) 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>
<dir>~/.fonts/kde-override</dir>
<include ignore_missing="yes">suse-font-dirs.conf</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/lib/ooo-2.0/share/fonts</dir>
<dir>/usr/lib/ooo-2.0/share/fonts/truetype</dir>
<dir>/usr/lib/jvm/java-1.5.0-sun-1.5.0_update10/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.

26.2.1 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, SUSE Linux Enterprise 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 SUSE Linux Enterprise contain at least the glyphs needed for European languages (formerly encoded as `iso-8859-*`).

26.2.2 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 SUSE Linux Enterprise, 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 26.2 *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.

26.3 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`.

27 Authentication with PAM

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`, `pam_pwcheck.conf`, `pam_unix2.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.

27.1 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 SUSE Linux Enterprise®, 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).

27.2 The PAM Configuration of sshd

To show how the theory behind PAM works, consider the PAM configuration of sshd as a practical example:

Example 27.1 *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 27.2, “Default Configuration for the `auth` Section” (page 500).

Example 27.2 *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 27.3, "Default Configuration for the `account` Section" (page 501). `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 27.4, "Default Configuration for the `password` Section" (page 501).

Example 27.3 *Default Configuration for the account Section*

```
account required        pam_unix2.so
```

Example 27.4 *Default Configuration for the password Section*

```
password required       pam_pwcheck.so  nullok
password required       pam_unix2.so    nullok use_first_pass 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 27.5 *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.

Although pam_unix2 is processed again, it has no practical consequences due to its none option specified in the respective configuration file of this module, pam_unix2.conf. The pam_limits module loads the file /etc/security/limits.conf, which may define limits on the use of certain system resources. The session modules are called a second time when the user logs out.

27.3 Configuration of PAM Modules

Some of the PAM modules are configurable. The corresponding configuration files are located in /etc/security. This section briefly describes the configuration files relevant to the sshd example—pam_unix2.conf, pam_env.conf, pam_pwcheck.conf, and limits.conf.

27.3.1 pam_unix2.conf

The traditional password-based authentication method is controlled by the PAM module pam_unix2. It can read the necessary data from /etc/passwd, /etc/shadow, NIS maps, NIS+ tables, or an LDAP database. The behavior of this module can be influenced by configuring the PAM options of the individual application itself or globally by editing /etc/security/pam_unix2.conf. A very basic configuration file for the module is shown in Example 27.6, “pam_unix2.conf” (page 502).

Example 27.6 *pam_unix2.conf*

```
auth:   nullok
account:
password:       nullok
session:        none
```

The nullok option for module types auth and password specifies that empty passwords are permitted for the corresponding type of account. Users are also allowed to change passwords for their accounts. The none option for the module type session specifies that no messages are logged on its behalf (this is the default). Learn about additional configuration options from the comments in the file itself and from the manual page pam_unix2(8).

27.3.2 pam_env.conf

This file can be used to define a standardized environment for users that is set whenever the pam_env module is called. With it, preset environment variables using the following syntax:

```
VARIABLE  [DEFAULT=[value]]  [OVERRIDE=[value]]
```

VARIABLE
: Name of the environment variable to set.

[DEFAULT=[value]]
: Default value the administrator wants set.

[OVERRIDE=[value]]
: Values that may be queried and set by pam_env, overriding the default value.

A typical example of how pam_env can be used is the adaptation of the DISPLAY variable, which is changed whenever a remote login takes place. This is shown in Example 27.7, “pam_env.conf” (page 503).

Example 27.7 *pam_env.conf*

```
REMOTEHOST     DEFAULT=localhost OVERRIDE=@{PAM_RHOST}
DISPLAY        DEFAULT=${REMOTEHOST}:0.0 OVERRIDE=${DISPLAY}
```

The first line sets the value of the REMOTEHOST variable to localhost, which is used whenever pam_env cannot determine any other value. The DISPLAY variable in turn contains the value of REMOTEHOST. Find more information in the comments in the file /etc/security/pam_env.conf.

27.3.3 pam_pwcheck.conf

This configuration file is for the pam_pwcheck module, which reads options from it for all password type modules. Settings stored in this file take precedence over the PAM settings of an individual application. If application-specific settings have not been defined, the application uses the global settings. Example 27.8, “pam_pwcheck.conf” (page 504) tells pam_pwcheck to allow empty passwords and modification of pass-

words. More options for the module are mentioned in the file `/etc/security/pam_pwcheck.conf`.

Example 27.8 *pam_pwcheck.conf*

```
password:    nullok
```

27.3.4 limits.conf

System limits can be set on a user or group basis in the file `limits.conf`, which is read by the `pam_limits` module. The file allows you to set hard limits, which may not be exceeded at all, and soft limits, which may be exceeded temporarily. To learn about the syntax and the available options, read the comments included in the file.

27.4 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.

Thorsten Kukuk has developed a number of PAM modules and made some information available about them at http://www.suse.de/~kukuk/pam/.

28 Power Management

Power management is especially important on laptop computers, but is also useful on other systems. Two technologies are available: APM (advanced power management) and ACPI (advanced configuration and power interface). In addition to these, it is also possible to control CPU frequency scaling to save power or decrease noise. These options can be configured manually or using a special YaST module.

► **zseries:** The features and hardware described in this chapter do not exist on IBM System z, making this chapter irrelevant for these platforms. ◄

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.

APM had been used in many older computers. Because APM largely consists of a function set implemented in the BIOS, the level of APM support may vary depending on the hardware. This is even more true of ACPI, which is even more complex. For this reason, it is virtually impossible to recommend one over the other. Simply test the various procedures on your hardware then select the technology that is best supported.

28.1 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 the power management systems APM and 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 APM and ACPI.

Battery Monitor
: ACPI and APM check the battery charge status and provide information about it. Additionally, both systems coordinate 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.

28.2 APM

Some of the power saving functions are performed by the APM BIOS itself. On many laptops, standby and suspend states can be activated with key combinations or by closing the lid without any special operating system function. However, to activate these modes with a command, certain actions must be triggered before the system is suspended. To view the battery charge level, you need special program packages and a suitable kernel.

SUSE Linux Enterprise® kernels have built-in APM support. However, APM is only activated if ACPI is not implemented in the BIOS and an APM BIOS is detected. To activate APM support, ACPI must be disabled with `acpi=off` at the boot prompt. Enter `cat /proc/apm` to check if APM is active. An output consisting of various numbers indicates that everything is OK. You should now be able to shut down the computer with the command `shutdown -h`.

BIOS implementations that are not fully standard-compliant can cause problems with APM. Some problems can be circumvented with special boot parameters. All parameters are entered at the boot prompt in the form of `apm=`*`parameter`* with *`parameter`* being one of:

on or off
: Enable or disable APM support.

(no-)allow-ints
: Allow interrupts during the execution of BIOS functions.

(no-)broken-psr
: The “GetPowerStatus” function of the BIOS does not work properly.

(no-)realmode-power-off
: Reset processor to real mode prior to shutdown.

(no-)debug
: Log APM events in system log.

(no-)power-off
: Power system off after shutdown.

bounce-interval=*n*
: Time in hundredths of a second after a suspend event during which additional suspend events are ignored.

idle-threshold=*n*
: System inactivity percentage from which the BIOS function `idle` is executed (0=always, 100=never).

idle-period=*n*
: Time in hundredths of a second after which the system activity is measured.

The APM daemon (apmd) is no longer used. Its functionality is now handled by the new powersaved, which also supports ACPI and provides many other features.

28.3 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 28.3.4, "Troubleshooting" (page 516) for more information about troubleshooting ACPI problems.

28.3.1 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 28.3.3, “ACPI Tools” (page 516). 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 28.3.3, “ACPI Tools” (page 516).

28.3.2 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 manufac-

turers. 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 APM or 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 SUSE Linux Enterprise 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 SUSE Linux Enterprise these technologies are controlled by the powersave daemon. The configuration is explained in Section 28.5, “The powersave Package” (page 519).

28.3.3 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`).

28.3.4 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 28.5.4, “Troubleshooting” (page 526).

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

Additional documentation and help on ACPI:

- 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)

28.4 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. Most of the functions can be controlled with powersaved and the YaST power management module, which is discussed in further detail in Section 28.6, “The YaST Power Management Module” (page 528).

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.

28.5 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, APM, 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.

28.5.1 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
- do_suspend_to_disk
- do_suspend_to_ram

- do_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.

set_disk_settings
: Executes the disk settings made in `/etc/sysconfig/powersave/disk`.

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/sleep`
: In this file, activate the sleep modes and determine which critical modules should be unloaded and which services should be stopped prior to a suspend or standby event. When the system is resumed, these modules are reloaded and the services are restarted. You can even delay a triggered sleep mode, for example, to save files. The default settings mainly concern USB and PCMCIA modules. A failure of suspend or standby is usually caused by certain modules. See Section 28.5.4, “Troubleshooting” (page 526) for more information about identifying the error.

`/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/disk`
: This configuration file controls the actions and settings made regarding the hard disk.

`/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.

28.5.2 Configuring APM and ACPI

Suspend and Standby

There are three basic ACPI sleep modes and two APM sleep modes:

Suspend to Disk (ACPI S4, APM suspend)
: Saves the entire memory content to the hard disk. The computer is switched off completely and does not consume any power. This sleep mode is enabled by default and should work on all systems.

Suspend to RAM (ACPI S3, APM suspend)
: Saves the states of all devices to the main memory. Only the main memory continues consuming power. SUSE Linux Enterprise does not generally support this sleep mode although you can use it for quite a number of machines.

  This sleep mode is enabled by default, but it is only *executed* if the current machine is listed in a database as capable of supporting this mode. This database is contained in the `/usr/sbin/s2ram` binary provided by the `suspend` package.

  To modify the default parameters (for example, to generally disable the `suspend to ram` sleep mode or to force it even for machines not listed in the database), find more information about available options in the `/etc/sysconfig/powersave/sleep` configuration file.

  To learn more about the `s2ram` binary, refer to the README files in `/usr/share/doc/packages/suspend`.

Standby (ACPI S1, APM standby)
: Switches some devices off (manufacturer-dependent).

Make sure that the following default options are set in the file `/etc/sysconfig/powersave/events` for the correct processing of suspend, standby, and resume (default settings following the installation of SUSE Linux Enterprise):

```
EVENT_GLOBAL_SUSPEND2DISK=
    "prepare_suspend_to_disk screen_saver do_suspend_to_disk"
EVENT_GLOBAL_SUSPEND2RAM=
    "prepare_suspend_to_ram screen_saver do_suspend_to_ram"
EVENT_GLOBAL_STANDBY=
```

```
    "prepare_standby screen_saver do_standby"
EVENT_GLOBAL_RESUME_SUSPEND2DISK=
    "restore_after_suspend_to_disk"
EVENT_GLOBAL_RESUME_SUSPEND2RAM=
    "restore_after_suspend_to_ram"
EVENT_GLOBAL_RESUME_STANDBY=
    "restore_after_standby"
```

Custom Battery States

In the file `/etc/sysconfig/powersave/battery`, define three battery charge levels (in percent) that trigger system alerts or specific actions when they are reached.

```
BATTERY_WARNING=12
BATTERY_LOW=7
BATTERY_CRITICAL=2
```

The actions or scripts to execute when the charge levels drop under the specified limits are defined in the configuration file `/etc/sysconfig/powersave/events`. The standard actions for buttons can be modified as described in Section 28.5.1, “Configuring the powersave Package” (page 520).

```
EVENT_BATTERY_NORMAL="ignore"
EVENT_BATTERY_WARNING="notify"
EVENT_BATTERY_LOW="notify"
EVENT_BATTERY_CRITICAL="wm_shutdown"
```

Adapting Power Consumption to Various Conditions

The system behavior can be adapted to the type of power supply. The power consumption of the system should be reduced when the system is disconnected from the AC power supply and operated with the battery. Similarly, the performance should automatically increase as soon as the system is connected to the AC power supply. The CPU frequency, the power saving function of IDE, and a number of other parameters can be modified.

The actions to execute when the computer is disconnected from or connected to the AC power supply are defined in `/etc/sysconfig/powersave/events`. Select the schemes to use in `/etc/sysconfig/powersave/common`:

```
AC_SCHEME="performance"
BATTERY_SCHEME="powersave"
```

The schemes are stored in files in `/etc/sysconfig/powersave`. The filenames are in the format `scheme_name-of-the-scheme`. The example refers to two schemes: `scheme_performance` and `scheme_powersave`. `performance`, `powersave`, `presentation`, and `acoustic` are preconfigured. Existing schemes can be edited, created, deleted, or associated with different power supply states with the help of the YaST power management module described in Section 28.6, "The YaST Power Management Module" (page 528).

28.5.3 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.

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.

28.5.4 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.

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/*`. If a special module or module option is needed, configure it in the file `/etc/sysconfig/powersave/cpufreq` by means of the variables `CPUFREQD_MODULE` and `CPUFREQD_MODULE_OPTS`.

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.

On ACPI and APM systems: 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 files generated by the powersave daemon in `/var/log/suspend2ram.log` and `/var/log/suspend2disk.log` are very helpful in this regard. If the computer does not enter the sleep mode, the cause lies in the last module unloaded. Manipulate the following settings in `/etc/sysconfig/powersave/sleep` to unload problematic modules prior to a suspend or standby.

```
UNLOAD_MODULES_BEFORE_SUSPEND2DISK=""
UNLOAD_MODULES_BEFORE_SUSPEND2RAM=""
UNLOAD_MODULES_BEFORE_STANDBY=""
SUSPEND2DISK_RESTART_SERVICES=""
SUSPEND2RAM_RESTART_SERVICES=""
STANDBY_RESTART_SERVICES=""
```

If you use suspend or standby in changing network environments or in connection with remotely mounted file systems, such as Samba and NIS, use automounter to mount them or add the respective services, for example, `smbfs` or `nfs`, in the above-mentioned variable. If an application accesses the remotely mounted file system prior to a suspend or standby, the service cannot be stopped correctly and the file system cannot be unmounted properly. After resuming the system, the file system may be corrupt and must be remounted.

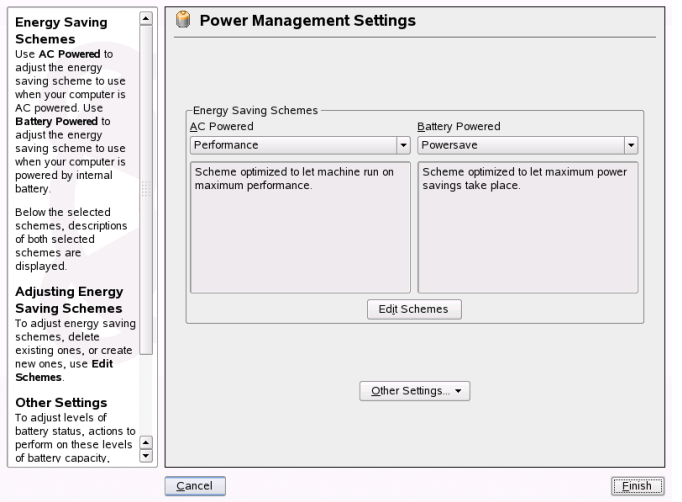
28.5.5 For More Information

- `/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

28.6 The YaST Power Management Module

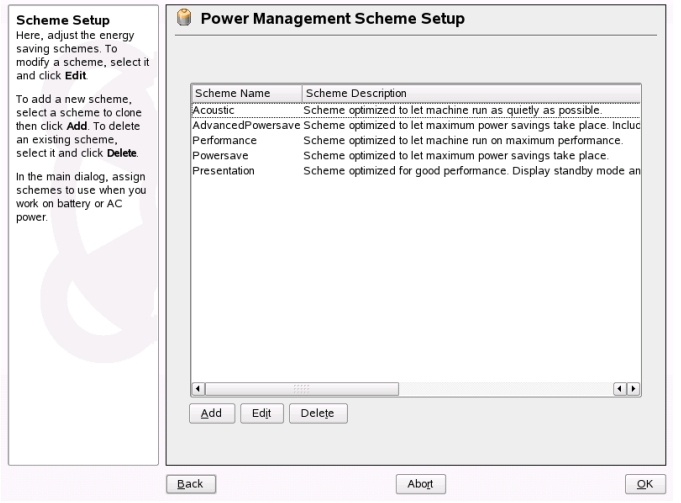
The YaST power management module can configure all power management settings already described. When started from the YaST Control Center with *System* > *Power Management*, the first dialog of the module opens (see Figure 28.1, “Scheme Selection” (page 528)).

Figure 28.1 *Scheme Selection*



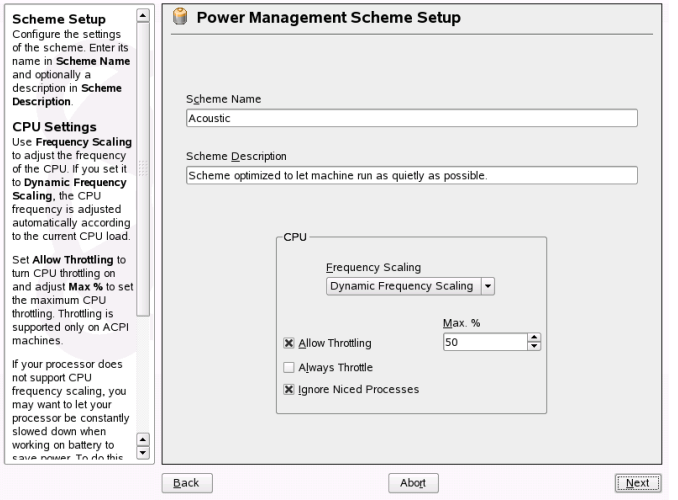
In this dialog, select the schemes to use for battery operation and AC operation. To add or modify the schemes, click *Edit Schemes*, which opens an overview of the existing schemes like that shown in Figure 28.2, “Overview of Existing Schemes” (page 529).

Figure 28.2 *Overview of Existing Schemes*



In the scheme overview, select the scheme to modify then click *Edit*. To create a new scheme, click *Add*. The dialog that opens is the same in both cases and is shown in Figure 28.3, “Configuring a Scheme” (page 530).

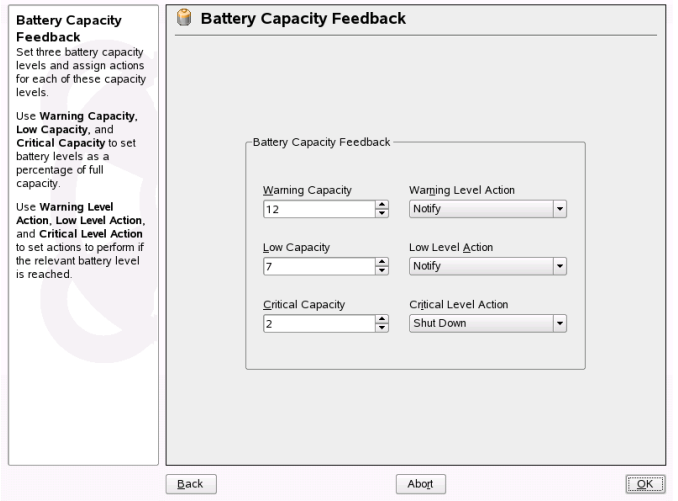
Figure 28.3 *Configuring a Scheme*



First, enter a suitable name and description for the new or edited scheme. Determine if and how the CPU performance should be controlled for this scheme. Decide if and to what extent frequency scaling and throttling should be used and whether processes with low priority (*niced* processes) should be ignored when adjusting the CPU frequency. In the following dialog for the hard disk, define a *Standby Policy* for maximum performance or for energy saving. The *Acoustic Policy* controls the noise level of the hard disk (supported by few hard disks). The *Cooling Policy* determines the cooling method to use. Unfortunately, this type of thermal control is rarely supported by the BIOS. Read `/usr/share/doc/packages/powersave/powersave_manual.html #Thermal` to learn how you can use the fan and passive cooling methods.

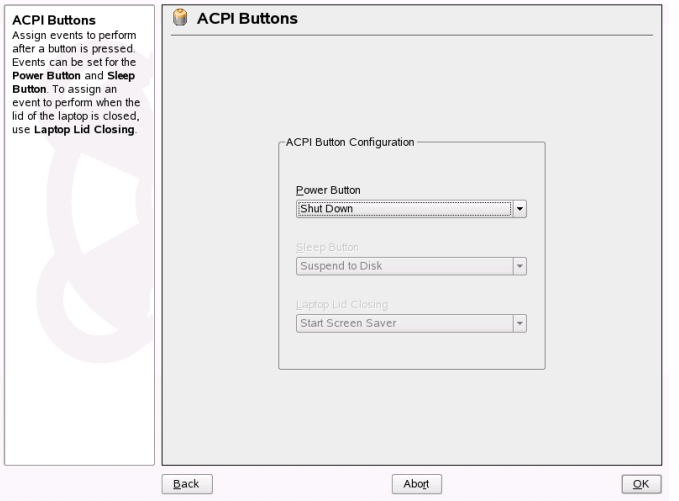
Global power management settings can also be made from the initial dialog using *Battery Warning*, *ACPI Settings*, or *Suspend Permissions*. Access these controls by clicking *Other Settings* and selecting the appropriate item from the menu. Click *Battery Warning* to access the dialog for the battery charge level, shown in Figure 28.4, “Battery Charge Level” (page 531).

Figure 28.4 *Battery Charge Level*



The BIOS of your system notifies the operating system whenever the charge level drops under certain configurable limits. In this dialog, define three limits: *Warning Capacity*, *Low Capacity*, and *Critical Capacity*. Specific actions are triggered when the charge level drops under these limits. Usually, the first two states merely trigger a notification to the user. The third critical level triggers a shutdown, because the remaining energy is not sufficient for continued system operation. Select suitable charge levels and the desired actions then click *OK* to return to the start dialog.

Figure 28.5 *ACPI Settings*



Access the dialog for configuring the ACPI buttons using *ACPI Settings*. It is shown in Figure 28.5, “ACPI Settings” (page 532). The settings for the ACPI buttons determine how the system should respond to certain switches. Configure the system response to pressing the power button, pressing the sleep button, and closing the laptop lid. Click *OK* to complete the configuration and return to the start dialog.

Click *Enable Suspend* to enter a dialog in which to determine if and how users of this system may use the suspend or standby functionality. Click *OK* to return to the main dialog. Click *OK* again to exit the module and confirm your power management settings.

29 Wireless Communication

Wireless LAN can be used to establish communication between your SUSE Linux Enterprise® machines. This chapter introduces the principles of wireless networking and the basic configuration for wireless networking.

29.1 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:

Table 29.1 *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

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.

29.1.1 Hardware

802.11 cards are not supported by SUSE Linux Enterprise®. 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:

- Aironet 4500, 4800
- Atheros 5210, 5211, 5212
- Atmel at76c502, at76c503, at76c504, at76c506
- Intel PRO/Wireless 2100, 2200BG, 2915ABG, 3945ABG
- Intersil Prism2/2.5/3
- Intersil PrismGT
- Lucent/Agere Hermes

- 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 SUSE Linux Enterprise 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`.

29.1.2 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 542)), 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 537)) 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 en-

terprises. 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.

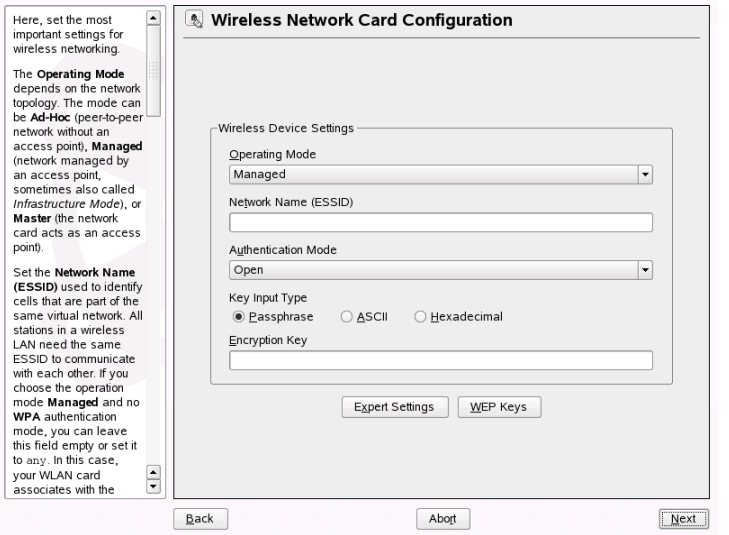
TKIP (defined in WPA/IEEE 802.11i)
: This key management protocol defined in the WPA standard uses the same encryption algorithm as WEP, but eliminates its weakness. Because a new key is generated for every data packet, attacks against these keys are in vain. TKIP is used together with WPA-PSK.

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.

29.1.3 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 29.1, "YaST: Configuring the Wireless Network Card" (page 538), make the basic settings for the WLAN operation:

Figure 29.1 *YaST: Configuring the Wireless Network Card*



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 537) and Section “Security” (page 542) 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 ac-

cording 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.

29.1.4 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.

29.1.5 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. When this documentation was prepared, WPA only worked with cards using Atheros, Intel PRO/Wireless, or Prism2/2.5/3 chips. On Prism2/2.5/3, WPA only works if the hostap driver is used (see Section "Problems with Prism2 Cards" (page 542)). 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.

29.1.6 Troubleshooting

If your WLAN card fails to respond, check if you have downloaded the needed firmware. Refer to Section 29.1.1, "Hardware" (page 534). 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`.

WPA

WPA support is quite new in SUSE Linux Enterprise and still under development. Thus, YaST does not support the configuration of all WPA authentication methods. Not all wireless LAN cards and drivers support WPA. Some cards need a firmware update to enable WPA. If you want to use WPA, read `/usr/share/doc/packages/wireless-tools/README.wpa`.

29.1.7 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.

Part IV. Services

30 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 30.1, “Several Protocols in the TCP/IP Protocol Family” (page 548) 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.”

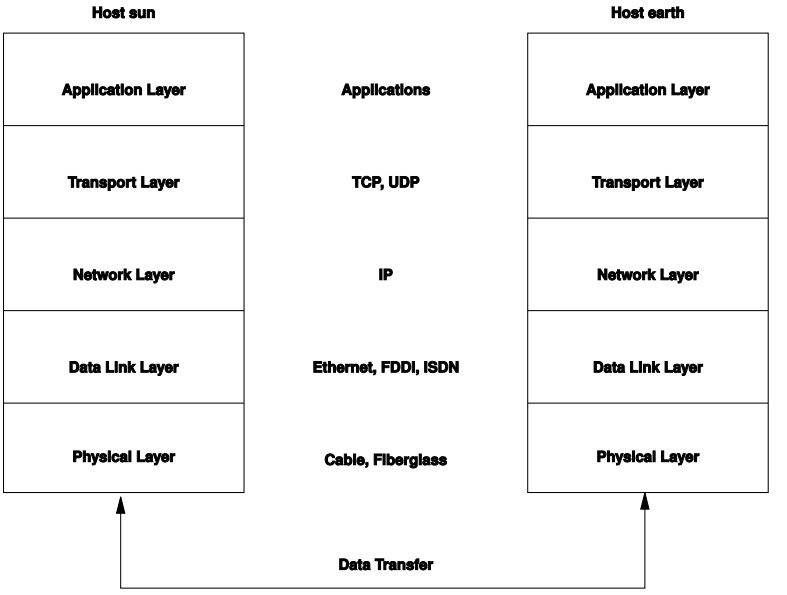
RFC stands for *Request for Comments*. RFCs are documents that describe various Internet protocols and implementation procedures for the operating system and its applications. The RFC documents describe the setup of Internet protocols. To expand your knowledge about any of the protocols, refer to the appropriate RFC documents. They are available online at `http://www.ietf.org/rfc.html`.

Table 30.1 *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 30.1, “Simplified Layer Model for TCP/IP” (page 549), 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 30.1 *Simplified Layer Model for TCP/IP*



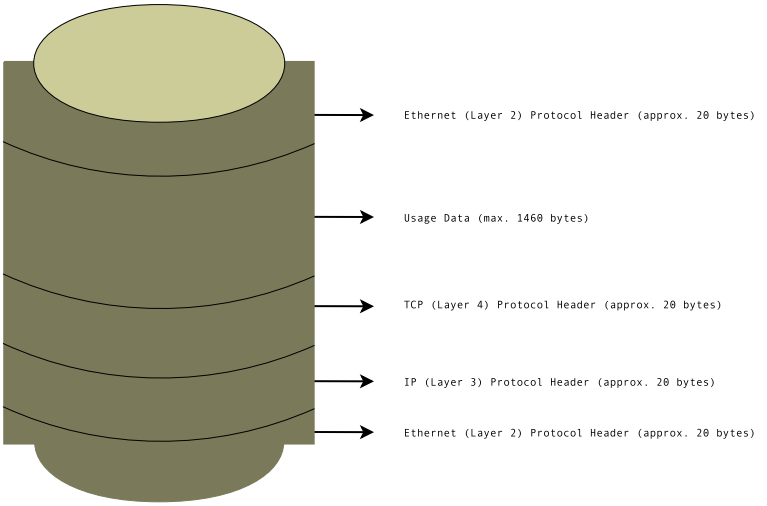
The diagram provides one or two examples for each layer. The layers are ordered according to *abstraction levels*. The lowest layer is very close to the hardware. The uppermost layer, however, is almost a complete abstraction from the hardware. Every layer has its own special function. The special functions 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 30.2, “TCP/IP Ethernet Packet” (page 550). The proof sum is

located at the end of the packet, not at the beginning. This simplifies things for the network hardware.

Figure 30.2 *TCP/IP Ethernet Packet*



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.

30.1 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 30.2, "IPv6—The Next Generation Internet" (page 554).

30.1.1 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 30.1, "Writing IP Addresses" (page 551).

Example 30.1 *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.

30.1.2 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 30.2, "Linking IP Addresses to the Netmask" (page 552). 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 30.2, "Linking IP Addresses to the Netmask" (page 552) the first net with 24 bits could also be written as `192.168.0.0/24`.

Example 30.2 *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 30.2 *Specific Addresses*

Address Type	Description
Base Network Address	This is the netmask AND any address in the network, as shown in Example 30.2, "Linking IP Addresses to the Netmask" (page 552) 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.
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 30.3, "Private IP Address Domains" (page 553).

Table 30.3 *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`

30.2 IPv6—The Next Generation Internet

IMPORTANT: IBM System z: IPv6 Support

IPv6 is not supported by the CTC and IUCV network connections of the IBM System z hardware.

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.

30.2.1 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 30.2.2, “Address Types and Structure” (page 556).

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 30.2.3, “Coexistence of IPv4 and IPv6” (page 561). 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*).

30.2.2 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 30.3, “Sample IPv6 Address” (page 557), where all three lines represent the same address.

Example 30.3 *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 30.4, “IPv6 Address Specifying the Prefix Length” (page 558), 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 30.4 *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 30.4, “Various IPv6 Prefixes” (page 558).

Table 30.4 *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).
`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 30.2.3, "Coexistence of IPv4 and IPv6" (page 561)) 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.

30.2.3 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 30.2.2, "Address Types and Structure" (page 556)).

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.

30.2.4 Configuring IPv6

To configure IPv6, you do not normally need to make any changes on the individual workstations. IPv6 is enabled by default. You can disable it during installation in the network configuration step described in Section 3.14.3, “Network Configuration” (page 38). To disable or enable IPv6 on an installed system, use YaST *Network Card*. Do not change the method and click *Next*. Then select a card and click *Advanced* > *IPv6* in the *Address* tab. 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.

30.2.5 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).

30.3 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 SUSE Linux Enterprise® is described in Chapter 33, *The Domain Name System* (page 615).

The protocol `whois` is closely related to DNS. With this program, quickly find out who is responsible for any given domain.

NOTE: MDNS and .local Domain Names

The `.local` top level domain is treated as link-local domain by the resolver. DNS requests are send as multicast DNS requests instead of normal DNS requests. If you already use the `.local` domain in your nameserver configuration, you must switch this option off in `/etc/host.conf`. Also read the `host.conf` manual page.

If you want to switch off MDNS during installation, use `nomdns=1` as a boot parameter.

For more information on multicast DNS, see `http://www.multicastdns.org`.

30.4 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 30.7, “Configuring a Network Connection Manually” (page 586).

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 SUSE Linux Enterprise.

TIP: IBM System z: Hotpluggable Network Cards

On IBM System z platforms, hotpluggable network cards are supported, but not their automatic network integration via DHCP (as is the case on the PC). After detection, manually configure the interface.

30.4.1 Configuring the Network Card with YaST

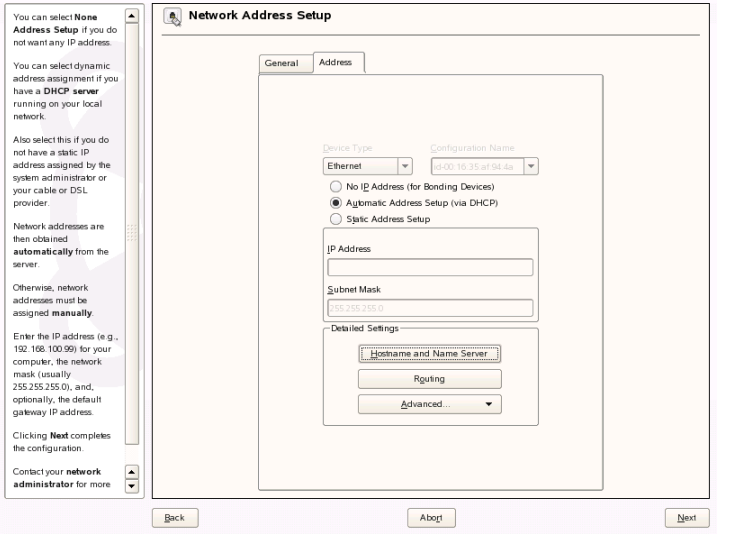
To configure your network wired or wireless card in YaST, select *Network Devices* > *Network Card*. After starting the module, YaST displays a general network configuration dialog. Choose whether to use YaST or NetworkManager to manage all your network devices. If you want to configure your network in the traditional way with the YaST, check *Traditional Method with ifup* and click *Next*. To use NetworkManager, check *User Controlled with NetworkManager* and click *Next*. Find detailed information about NetworkManager in Section 30.6, “Managing Network Connections with NetworkManager” (page 584).

NOTE: Network Method and Xen

NetworkManager does not work with Xen. Only *Traditional Method with ifup* is available in Xen.

The upper part of the next dialog shows a list with all the network cards available for configuration. Any card properly detected is listed with its name. To change the configuration of the selected device, click *Edit*. Devices that could not be detected can be configured using *Add* as described in Section “Configuring an Undetected Network Card” (page 571).

Figure 30.3 *Configuring a Network Card*



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 YaST network card configuration module and click *Edit*. The *Network Address Setup* dialog appears in which to adjust the card configuration using the *Address* and *General* tabs. For information about wireless card configuration, see Section 29.1.3, “Configuration with YaST” (page 538).

Configuring IP Addresses

When possible, wired network cards available during installation are automatically configured to use automatic address setup, DHCP.

NOTE: IBM System z and DHCP

On IBM System z platforms, DHCP-based address configuration is only supported with network cards that have a MAC address. This is only the case with OSA and OSA Express cards.

DHCP should also be used for a DSL line with no static IP assigned by the ISP. If you decide to use DHCP, configure the details in *DHCP Client Options*. Find this dialog from the *Address* tab by selecting *Advanced* > *DHCP Options*. Specify whether the DHCP server should always honor broadcast requests and any identifier to use. If you have a virtual host setup where different hosts communicate through the same interface, an 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 YaST network card configuration module and click *Edit*.
2. In the *Address* tab, choose *Static Address Setup*.
3. Enter *IP Address* and *Subnet Mask*.
4. Click *Next*.
5. To activate the configuration, click *Finish*.

If you use the static address, name servers and a default gateway are not configured automatically. To configure a gateway, click *Routing* and add the default gateway. To configure name servers, click *Hostname and Name Server* and add addresses of name servers and domains.

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 YaST network card configuration module and click *Edit*.

2 In the *Address* tab, choose *Advanced* > *Additional Addresses*.

3 Click *Add*.

4 Enter *Alias Name*, *IP Address*, and *Netmask*.

5 Click *OK*.

6 Click *OK* again.

7 Click *Next*.

8 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 Select a card from the list of detected cards in the YaST network card configuration module and click *Edit*.

2 In the *Address* tab, click *Hostname and Name Server*.

3 To disable DHCP-driven host name configuration, deselect *Change Hostname via DHCP*.

4 Enter *Hostname* and, if it is needed, *Domain Name*.

5 To disable DHCP driven updates of the name server list, deselect *Update Name Servers and Search List via DHCP*.

6 Enter the name servers and domain search list.

7 Click *OK*.

8 Click *Next*.

9 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 Select a card from the list of detected cards in the YaST network card configuration module and click *Edit*.

2 In the *Address* tab, click *Routing*.

3 Enter the IP of the *Default Gateway*.

4 Click *OK*.

5 Click *Next*.

6 To activate the configuration, click *Finish*.

Adding Special Hardware Options

Sometimes a module of a network card needs special parameters to work correctly. To set them with YaST, proceed as follows:

1 Select a card from the list of detected cards in the YaST network card configuration module and click *Edit*.

2 In the *Address* tab, click *Advanced* > *Hardware Details*.

3 In *Options*, enter the parameters for your network card. If two cards are configured that use the same module, these parameters are used for both.

4 Click *OK*.

5 Click *Next*.

6 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 *Edit*.

2 In the *General* tab, select the desired entry from *Device Activation*.

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 43.4.1, "Configuring the Firewall with YaST" (page 829), 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 *Edit*.

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:

No Zone, All Traffic Blocked
: All traffic will be blocked for this interface.

Internal Zone (Unprotected)
: The firewall is run, but does not enforce any rules to protect this interface. Only use this option, if your machine 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 run 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

It may happen that your card is not 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 Click *Add*.

2 Set the *Device Type* of the interface from the available options, *Configuration Name*, and *Module 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.

   Hardware Configuration Name specifies the name of the `/etc/sysconfig/hardware/hwcfg-*` file containing the hardware settings of your network card. This contains the name of the kernel module as well as the options needed to initialize the hardware.

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 *Static Address Setup* then complete *IP Address* and *Subnet Mask*. Here, you can also select to configure the hostname,

name server, and routing details (see Section “Configuring Hostname and DNS” (page 568) and Section “Configuring Routing” (page 569)).

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 29.1, “Wireless LAN” (page 533).

5 In the *General* tab, set the *Firewall Zone* and *Device Activation*. With *User Controlled*, grant connection control to ordinary users.

6 Click *Next*.

7 To activate the new network configuration, click *Finish*.

Information about the conventions for configuration names is available in the `getcfg(8)` man page.

30.4.2 Modem

TIP: IBM System z: Modem

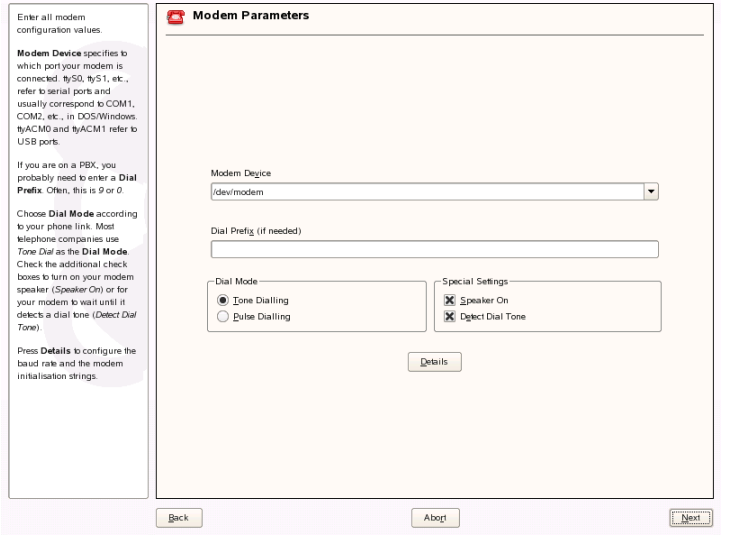
The configuration of this type of hardware is not supported on IBM System z platforms.

In the YaST Control Center, access the modem configuration with *Network Devices* > *Modem*. If your modem was not automatically detected, open the dialog for manual configuration by clicking *Add*. In the dialog that opens, enter the interface to which the modem is connected for *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 30.4 *Modem Configuration*



If 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 *User Controlled*. 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 (Internet service provider). 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.

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.

Stupid Mode
: This option is enabled by default. With it, input prompts sent by the ISP's server are ignored to prevent them from interfering with the connection process.

External Firewall Interface
: Selecting this option activates the SUSEfirewall2 and sets the interface as external. This way, the system 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*.

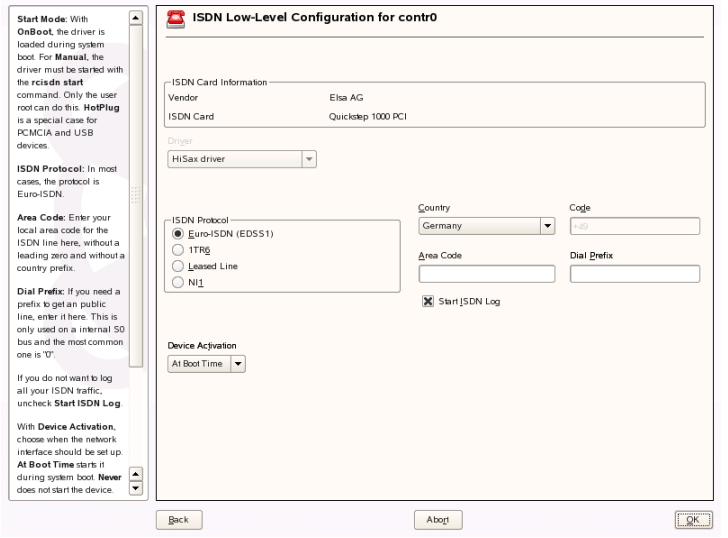
30.4.3 ISDN

TIP: IBM System z: ISDN

The configuration of this type of hardware is not supported on IBM System z platforms.

Use this module to configure one or several ISDN cards for your system. If YaST did not detect your ISDN card, click *Add* and manually select it. 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 30.5 *ISDN Configuration*



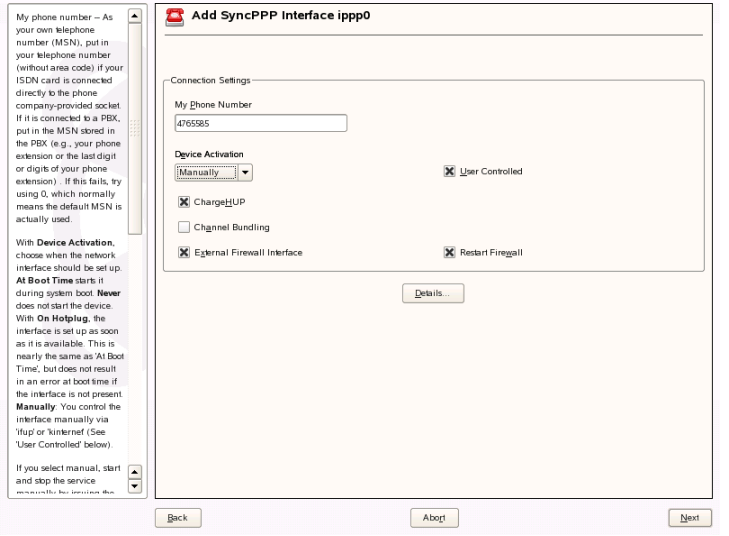
In the next dialog, shown in Figure 30.5, “ISDN Configuration” (page 575), select the protocol to use. The default is *Euro-ISDN (EDSS1)*, but for older or larger exchanges, select *1TR6*. If you are in the US, select *NI1*. Select your country in the relevant field. The corresponding country code then appears in the field next to it. Finally, provide your *Area Code* and the *Dial Prefix* if necessary.

Device Activation 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 30.6 *ISDN Interface Configuration*



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 that came 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 *User Controlled*.

Details opens a dialog in which to configure callback mode, remote connections to this interface and additional `ippd` options. 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 make all these settings active, select *Finish*.

30.4.4 Cable Modem

TIP: IBM System z: Cable Modem

The configuration of this type of hardware is not supported on IBM System z platforms.

In some countries, such as Austria and the US, 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 *Automatic Address Setup (via DHCP)* or *Static Address Setup*. 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`.

30.4.5 DSL

TIP: IBM System z: DSL

The configuration of this type of hardware is not supported on IBM System z platforms.

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:

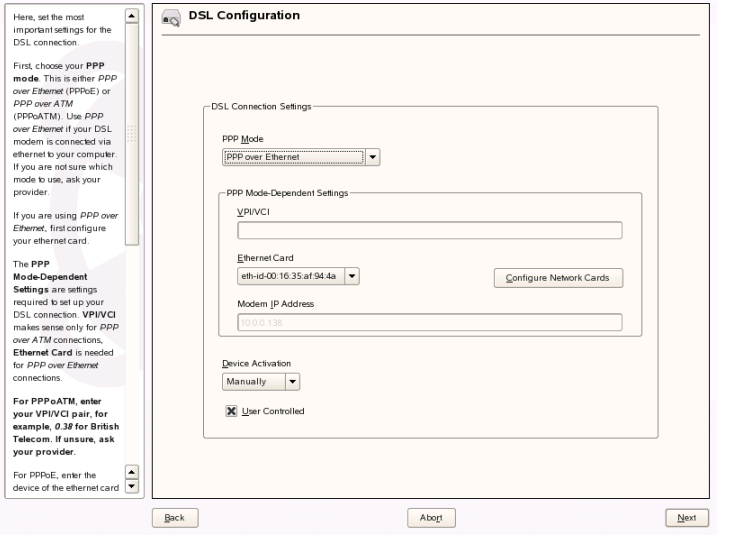
- PPP over Ethernet (PPPoE)
- PPP over ATM (PPPoATM)
- CAPI for ADSL (Fritz Cards)
- Point-to-Point Tunneling Protocol (PPTP)—Austria

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 30.4.1, "Configuring the Network Card with YaST" (page 565)). 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 *Automatic address setup (via DHCP)*. 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.

Figure 30.7 *DSL Configuration*



To begin the DSL configuration (see Figure 30.7, “DSL Configuration” (page 580)), first select the PPP mode and the ethernet card to which the DSL modem is connected (in most cases, this is `eth0`). Then use *Device Activation* to specify whether the DSL link should be established during the boot process. Click *User Controlled* to authorize the normal user without root permissions to activate or deactivate the interface with KInternet. The dialog also lets you select your country 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.

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.

30.4.6 IBM System z: Configuring Network Devices

SUSE Linux Enterprise for IBM System z supports several different types of network interfaces. YaST can be used to configure all of them.

The qeth-hsi Device

To add a `qeth-hsi` (Hipersockets) interface to the installed system, start the YaST network card module (*Network Devices* > *Network Card*). Select one of the devices marked *IBM Hipersocket* to use as the READ device address and click *Configure*. In the *Network address setup* dialog, specify the IP address and netmask for the new interface and leave the network configuration by pressing *Next* and *Finish*.

The qeth-ethernet Device

To add a `qeth-ethernet` (IBM OSA Express Ethernet Card) interface to the installed system, start the YaST network card module (*Network Devices* > *Network Card*). Select one of the devices marked *IBM OSA Express Ethernet Card* to use as the READ device address and click *Configure*. Enter the needed port name, some additional options (see the *Linux for IBM System z: Device Drivers, Features, and Commands* manual for reference at `http://www.ibm.com/developerworks/linux/linux390/documentation_novell_suse.html`), your IP address, and an appropriate netmask. Leave the network configuration with *Next* and *Finish*.

The ctc Device

To add a `ctc` (IBM parallel CTC Adapter) interface to the installed system, start the YaST network card module (*Network Devices* > *Network Card*). Select one of the devices marked *IBM parallel CTC Adapter* to use as your read channel and click *Configure*.

Choose the *Device Settings* that fit your devices (usually this would be *Compatibility mode*). Specify both your IP address and the IP address of the remote partner. If needed, adjust the MTU size with *Advanced* > *Detailed Settings*. Leave the network configuration with *Next* and *Finish*.

WARNING

The use of this interface is deprecated. This interface will not be supported in future versions of SUSE Linux Enterprise.

The lcs Device

To add an `lcs` (IBM OSA-2 Adapter) interface to the installed system, start the YaST network card module (*Network Devices* > *Network Card*). Select one of the devices marked *IBM OSA-2 Adapter* and click *Configure*. Enter the needed port number, some additional options (see the *Linux for IBM System z: Device Drivers, Features, and Commands* manual for reference at `http://www.ibm.com/developerworks/linux/linux390/documentation_novell_suse.html`), your IP address, and an appropriate netmask. Leave the network configuration with *Next* and *Finish*.

The IUCV Device

To add an `iucv` (IUCV) interface to the installed system, start the YaST network card module (*Network Devices* > *Network Card*). Select a device marked *IUCV* and click *Configure*. YaST prompts you for the name of your IUCV partner. Enter the name (this entry is case-sensitive) and select *Next*. Specify both your IP address and the IP address of your partner. If needed, adjust the MTU size with *Advanced* > *Detailed Settings*. Leave the network configuration with *Next* and *Finish*.

WARNING

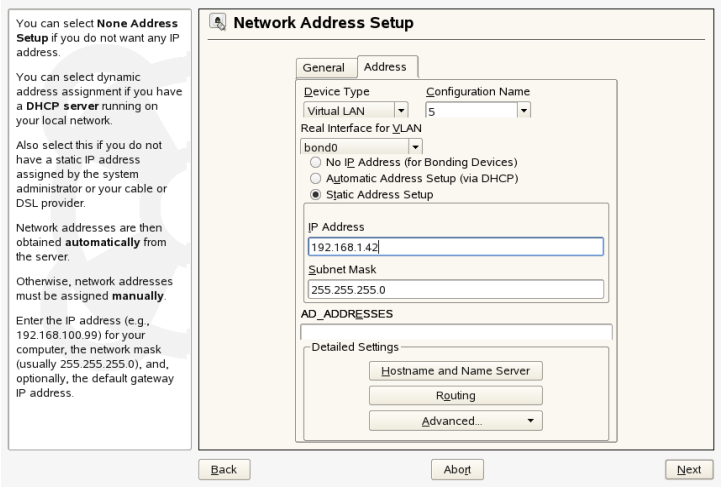
The use of this interface is deprecated. This interface will not be supported in future versions of SUSE Linux Enterprise.

30.5 Configuring VLAN Interfaces on SUSE Linux

VLAN is an abbreviation of *Virtual Local Area Network*. It allows the running of multiple *logical* (virtual) ethernets over one single physical ethernet. It logically splits the network into different broadcast domains so that packets are only switched between ports that are designated for the same VLAN. If you intend to use VLAN in your network setup, make sure that the package `vlan` is installed.

If the network connection of Linux is not dedicated to a specific logical LAN, you can set up access to one or more of these logical LANs. The VLAN interface configuration is supported via the normal ifup and ifdown scripts used for all other network interfaces, as well. The setup of VLAN devices is supported by YaST.

Figure 30.8 *The YaST VLAN Configuration*



Run the YaST module *Network Devices* > *Network Card*, select *Traditional Method with ifup* and press *Next*. Follow this procedure to actually setup the VLAN device:

Procedure 30.1 *Setting up VLAN Interfaces with YaST*

1 Press *Add* to create a new network interface.

2 In *Network Configuration*, select *Device Type Virtual Lan*.

3 Change the value of *Configuration Name* to the ID of your VLAN. Note that VLAN ID `1` is commonly used for managing purposes.

4 Press *Next*.

5 Select the interface that the VLAN device should connect to below *Real Interface for VLAN*.

6 Select the desired method for assigning an IP address to the VLAN device.

7 Press *Next* to finish the configuration.

For more information about VLAN, see `http://www.candelatech.com/~greear/vlan.html` and the package documentation found at `/usr/share/doc/packages/vlan/`.

30.6 Managing Network Connections with NetworkManager

NetworkManager is the ideal solution for a mobile workstation. With NetworkManager, you do not need to worry about reconfiguring network interfaces and switching between networks when your location changes. NetworkManager can automatically connect to known WLAN networks. If you have two or more connection possibilities, it can connect to the faster one.

NetworkManager is not a suitable solution in the following cases:

- You want to use more than one provider for dial-up for one interface.
- Your computer is a router for your network.
- Your computer provides network services for other computers in your network, for example, it is a DHCP or DNS server.
- Your computer is a Xen server or your system is a virtual system inside Xen.

- You want to use SCPM for network configuration management. To use SCPM and NetworkManager at the same time, SCPM cannot control network resources. .
- You want to use more than one active network connection simultaneously.

To enable or disable NetworkManager during the installation, click *Enable NetworkManager* or *Disable NetworkManager* in *Network Mode* of *Network Configuration*. To enable or disable NetworkManager on an installed system, follow these steps:

1. Open YaST.
2. Choose *Network Devices* > *Network Card*.
3. On the first screen, set the *Network Setup Method* option to *User Controlled with NetworkManager* to use NetworkManager. To disable NetworkManager, set the *Network Setup Method* option to *Traditional Method with ifup*.

After choosing the method, set up your network card using automatic configuration via DHCP or a static IP address or configure your modem. Find a detailed description of the network configuration with YaST in Section 30.4, “Configuring a Network Connection with YaST” (page 565) and Section 29.1, “Wireless LAN” (page 533). Configure supported wireless cards directly in NetworkManager.

To configure NetworkManager, use NetworkManager applets. KDE and GNOME each have their own applets for NetworkManager. An appropriate applet should start automatically with the desktop environment. The applet is then shown as an icon in the system tray. The functions of the applets are similar, but their interfaces are a little different. They can also be used in other graphical environments with standard system tray support.

30.6.1 Differences between ifup and NetworkManager

If you use NetworkManager for network setup, you can easily switch, stop, or start your network connection at any time from within your desktop environment using an applet. NetworkManager also makes it possible to change and configure wireless card connections without requiring `root` privileges. For this reason, NetworkManager is the ideal solution for a mobile workstation.

Traditional configuration with ifup also provides some ways to switch, stop, or start the connection with or without user intervention, like user-managed devices, but it always requires `root` privileges to change or configure a network device. This is often a problem for mobile computing, where is not possible to preconfigure all connection possibilities.

Both traditional configuration and NetworkManager can handle network connections with a wireless network (with WEP, WPA-PSK, and WPA-Enterprise access), dial-up, and wired networks both using DHCP and static configuration. They also support connection through VPN.

NetworkManager tries to keep your computer connected at all times using the best connection available. If available, it uses the fastest wired connection. If the network cable is accidentally disconnected, it tries to reconnect. It can find a network with the best signal strength from the list of your wireless connections and automatically use it to connect. To get the same functionality with ifup, a great deal of configuration effort is required.

30.6.2 For More Information

Find more information about NetworkManager on the following Web sites and directories:

- `http://www.gnome.org/projects/NetworkManager/`—NetworkManager project page
- `http://en.opensuse.org/Projects/KNetworkManager`—NetworkManager KNetworkManager project page

30.7 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.

All built-in network cards and hotplug network cards (PCMCIA, USB, some PCI cards) are detected and configured via hotplug. The system sees a network card in two different ways: first as a physical device and second as an interface. The insertion or detection of a device triggers a hotplug event. This hotplug event triggers the initialization of the device with the script `hwup`. When the network card is initialized as a new network interface, the kernel generates another hotplug event that triggers the setup of the interface with `ifup`.

The kernel numbers interface names according to the temporal order of their registration. The initialization sequence is decisive for the assignment of names. If one of several network card fails, the numbering of all subsequently initialized cards is shifted. For real hotpluggable cards, the order in which the devices are connected is what matters.

To achieve a flexible configuration, the configuration of the device (hardware) and the interface has been separated and the mapping of configurations to devices and interfaces is no longer managed on the basis of the interface names. The device configurations are located in `/etc/sysconfig/hardware/hwcfg-*`. The interface configurations are located in `/etc/sysconfig/network/ifcfg-*`. The names of the configurations are assigned in such a way that they describe the devices and interfaces with which they are associated. Because the former mapping of drivers to interface name required static interface names, this mapping can no longer take place in `/etc/modprobe.conf`. In the new concept, alias entries in this file would cause undesirable side effects.

The configuration names—everything after `hwcfg-` or `ifcfg-`—can describe the devices by means of the slot, a device-specific ID, or the interface name. For example, the configuration name for a PCI card could be `bus-pci-0000:02:01.0` (PCI slot) or `vpid-0x8086-0x1014-0x0549` (vendor and product ID). The name of the associated interface could be `bus-pci-0000:02:01.0` or `wlan-id-00:05:4e:42:31:7a` (MAC address).

To assign a certain network configuration to any card of a certain type (of which only one is inserted at a time) instead of a certain card, select less specific configuration names. For example, `bus-pcmcia` would be used for all PCMCIA cards. On the other hand, the names can be limited by a preceding interface type. For example, `wlan-bus-usb` would be assigned to WLAN cards connected to a USB port.

The system always uses the configuration that best describes an interface or the device providing the interface. The search for the most suitable configuration is handled by

getcfg. The output of getcfg delivers all information that can be used for describing a device. Details regarding the specification of configuration names are available in the manual page of getcfg.

With the described method, a network interface is configured with the correct configuration even if the network devices are not always initialized in the same order. However, the name of the interface still depends on the initialization sequence. There are two ways to ensure reliable access to the interface of a certain network card:

- getcfg-interface *configuration name* returns the name of the associated network interface. Therefore, the configuration name, such as firewall, dhcpd, routing, or various virtual network interfaces (tunnels), can be entered in some configuration files instead of the interface name, which is not persistent.

- Persistent interface names are assigned to each interface automatically. You may adjust them to suit your needs. When creating interface names, proceed as outlined in /etc/udev/rules.d/30-net_persistent_names.rules. However, the persistent name *pname* should not be the same as the name that would automatically be assigned by the kernel. Therefore, eth*, tr*, wlan*, qeth*, iucv*, and so on are not permitted. Instead, use net* or descriptive names like external, internal, or dmz. Make sure that the same interface name is not used twice. Allowed characters in interface names are restricted to [a-zA-Z0-9]. A persistent name can only be assigned to an interface immediately after its registration, which means that the driver of the network card must be reloaded or hwup *device description* must be executed. The command rcnetwork restart is not sufficient for this purpose.

IMPORTANT: Using Persistent Interface Names

The use of persistent interface names has not been tested in all areas. Therefore, some applications may not be able to handle freely selected interface names.

---

ifup requires an existing interface, because it does not initialize the hardware. The initialization of the hardware is handled by the command hwup (executed by hotplug or coldplug). When a device is initialized, ifup is automatically executed for the new interface via hotplug and the interface is set up if the start mode is onboot, hotplug, or auto and the network service was started. Formerly, the command

`ifup` *`interfacename`* triggered the hardware initialization. Now the procedure has been reversed. First, a hardware component is initialized then all other actions follow. In this way, a varying number of devices can always be configured in the best way possible with an existing set of configurations.

Table 30.5, "Manual Network Configuration Scripts" (page 589) summarizes the most important scripts involved in the network configuration. Where possible, the scripts are distinguished by hardware and interface.

Table 30.5 *Manual Network Configuration Scripts*

Configuration Stage	Command	Function
Hardware	`hw{up,down,status}`	The `hw*` scripts are executed by the hotplug subsystem to initialize a device, undo the initialization, or query the status of a device. More information is available in the manual page of `hwup`.
Interface	`getcfg`	`getcfg` can be used to query the interface name associated with a configuration name or a hardware description. More information is available in the manual page of `getcfg`.
Interface	`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`.

More information about hotplug and persistent device names is available in Chapter 24, *Dynamic Kernel Device Management with udev* (page 463).

30.7.1 Configuration Files

This section provides an overview of the network configuration files and explains their purpose and the format used.

/etc/syconfig/hardware/hwcfg-*

These files contain the hardware configurations of network cards and other devices. They contain the needed parameters, such as the kernel module, start mode, and script associations. Refer to the manual page of `hwup` for details. Regardless of the existing hardware, the `hwcfg-static-*` configurations are applied when coldplug is started.

/etc/sysconfig/network/ifcfg-*

These files contain the configurations for network interface. 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.

► **zseries:** IBM System z do not support USB. The names of the interface files and network aliases contain System z-specific elements like `qeth`. ◄

/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 30.5, "`/etc/resolv.conf`" (page 591) shows what `/etc/resolv.conf` could look like.

Example 30.5 */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`), `pcmcia`, and `hotplug`, 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`, `pcmcia`, and `hotplug`.

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 30.6, "`/etc/hosts`" (page 592), 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 30.6 */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 30.7, "`/etc/networks`" (page 593).

Example 30.7 */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 30.6, "Parameters for /etc/host.conf" (page 593) shows the parameters available. A sample `/etc/host.conf` is shown in Example 30.8, "`/etc/host.conf`" (page 594).

Table 30.6 *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 30.8 */etc/host.conf*

```
# We have named running
order hosts bind
# Allow multiple addrs
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 30.9, "`/etc/nsswitch.conf`" (page 594). 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 (see Chapter 33, *The Domain Name System* (page 615)).

Example 30.9 */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 30.7, “Databases Available via /etc/nsswitch.conf” (page 595). In addition, `automount`, `bootparams`, `netmasks`, and `publickey` are expected in the near future. The configuration options for NSS databases are listed in Table 30.8, “Configuration Options for NSS “Databases”” (page 596).

Table 30.7 *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 30.8 *Configuration Options for NSS "Databases"*

`files`	directly access files, for example, `/etc/aliases`
`db`	access via a database
`nis`, `nisplus`	NIS, see also Chapter 35, *Using NIS* (page 659)
`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.

30.7.2 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 a 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 get 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 30.10, "Output of the Command ping" (page 599), 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 example.com` or `ping 130.57.5.75`. 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 3 192.168.0`.

Example 30.10 *Output of the Command ping*

```
ping -c 3 example.com
PING example.com (130.57.5.75) 56(84) bytes of data.
64 bytes from example.com (130.57.5.75): icmp_seq=1 ttl=49 time=188 ms
64 bytes from example.com (130.57.5.75): icmp_seq=2 ttl=49 time=184 ms
64 bytes from example.com (130.57.5.75): icmp_seq=3 ttl=49 time=183 ms
--- 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 10 192.168.0`.

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 192.168.0`.

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 30.11, "Output of the ifconfig Command" (page 600), 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 30.11 *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 30.12 *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.

30.7.3 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 30.9, “Some Start-Up Scripts for Network Programs” (page 601).

Table 30.9 *Some Start-Up Scripts for Network Programs*

`/etc/init.d/network`	This script handles the configuration of the network interfaces. The hardware must already have been initialized by `/etc/init.d/coldplug` (via `hotplug`). If the `network` service was not started, no network interfaces are implemented when they are inserted via hotplug.
`/etc/init.d/inetd`	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.

30.8 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.

30.8.1 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 plain-text 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.

30.8.2 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.

31 SLP Services in the Network

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.

SUSE Linux Enterprise® 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 SUSE Linux Enterprise

Services that offer SLP support include cupsd, rsyncd, ypserv, openldap2, openwbem (CIM), ksysguardd, saned, kdm vnc login, smpppd, rpasswd, postfix, and sshd (via fish).

31.1 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 SUSE Linux Enterprise, the slpd daemon is controlled by means of a separate initialization 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.

31.2 SLP Front-Ends in SUSE Linux Enterprise

To find services provided by SLP in your network, use an SLP front-end. SUSE Linux Enterprise 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.

YaST SLP Browser
: YaST contains a separate SLP browser that lists all services in the local network announced by SLP in a tree diagram. Find it as *Network Services > SLP Browser*.

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.

31.3 Installation over SLP

If you offer an installation server with SUSE Linux Enterprise installation media within your network, this can be registered with SLP. For details, see Section 4.2.1, “Setting Up an Installation Server Using YaST” (page 56). 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.

31.4 Providing Services with SLP

Many applications in SUSE Linux Enterprise 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 with 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-tcp-port` and `description`. `watch-tcp-port` 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.

TIP: YaST and SLP

Some services brokered by YaST, such as an installation server or YOU server, perform this registration automatically when you activate SLP in the module dialogs. YaST then creates registration files for these services.

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.

31.5 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 SUSE Linux Enterprise 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*.

Time Synchronization with NTP

32

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.

32.1 Configuring an NTP Client with YaST

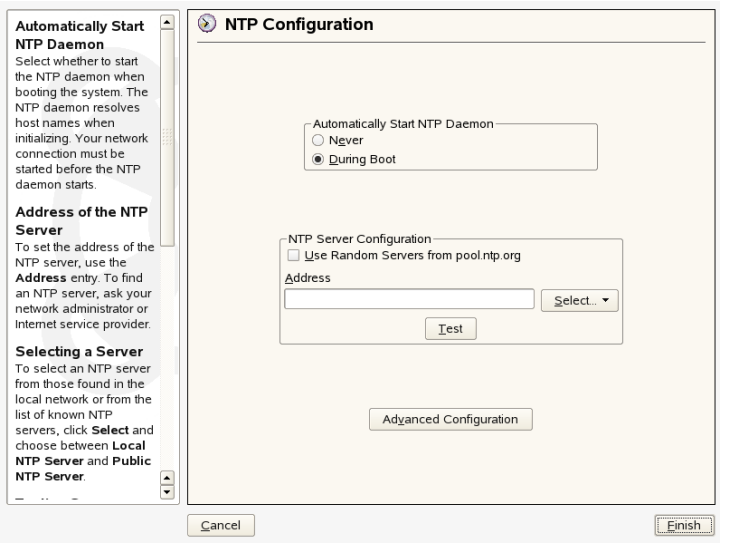
xntp is preset to use the local computer clock as a time reference. Using the (BIOS) clock, however, only serves as a fallback for the case that no time source of greater precision is available. YaST facilitates the configuration of an NTP client. For a system that is not running a firewall, use either the quick or advanced configuration. For a

firewall-protected system, the advanced configuration can open the required ports in SuSEfirewall2.

32.1.1 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 32.1 *YaST: Configuring an NTP Client*



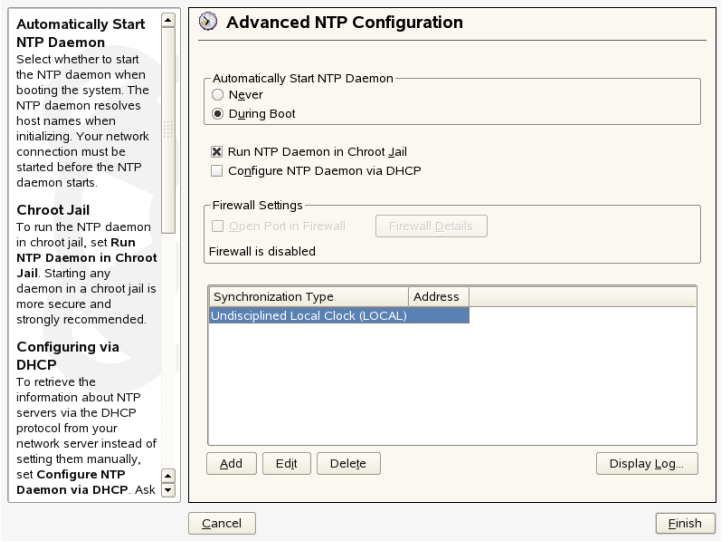
In the detailed server selection dialog, determine whether to implement time synchronization using a time server from your local network (*Local NTP Server*) or an Internet-based time server that takes care of your time zone (*Public NTP Server*). For a local time server, click *Lookup* to start an SLP 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*.

32.1.2 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 32.1, “YaST: Configuring an NTP Client” (page 610), after selecting the start-up mode as described in the quick configuration.

Figure 32.2 *YaST: Complex NTP Configuration*



In *Advanced NTP Configuration*, 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. *Configure NTP Daemon via DHCP* sets up the NTP client to get a list of the NTP servers available in your network via DHCP.

Enable *Open Port in Firewall* if SuSEfirewall 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.

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 32.1.1, "Quick NTP Client Configuration" (page 610)). 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. 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.

32.2 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`.

32.3 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.

33 The Domain Name System

DNS (domain name system) is needed to resolve the domain names and hostnames into IP addresses. In this way, the IP address 192.168.0.1 is assigned to the hostname `earth`, for example. Before setting up your own name server, read the general information about DNS in Section 30.3, "Name Resolution" (page 563). The following configuration examples refer to BIND.

33.1 DNS Terminology

Zone
: The domain name space is divided into regions called zones. For instance, if you have `example.org`, you have the `example` section, or zone, of the `org` 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.

33.2 Configuration with YaST

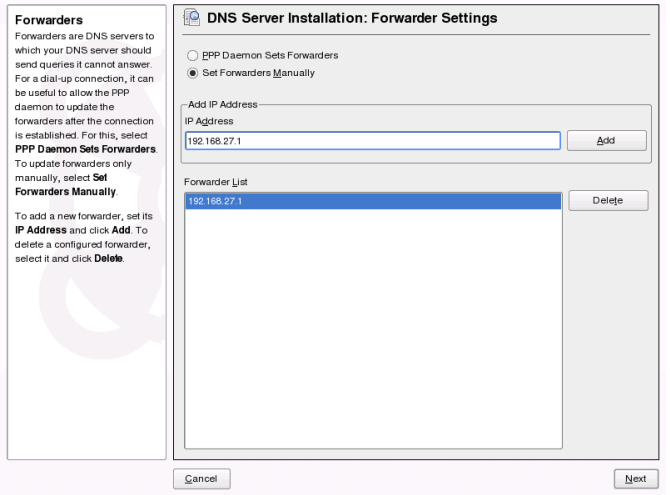
You can use the DNS module of YaST to configure a DNS server for your local network. To configure a Samba server, start YaST and select *Network Services > DNS Server*. When starting the module for the first time, a wizard starts, prompting you to make just a few basic decisions concerning administration 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, such as setting up ACLs, logging, TSIG keys, and other options.

33.2.1 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.

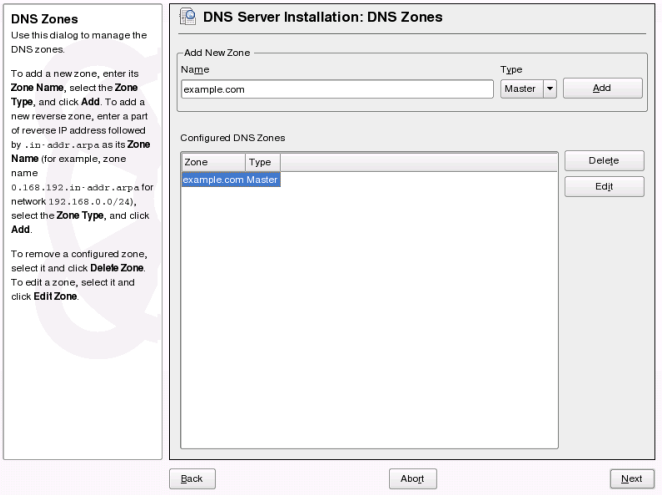
1 When starting the module for the first time, the *Forwarder Settings* dialog, shown in Figure 33.1, “DNS Server Installation: Forwarder Settings” (page 617), opens. In it, decide whether the PPP daemon should provide a list of forwarders on dial-up via DSL or ISDN (*PPP Daemon Sets Forwarders*) or whether you want to supply your own list (*Set Forwarders Manually*).

Figure 33.1 *DNS Server Installation: Forwarder Settings*



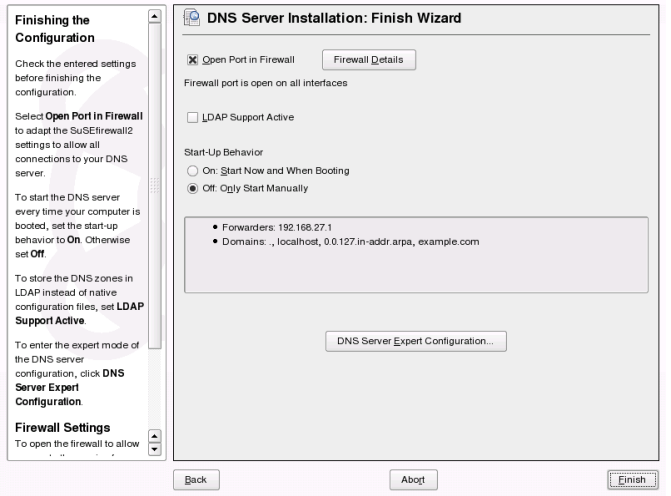
2 The *DNS Zones* dialog consists of several parts and is responsible for the management of zone files, described in Section 33.5, “Zone Files” (page 633). For a new zone, provide a name for it in *Zone Name*. To add a reverse zone, the name must end in `.in-addr.arpa`. Finally, select the *Zone Type* (master or slave). See Figure 33.2, “DNS Server Installation: DNS Zones” (page 618). Click *Edit Zone* to configure other settings of an existing zone. To remove a zone, click *Delete Zone*.

Figure 33.2 *DNS Server Installation: DNS Zones*



3 In the final dialog, you can open the DNS port in the firewall by clicking *Open Port in Firewall*. Then decide whether or not the DNS server should be started (*On* or *Off*). You can also activate LDAP support. See Figure 33.3, “DNS Server Installation: Finish Wizard” (page 619).

Figure 33.3 *DNS Server Installation: Finish Wizard*



33.2.2 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*.

By selecting *LDAP Support Active*, the zone files are managed by an LDAP database. Any changes to zone data written to the LDAP database are picked up by the DNS server as soon as it is restarted or prompted to reload its configuration.

DNS Server: Basic Options

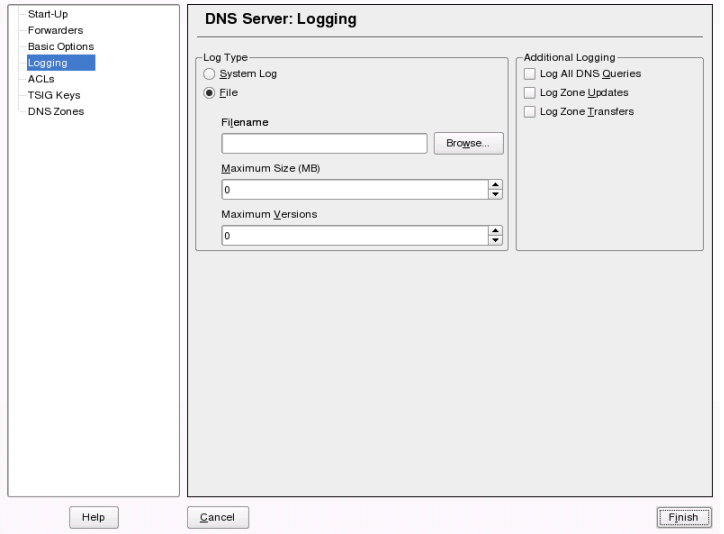
In this section, set basic server options. From the *Option* menu, select the desired item then specify the value in the corresponding entry field. Include the new entry by selecting *Add*.

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 33.4, “DNS Server: Logging” (page 620).

Figure 33.4 *DNS Server: Logging*



Using ACLs

Use this window to define ACLs (access control lists) to enforce access restrictions. After providing a distinct name under *Name*, specify an IP address (with or without netmask) under *Value* in the following fashion:

```
{ 10.10/16; }
```

The syntax of the configuration file requires that the address ends with a semicolon and is put into curly braces.

TSIG Keys

The main purpose of TSIGs (transaction signatures) is to secure communications between DHCP and DNS servers. They are described in Section 33.7, “Secure Transactions” (page 637).

To generate a TSIG key, enter a distinctive name in the field labeled *Key ID* and specify the file where the key should be stored (*Filename*). Confirm your choices with *Add*.

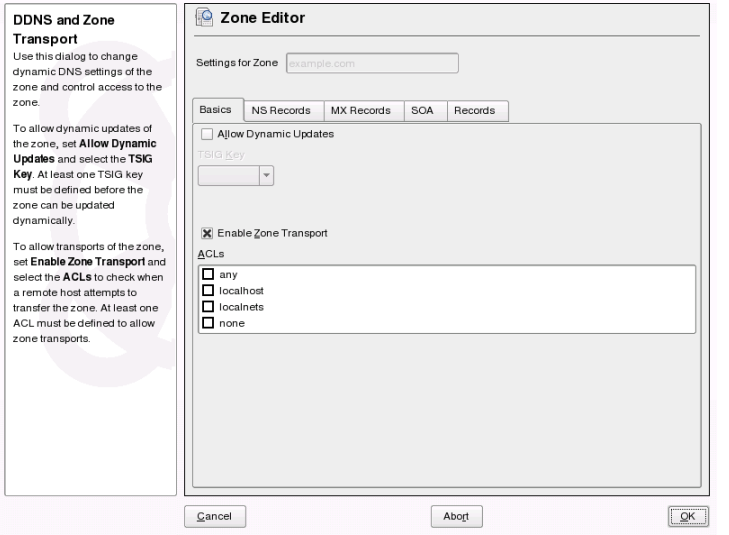
To use a previously created key, leave the *Key ID* field blank and select the file where it is stored under *File Name*. After that, confirm with *Add*.

Adding a Slave Zone

To add a slave zone, select *DNS Zones*, choose the zone type *Slave*, 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 33.5, “DNS Server: Slave Zone Editor” (page 622).

Figure 33.5 *DNS Server: Slave Zone Editor*



Adding a Master Zone

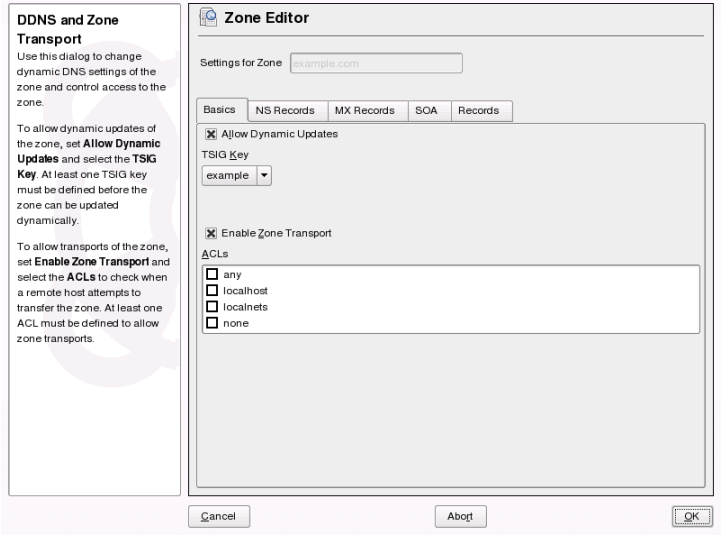
To add a master zone, select *DNS Zones*, choose the zone type *Master*, write the name of the new zone, and click *Add*.

Editing a Master Zone

To edit a master zone, select *DNS Zones*, choose the zone type *Master*, select the master zone from the table, and click *Edit*. The dialog consists of several pages: *Basic* (the one opened first), *NS Records*, *MX Records*, *SOA*, and *Records*.

The basic dialog, shown in Figure 33.6, “DNS Server: Zone Editor (Basic)” (page 623), lets you define settings for dynamic DNS and access options for zone transfers to clients and slave name servers. To permit the dynamic update of zones, select *Allow Dynamic Updates* as well as the corresponding TSIG key. The key must have been defined before the update action starts. To enable zone transfers, select the corresponding ACLs. ACLs must have been defined already.

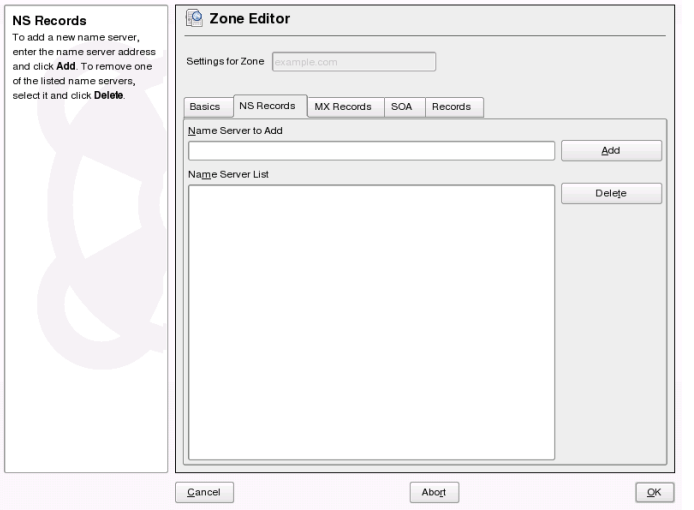
Figure 33.6 *DNS Server: Zone Editor (Basic)*



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 33.7, “DNS Server: Zone Editor (NS Records)” (page 624).

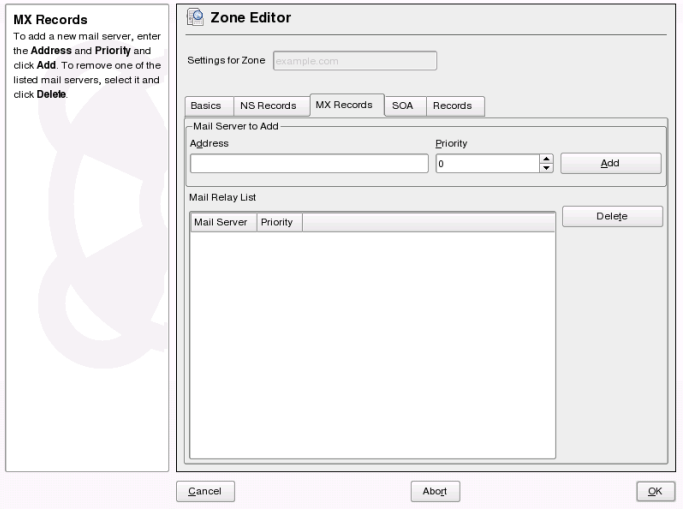
Figure 33.7 *DNS Server: Zone Editor (NS Records)*



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 33.8, “DNS Server: Zone Editor (MX Records)” (page 625).

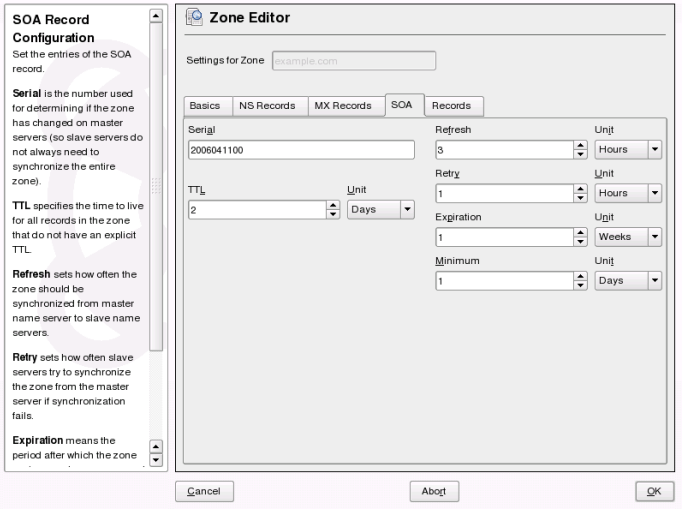
Figure 33.8 *DNS Server: Zone Editor (MX Records)*



Zone Editor (SOA)

This page allows you to create SOA (start of authority) records. For an explanation of the individual options, refer to Example 33.6, “File /var/lib/named/world.zone” (page 633). Changing SOA records is not supported for dynamic zones managed via LDAP.

Figure 33.9 *DNS Server: Zone Editor (SOA)*



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.

33.3 Starting the Name Server BIND

On a SUSE Linux Enterprise® 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 33.1, "Forwarding Options in named.conf" (page 628) are just examples. Adjust these entries to your own setup.

Example 33.1 *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; 192.168.0.99; };
        allow-query { 127/8; 192.168.0/24; };
        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`.

33.4 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 33.2, "A Basic /etc/named.conf" (page 629).

Example 33.2 *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";
};
```

33.4.1 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 `192.168.1/24`. The `/24` at the end is an abbreviated expression for the netmask, in this case, `255.255.255.0`.

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 de-

tected 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.

33.4.2 Logging

What, how, and where logging takes place can be extensively configured in BIND. Normally, the default settings should be sufficient. Example 33.3, “Entry to Disable Logging” (page 631) shows the simplest form of such an entry and completely suppresses any logging.

Example 33.3 *Entry to Disable Logging*

```
logging {
        category default { null; };
};
```

33.4.3 Zone Entries

Example 33.4 *Zone Entry for my-domain.de*

```
zone "my-domain.de" in {
      type master;
      file "my-domain.zone";
      notify no;
};
```

After `zone`, specify the name of the domain to administer (`my-domain.de`) followed by `in` and a block of relevant options enclosed in curly braces, as shown in Example 33.4, “Zone Entry for my-domain.de” (page 631). 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 33.5, “Zone Entry for other-domain.de” (page 632).

Example 33.5 *Zone Entry for other-domain.de*

```
zone "other-domain.de" in {
      type slave;
      file "slave/other-domain.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 `my-domain.zone` or file “slave/other-domain.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.

33.5 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 `world.zone`, responsible for the domain `world.cosmos`, shown in Example 33.6, "File /var/lib/named/world.zone" (page 633).

Example 33.6 *File /var/lib/named/world.zone*

```
$TTL 2D
world.cosmos. IN SOA      gateway  root.world.cosmos. (
            2003072441  ; serial
            1D          ; refresh
            2H          ; retry
            1W          ; expiry
            2D )        ; minimum

            IN NS       gateway
            IN MX       10 sun

gateway     IN A        192.168.0.1
            IN A        192.168.1.1
sun         IN A        192.168.0.2
moon        IN A        192.168.0.3
earth       IN A        192.168.1.2
mars        IN A        192.168.1.3
www         IN CNAME    moon
```

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 `world.cosmos` in the first position. This ends with a `.`, 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 `gateway` to `gateway.world.cosmos`, 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@world.cosmos` the entry must read `root.world.cosmos.`. 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. `gateway` is extended to `gateway.world.cosmos` 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 `world.cosmos`. In this example, this is the host `sun.world.cosmos`. 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 12–17:

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 `world.cosmos` is added to all of them. Two IP addresses are assigned to the host `gateway`, 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

A 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 18:

The alias `www` can be used to address `mond` (`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 `192.168.1` is resolved into `1.168.192.in-addr.arpa`. See Example 33.7, “Reverse Lookup” (page 636).

Example 33.7 *Reverse Lookup*

```
$TTL 2D
1.168.192.in-addr.arpa. IN SOA gateway.world.cosmos. root.world.cosmos. (
                        2003072441      ; serial
                        1D              ; refresh
                        2H              ; retry
                        1W              ; expiry
                        2D )            ; minimum

                        IN NS           gateway.world.cosmos.

1                       IN PTR          gateway.world.cosmos.
2                       IN PTR          earth.world.cosmos.
3                       IN PTR          mars.world.cosmos.
```

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 `192.168.1.0`. Given that the zone is called `1.168.192.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 `world.cosmos` example.

Lines 3–7:

See the previous example for `world.cosmos`.

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.

33.6 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 33.7, “Secure Transactions” (page 637).

33.7 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 `192.168.2.3` in this example), the server's `/etc/named.conf` must include the following rule:

```
server 192.168.2.3 {
  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`.

33.8 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`.

33.9 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 SUSE Linux Enterprise.

34 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.

TIP: IBM System z: DHCP Support

On IBM System z platforms, DHCP only works on interfaces using the OSA and OSA Express network cards. These cards are the only ones with a MAC, which is required for the DHCP autoconfiguration features.

One way to configure a DHCP server is to identify each client using the hardware address of its network card (which should be fixed in most cases), 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 reconfig-

uring 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.

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.

34.1 Configuring a DHCP Server with YaST

IMPORTANT: LDAP Support

In this version of SUSE Linux Enterprise, 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, set up 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.

34.1.1 Initial Configuration (Wizard)

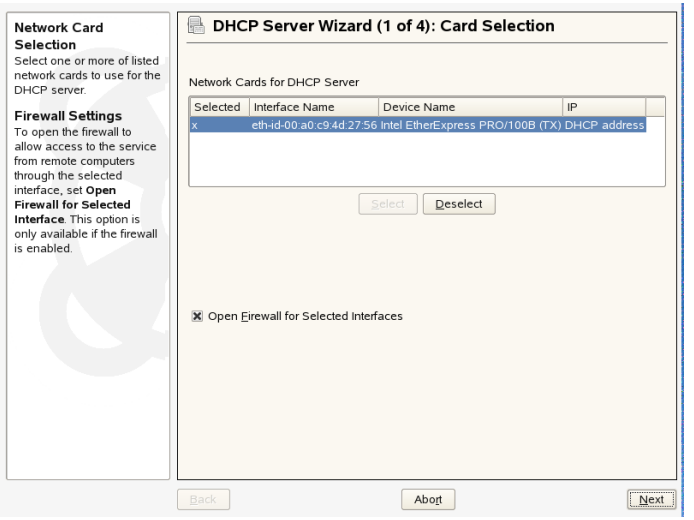
When the module is started for the first time, a wizard starts, prompting you to make a few basic decision concerning server administration. Completing this initial setup produces a very basic server configuration that should function in essential aspects. The expert mode can be used to deal with more 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 34.1, “DHCP Server: Card Selection” (page 643).

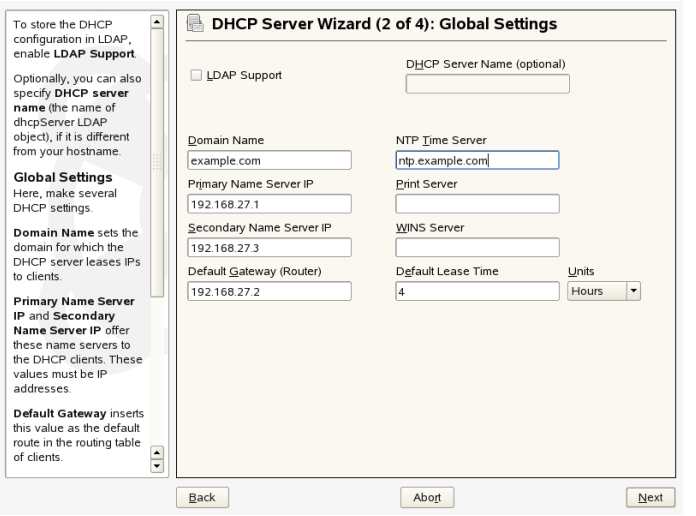
Figure 34.1 *DHCP Server: Card Selection*



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 34.2, “DHCP Server: Global Settings” (page 644).

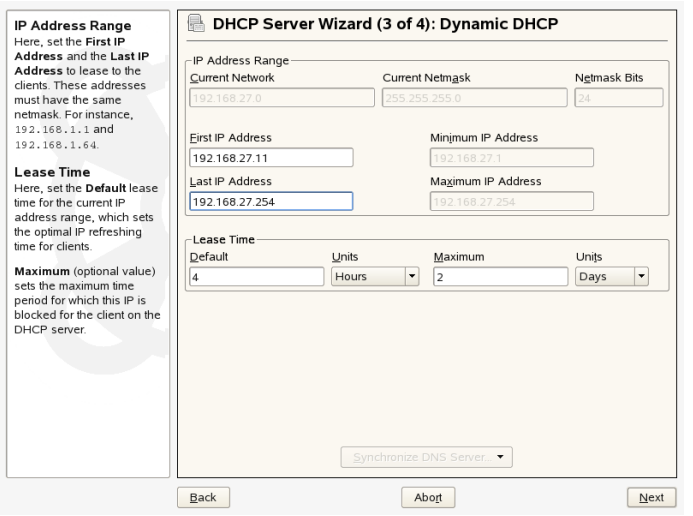
Figure 34.2 *DHCP Server: Global Settings*



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 34.3, “DHCP Server: Dynamic DHCP” (page 645).

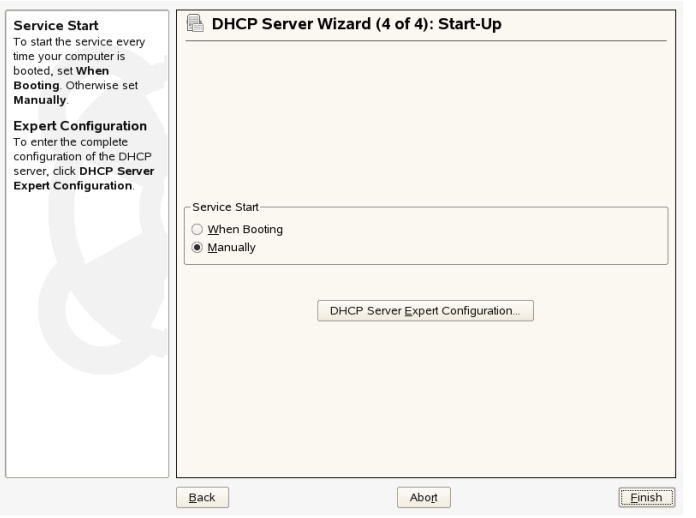
Figure 34.3 *DHCP Server: Dynamic DHCP*



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 34.4, “DHCP Server: Start-Up” (page 646). Alternatively, you can select *Host Management* from the tree structure to the left to configure special host management features in addition to the basic configuration (see Figure 34.5, “DHCP Server: Host Management” (page 647)).

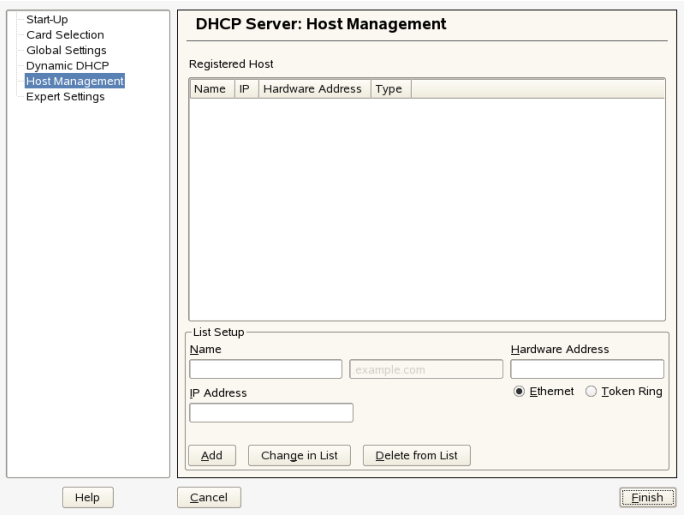
Figure 34.4 *DHCP Server: Start-Up*



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 34.5, “DHCP Server: Host Management” (page 647).

Figure 34.5 *DHCP Server: Host Management*



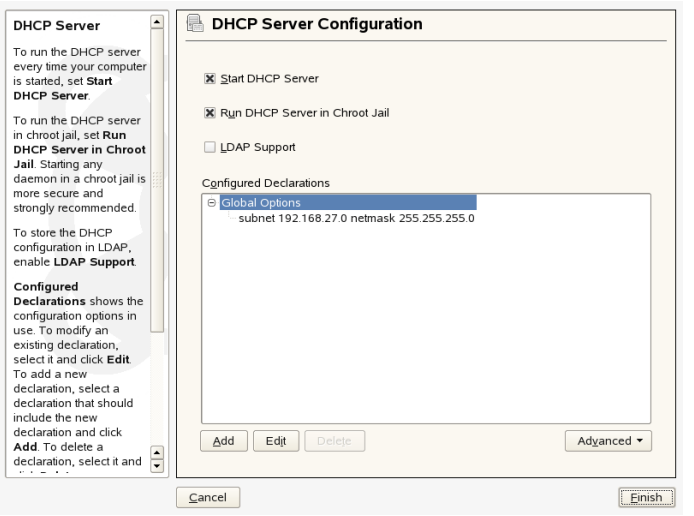
34.1.2 Expert Configuration

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 34.6, “DHCP Server: Chroot Jail and Declarations” (page 648). 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.

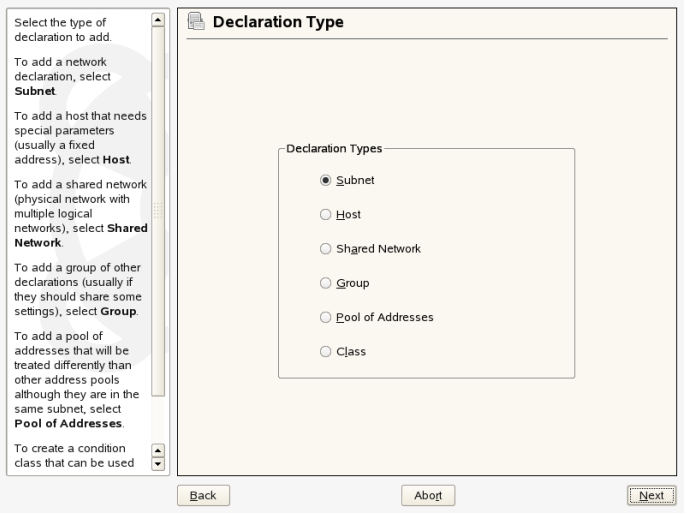
Figure 34.6 *DHCP Server: Chroot Jail and Declarations*



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 34.7, “DHCP Server: Selecting a Declaration Type” (page 649)).

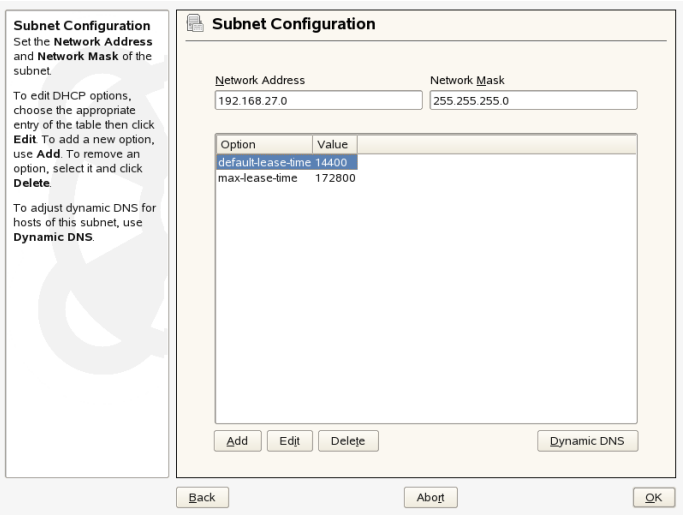
Figure 34.7 *DHCP Server: Selecting a Declaration Type*



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*.

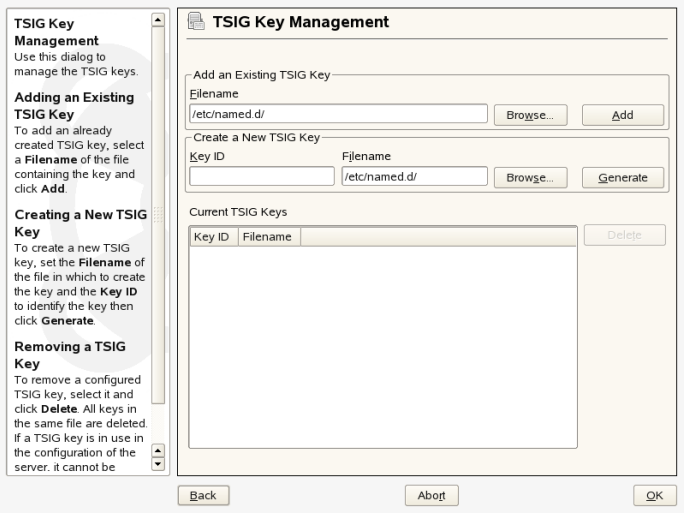
Figure 34.8 *DHCP Server: Configuring Subnets*



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 34.10, “DHCP Server: Interface Configuration for Dynamic DNS” (page 652)).

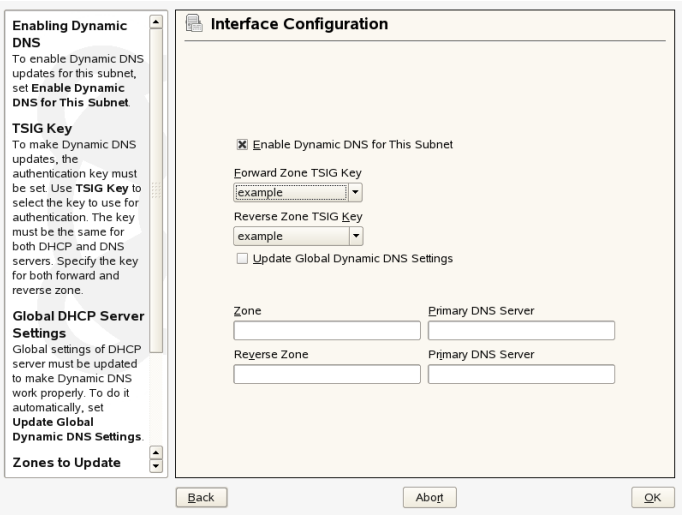
Figure 34.9 *DHCP Server: TSIG Configuration*



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 34.8, “DHCP Server: Configuring Subnets” (page 650)). Selecting *Ok* again returns to the original expert configuration dialog.

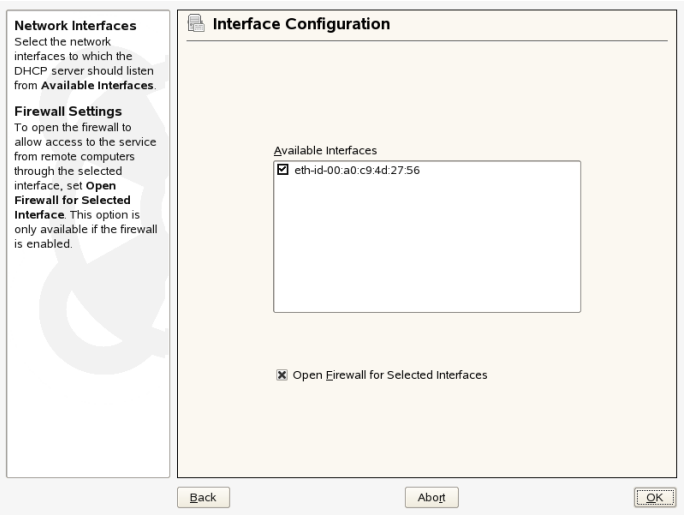
Figure 34.10 *DHCP Server: Interface Configuration for Dynamic DNS*



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 34.11, “DHCP Server: Network Interface and Firewall” (page 653)), after which you can return to the original dialog by selecting *Ok*.

Figure 34.11 *DHCP Server: Network Interface and Firewall*



After completing all configuration steps, close the dialog with *Ok*. The server is now started with its new configuration.

34.2 DHCP Software Packages

Both a DHCP server and DHCP clients are available for SUSE Linux Enterprise. The DHCP server available is dhcpd (published by the Internet Software 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.

SUSE Linux Enterprise 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`.

34.3 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 34.1, “The Configuration File /etc/dhcpd.conf” (page 654).

Example 34.1 *The Configuration File /etc/dhcpd.conf*

```
default-lease-time 600;         # 10 minutes
max-lease-time 7200;            # 2  hours

option domain-name "cosmos.all";
option domain-name-servers 192.168.1.1, 192.168.1.2;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option subnet-mask 255.255.255.0;

subnet 192.168.1.0 netmask 255.255.255.0
 {
  range 192.168.1.10 192.168.1.20;
  range 192.168.1.100 192.168.1.200;
 }
```

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 33, *The Domain Name System* (page 615).

- 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 34.1, "The Configuration File /etc/dhcpd.conf" (page 654), clients may be given any address between `192.168.1.10` and `192.168.1.20` as well as `192.168.1.100` and `192.168.1.200`.

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 SUSE Linux Enterprise 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.

34.3.1 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, `00:00:45:12:EE:F4`). If the respective lines, like the ones in Example 34.2, "Additions to the Configuration File" (page 656), are added to the configuration file of Example 34.1, "The Configuration File /etc/dhcpd.conf" (page 654), the DHCP daemon always assigns the same set of data to the corresponding client.

Example 34.2 *Additions to the Configuration File*

```
host earth {
hardware ethernet 00:00:45:12:EE:F4;
fixed-address 192.168.1.21;
}
```

The name of the respective client (`host` *`hostname`*, here `earth`) 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 00:00:45:12:EE:F4
```

In the preceding example, a client with a network card having the MAC address `00:00:45:12:EE:F4` is assigned the IP address `192.168.1.21` and the hostname earth 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.

34.3.2 The SUSE Linux Enterprise Version

To improve security, the SUSE Linux Enterprise 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`.

34.4 For More Information

More information about DHCP is available at the Web site of the *Internet Software Consortium* (`http://www.isc.org/products/DHCP/`). Information is also available in the `dhcpd`, `dhcpd.conf`, `dhcpd.leases`, and `dhcp-options` man pages.

35 Using NIS

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.

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."

35.1 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 35.1.1, "Configuring a NIS Master Server" (page 660).
- If your NIS master server should export its data to slave servers, set up the master server as described in Section 35.1.1, "Configuring a NIS Master Server" (page 660)

and set up slave servers in the subnets as described in Section 35.1.2, “Configuring a NIS Slave Server” (page 664).

35.1.1 Configuring a NIS Master Server

To configure a NIS master server for your network, proceed as follows:

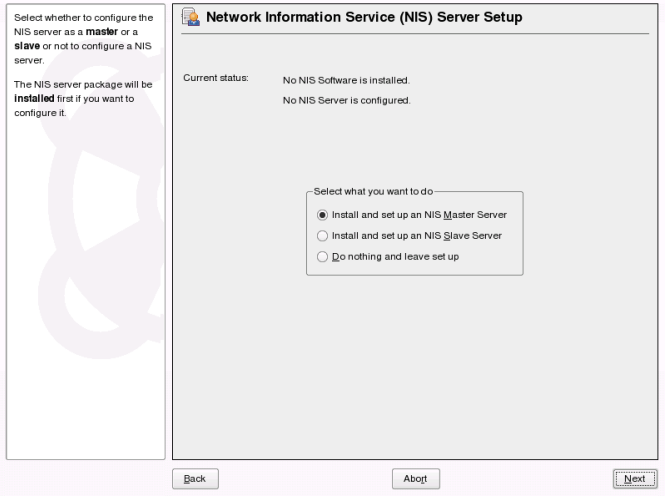
1. Start *YaST > Network Services > NIS Server*.

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 35.1 *NIS Server Setup*



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*.

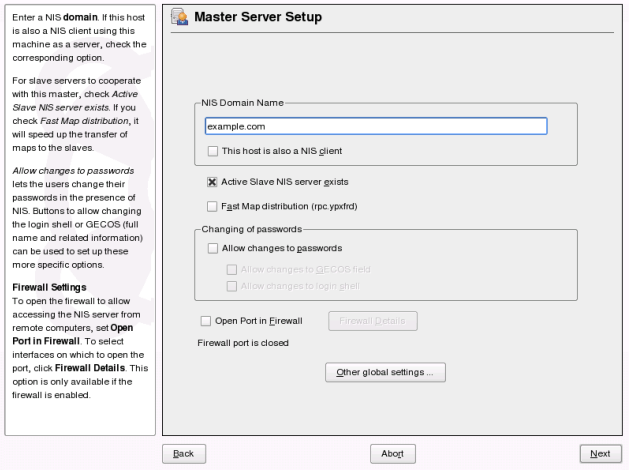
Select *Changing of passwords* to allow users in your network (both local users and those managed through the NIS server) to change their passwords on the NIS server (with the command `yppasswd`).

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*.

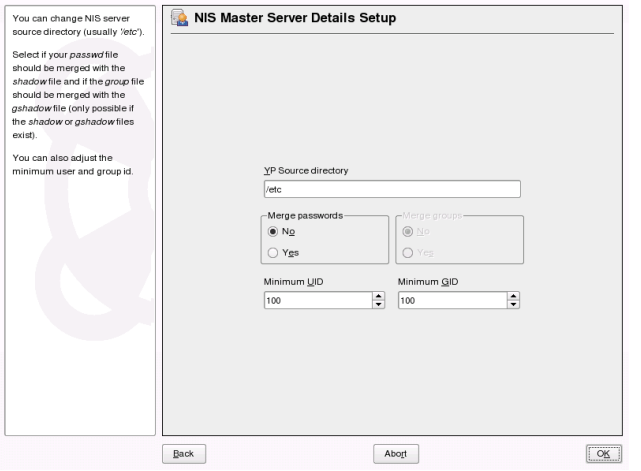
3d Select *Open Ports in Firewall* to have YaST adapt the firewall settings for the NIS server.

Figure 35.2 *Master Server Setup*



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 35.3 *Changing the Directory and Synchronizing Files for a NIS Server*

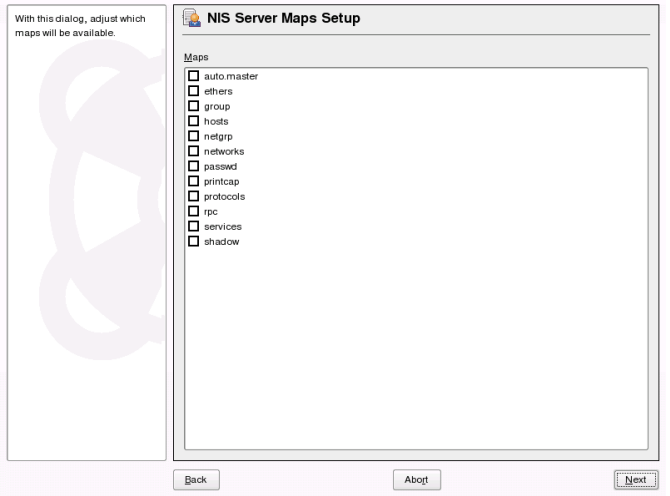


4 If you previously enabled *Active Slave NIS Server Exists*, enter the hostnames used as slaves and click *Next*.

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 35.4 *NIS Server Maps Setup*

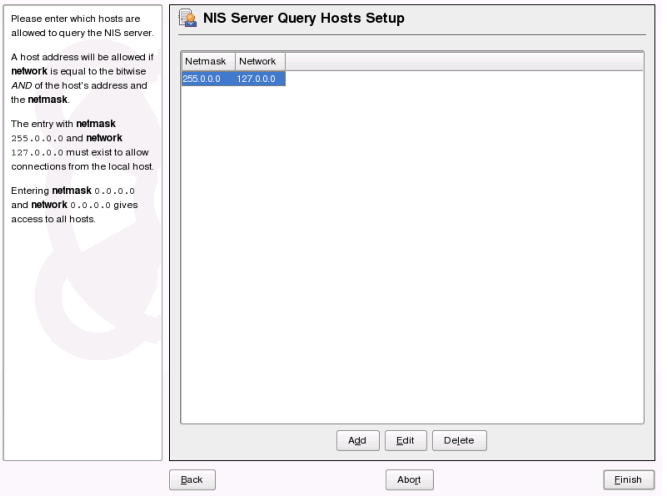


7 Enter the hosts that are allowed to query the NIS server. You can add, edit, or delete hosts by clicking the appropriate button. Specify from which networks requests can be sent to the NIS server. Normally, this is your internal network. In this case, there should be the following two entries:

```
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 35.5 *Setting Request Permissions for a NIS Server*



8 Click *Finish* to save changes and exit the setup.

35.1.2 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.

35.2 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 34, *DHCP* (page 641). If a static IP address is used, specify the NIS domain and the NIS server manually. See Figure 35.6, “Setting Domain and Address of a NIS Server” (page 666). *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.

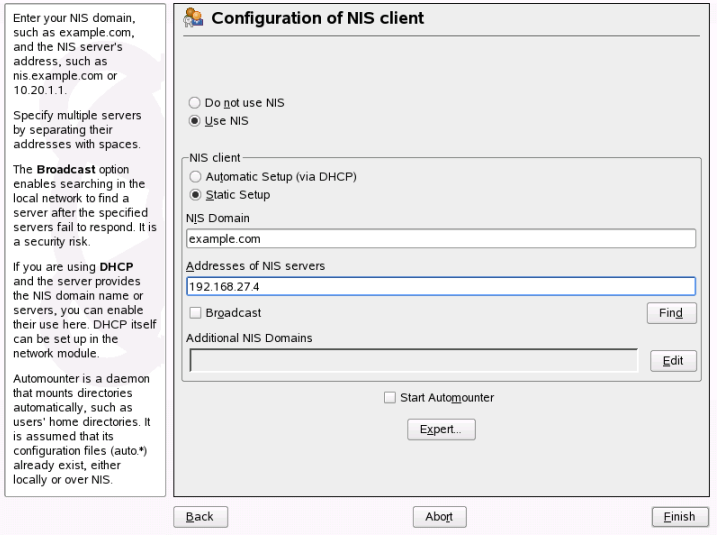
You can also specify multiple servers by entering their addresses in *Addresses of NIS servers* and separating them by spaces.

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 35.6 *Setting Domain and Address of a NIS Server*



36 LDAP—A Directory Service

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.

36.1 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.

36.2 Structure of an LDAP Directory Tree

To get a deep background knowledge on how an 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 SUSE Linux Enterprise.Read on at Section 36.5, “Configuring an LDAP Server with YaST” (page 683) or Section 36.3, “Server Configuration with slapd.conf” (page 673), 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:

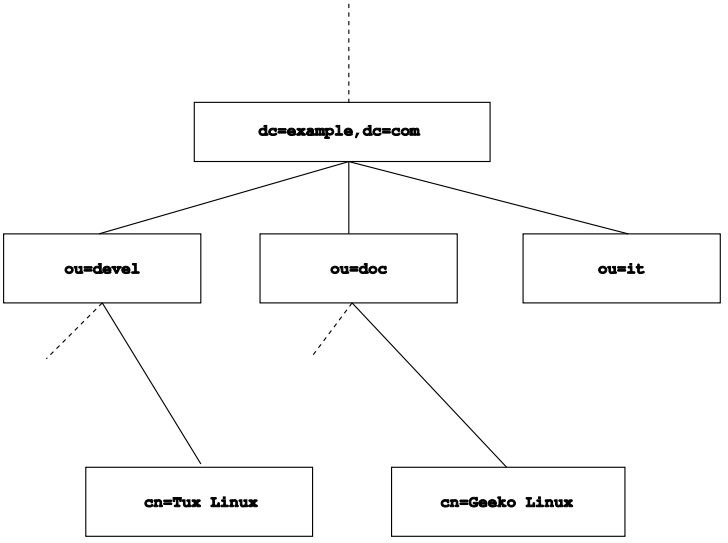
container
: These objects can themselves contain other objects. Such object classes are `root` (the root element of the directory tree, which does not really exist), `c` (country), `ou` (organizational unit), and `dc` (domain component). This model is comparable to the directories (folders) in a file 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 36.1, “Structure of an LDAP Directory” (page 670).

Figure 36.1 *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 36.1, “Commonly Used Object Classes and Attributes” (page 671) 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 36.1 *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 36.1, “Excerpt from schema.core ” (page 672) shows an excerpt from a scheme directive with explanations (line numbering for explanatory reasons).

Example 36.1 *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`.

36.3 Server Configuration with slapd.conf

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.

36.3.1 Global Directives in slapd.conf

Example 36.2 *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 36.2, “slapd.conf: Include Directive for Schemes” (page 673), 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 36.3 *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 with which the `slapd` process is started. There is no need for modifications here.

Example 36.4 *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 36.4, “slapd.conf: Access Control” (page 674) 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 36.2, "User Groups and Their Access Grants" (page 675) are possible.

Table 36.2 *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 36.3, "Types of Access" (page 675).

Table 36.3 *Types of Access*

Tag	Scope of Access
`none`	No access
`auth`	For contacting the server

Tag	Scope of Access
`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 36.5, “slapd.conf: Example for Access Control” (page 676) shows an example of a simple access control that can be arbitrarily developed using regular expressions.

Example 36.5 *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.

36.3.2 Database-Specific Directives in slapd.conf

Example 36.6 *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 36.6, “slapd.conf: Database-Specific Directives” (page 677)).

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.

36.3.3 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 36.4, “Data Handling in the LDAP Directory” (page 679), 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 20.2.3, “Configuring System Services (Runlevel) with YaST” (page 400), 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 20.2.2, “Init Scripts” (page 396).

36.4 Data Handling in the LDAP Directory

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 briefly explained below.

36.4.1 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 36.1, “Structure of an LDAP Directory” (page 670) would look like that in Example 36.7, “Example for an LDIF File” (page 680).

Example 36.7 *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
```

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.

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`. `-f` passes the filename. See the details of running `ldapadd` in Example 36.8, “ldapadd with example.ldif” (page 681).

Example 36.8 *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 36.9, "LDIF Data for Tux" (page 681) adds `Tux` to the new LDAP directory.

Example 36.9 *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.

36.4.2 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 36.10, "Modified LDIF File tux.ldif" (page 682).

Example 36.10 *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.

36.4.3 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)).

36.4.4 Deleting Data from an LDAP Directory

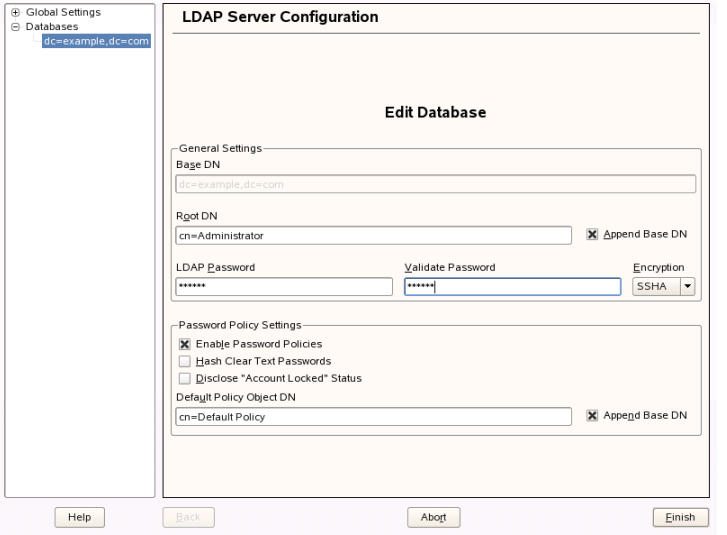
Delete unwanted entries with `ldapdelete`. The syntax is similar to that of the other commands. To delete, for example, the complete entry for `Tux Linux`, issue the following command:

```
ldapdelete -x -D cn=Administrator,dc=example,dc=com -W cn=Tux \
Linux,ou=devel,dc=example,dc=com
```

36.5 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 36.2 *YaST LDAP Server Configuration*



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 nonauthenticated (anonymous) update operations. Access is restricted according to ACLs and other rules (see Section 36.3.1, "Global Directives in slapd.conf" (page 673)).

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 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 grace 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.

36.6 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.

36.6.1 Standard Procedure

Background knowledge of the processes acting in the background of a client machine helps you understand how the YaST LDAP client module works. If LDAP is activated for network authentication or the YaST module is called, the packages `pam_ldap` and `nss_ldap` are installed and the two corresponding configuration files are adapted. `pam_ldap` is the PAM module responsible for negotiation between login processes and the LDAP directory as the source of authentication data. The dedicated module `pam_ldap.so` is installed and the PAM configuration is adapted (see Example 36.11, "pam_unix2.conf Adapted to LDAP" (page 689)).

Example 36.11 *pam_unix2.conf Adapted to LDAP*

```
auth:       use_ldap
account:    use_ldap
password:   use_ldap
session:    none
```

When manually configuring additional services to use LDAP, include the PAM LDAP module in the PAM configuration file corresponding to the service in `/etc/pam.d`. Configuration files already adapted to individual services can be found in `/usr/share/doc/packages/pam_ldap/pam.d/`. Copy appropriate files to `/etc/pam.d`.

`glibc` name resolution through the `nsswitch` mechanism is adapted to the employment of LDAP with `nss_ldap`. A new, adapted file `nsswitch.conf` is created in `/etc` with the installation of this package. Find more about the workings of `nsswitch.conf` in Section 30.7.1, "Configuration Files" (page 589). The following lines must be present in `nsswitch.conf` for user administration and authentication with LDAP. See Example 36.12, "Adaptations in nsswitch.conf" (page 689).

Example 36.12 *Adaptations in nsswitch.conf*

```
passwd: compat
group: compat

passwd_compat: ldap
group_compat: ldap
```

These lines order the resolver library of `glibc` first to evaluate the corresponding files in `/etc` and additionally access the LDAP server as sources for authentication and user data. Test this mechanism, for example, by reading the content of the user database

with the command `getent passwd`. The returned set should contain a survey of the local users of your system as well as all users stored on the LDAP server.

To prevent regular users managed through LDAP from logging in to the server with `ssh` or `login`, the files `/etc/passwd` and `/etc/group` each need to include an additional line. This is the line `+:::::::/sbin/nologin` in `/etc/passwd` and `+:::` in `/etc/group`.

36.6.2 Configuring the LDAP Client

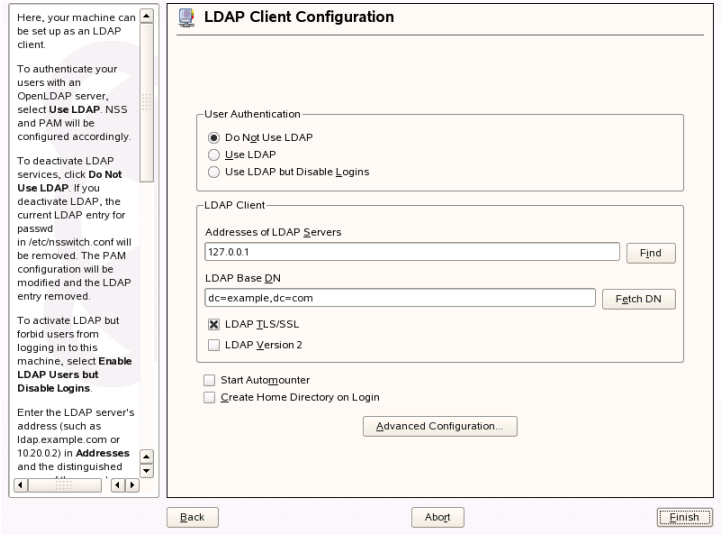
After the initial adjustments of `nss_ldap`, `pam_ldap`, `/etc/passwd`, and `/etc/group` have been taken care of by YaST, you can simply connect your client to the server and let YaST manage users over LDAP. This basic setup is described in Section "Basic Configuration" (page 690).

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 "Configuring the YaST Group and User Administration Modules" (page 694).

Basic Configuration

The basic LDAP client configuration dialog (Figure 36.3, "YaST: Configuration of the LDAP Client" (page 691)) 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 36.3 *YaST: Configuration of the LDAP Client*



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*.

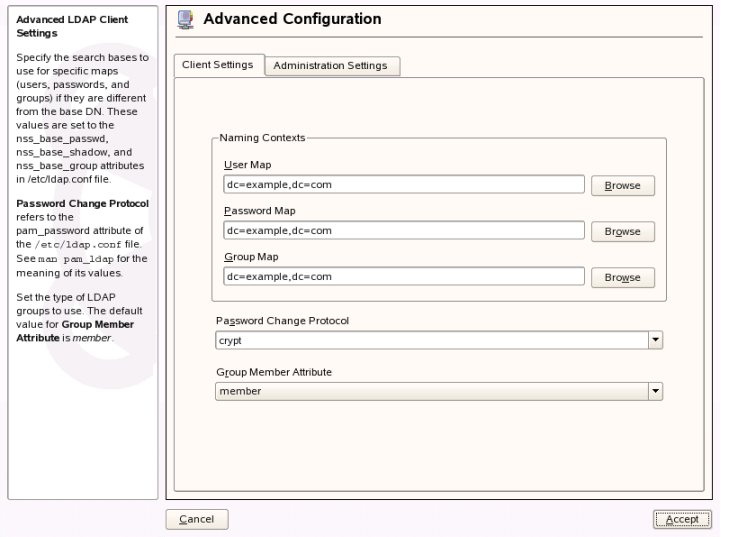
5. If the LDAP server still uses LDAPv2, explicitly enable the use of this protocol version by selecting *LDAP Version 2*.

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.

Figure 36.4 *YaST: Advanced Configuration*



To modify data on the server as administrator, click *Advanced Configuration*. The following dialog is split in two tabs. See Figure 36.4, “YaST: Advanced Configuration” (page 692).

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.

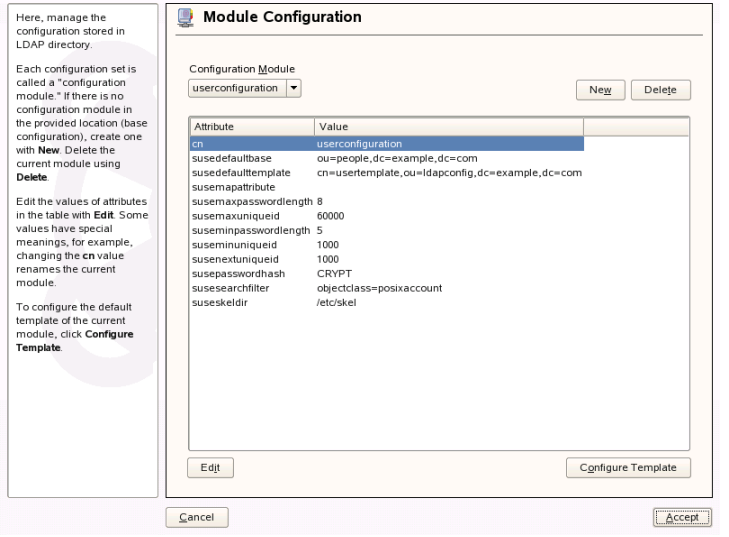
2f Click *Accept* to leave the *Advanced Configuration* then *Finish* to apply your settings.

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 “Configuring the YaST Group and User Administration Modules” (page 694).

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 36.5 *YaST: Module Configuration*



The dialog for module configuration (Figure 36.5, “YaST: Module Configuration” (page 694)) 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.

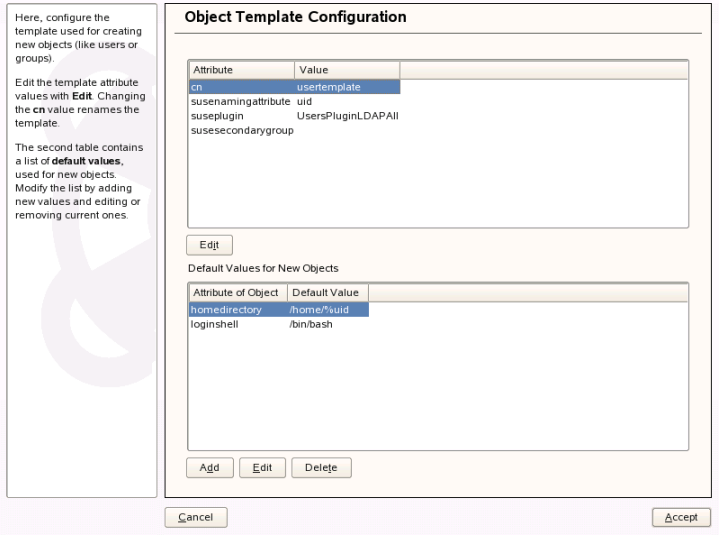
The YaST modules for group and user administration embed templates with sensible standard values. To edit a template associated with a configuration module, proceed as follows:

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 36.6 *YaST: Configuration of an Object Template*



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.

36.7 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.

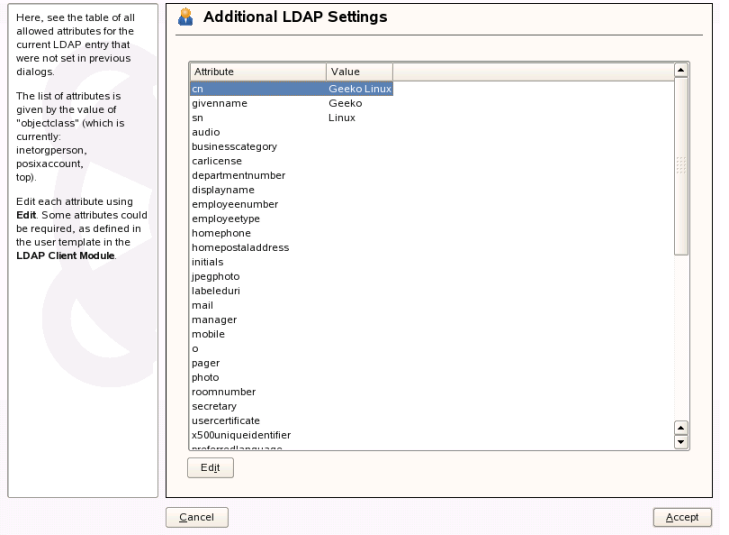
   3b Check the *Details* tab for the group membership, login shell, and home directory of the new user. If necessary, change the default to values that better suit your needs. The default values as well as those of the password settings can be defined with the procedure described in Section "Configuring the YaST Group and User Administration Modules" (page 694).

   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 36.7, "YaST: Additional LDAP Settings" (page 698)).

4. Click *Accept* to apply your settings and leave the user configuration.

Figure 36.7 *YaST: Additional LDAP Settings*



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*.

36.8 Browsing the LDAP Directory Tree

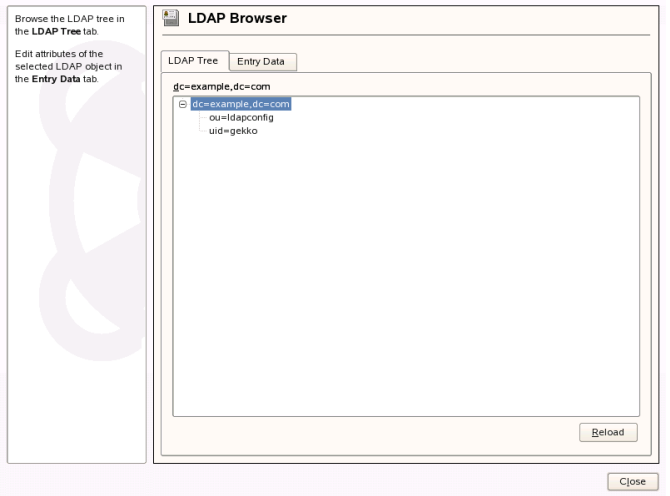
To browse the LDAP directory tree and all its entries conveniently, use the YaST LDAP Browser:

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.

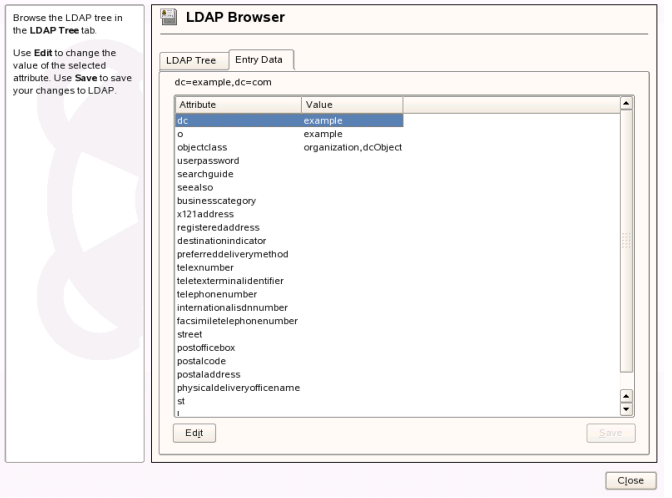
Figure 36.8 *Browsing the LDAP Directory Tree*



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 36.9 *Browsing the Entry Data*



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*.

36.9 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.

37 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.

37.1 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.

TIP: IBM System z: NetBIOS Support

IBM System z merely support SMB over TCP/IP. NetBIOS support is not available on these systems.

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.

37.2 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 37.3.1, “Configuring a Samba Server with YaST” (page 705).

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`.

37.3 Configuring a Samba Server

A samba server in SUSE Linux Enterprise® 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.

37.3.1 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 37.2, "Starting and Stopping Samba" (page 705).

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, for example LDAP, click *Advanced Settings*.

Users from Other Domains

To enable users from other domains to access your domain, make the appropriate settings in the *Trusted Domains* tab. To add a new domain, click *Add*. To remove the selected domain, click *Delete*.

Using LDAP

In the tab *LDAP Settings*, you can determine the LDAP server to use for authentication. To test the connection to your LDAP server, click *Test Connection*. To set expert LDAP settings or use default values, click *Advanced Settings*.

Find more information about LDAP configuration in Chapter 36, *LDAP—A Directory Service* (page 667).

37.3.2 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)*.

37.3.3 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 37.1 *A CD-ROM Share*

```
;[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 37.2 *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`.

37.4 Configuring Clients

Clients can only access the Samba server via TCP/IP. NetBEUI and NetBIOS via IPX cannot be used with Samba.

37.4.1 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.

37.4.2 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.

37.5 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 37.3, “Global Section in smb.conf” (page 712).

Example 37.3 *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 37.4 *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 37.5 *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`.

37.6 Samba Server in the Network with Active Directory

If you run Linux servers and Windows servers together, you can build two independent authentication systems and networks or connect servers to one network with one central

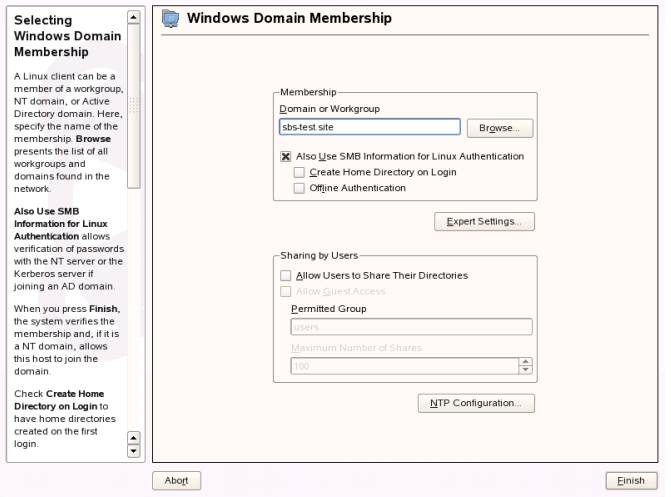
authentication system. Because Samba can cooperate with an active directory domain, you can join your SUSE Linux Enterprise Server to Active Directory (AD).

Join an existing AD domain during installation or by later activating SMB user authentication with YaST in the installed system. Domain join during installation is covered in Section 3.14.7, “Users” (page 43).

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. Alternately, use *Browse* to get a list of all available domains and select one.

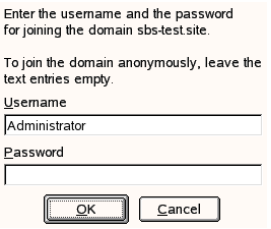
Figure 37.1 *Determining Windows Domain Membership*



4. Check *Also Use SMB Information for Linux Authentication* to use the SMB source for Linux authentication on your SUSE Linux Enterprise Server.
5. Click *Finish* and confirm the domain join when prompted for it.

6 Provide the password for the Windows Administrator on the AD server and click *OK*.

Figure 37.2 *Providing Administrator Credentials*



Your server is now set up to pull in all authentication data from the Active Directory domain controller.

37.7 Migrating a Windows NT Server to Samba

Apart from the Samba and LDAP configuration, the migration of a Windows NT server to a SUSE Linux Enterprise Server Samba server consists of two basic steps. First, migrate profiles then migrate accounts.

37.7.1 Preparing the LDAP Server

The first step of your migration should be the configuration of the LDAP server. You need to add base DN information and entries for accounts of your software clients with passwords. Detailed information about LDAP configuration is provided in Chapter 36, *LDAP—A Directory Service* (page 667).

It is not necessary to configure it all manually. You can use scripts from smbldap-tools. These scripts are part of the package `samba-doc` and, after installation of the package, are available in `/usr/share/doc/packages/samba/examples/LDAP`.

NOTE: LDAP and Security

The LDAP administration DN should be an account other than Root DN. To make the network more secure, you can also use a secure connection with TSL.

37.7.2 Preparing the Samba Server

Before you start migration, configure your Samba server. Find configuration of `profile`, `netlogon`, and `home` shares in the *Shares* tab of the YaST *Samba Server* module. To do the default value, select the share and click *Edit*.

To add LDAP configuration for your Samba server and the credentials of the LDAP administrator, use the *LDAP Settings* tab of the YaST *Samba Server* module. The LDAP administration DN (label *Administration DN*) and password are essential to add or modify accounts stored in the LDAP directory.

37.7.3 Migrating the Windows Profiles

For every profile to migrate, complete these steps:

Procedure 37.1 *Migrating a Profile*

1. On your NT4 domain controller, right-click *My Computer* then select *Properties*. Select the *User Profiles* tab.
2. Select a user profile you to migrate and click it.
3. Click *Copy To*.
4. In *Copy Profile*, add your new path, for example, `c:\temp\profiles`.
5. Click *Change* in *Permitted*.
6. Click *Everyone*. To close the box, click *OK*.
7. To finish saving the profile, click *OK*.
8. Copy saved profiles to the appropriate profile directories on your Samba server.

37.7.4 Migrating the Windows Accounts

Procedure 37.2 *The Account Migration Process*

1. Create a BDC account in the old NT4 domain for the Samba server using NT Server Manager. Samba must not be running.

```
     net rpc join -S NT4PDC -w DOMNAME -U Administrator%passwd net rpc
vampire
     -S NT4PDC -U administrator%passwd pdbedit -L
```

2. Assign each of the UNIX groups to NT groups:

Example 37.6 *Example Script initGroups.sh*

```
#!/bin/bash #### Keep this as a shell script for future re-use #
      Known domain global groups net groupmap modify ntgroup="Domain
Admins"
      unixgroup=root net groupmap modify ntgroup="Domain Users"
      unixgroup=users net groupmap modify ntgroup="Domain Guests"
      unixgroup=nobody # Our domain global groups net groupmap add
      ntgroup="Operation" unixgroup=operation type=d net groupmap add
      ntgroup="Shipping" unixgroup=shipping type=d
```

3. Check that all groups are recognized:

```
net groupmap list
```

37.8 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.

38 Sharing File Systems with NFS

Distributing and sharing file systems over a network is a common task in corporate environments. `NFS` is a proven system that also works together with the yellow pages protocol `NIS`. For a more secure protocol that works together with `LDAP` and may also be kerberized, check `NFSv4`.

NFS works with NIS to make a network transparent to the user. With NFS, it is possible to distribute arbitrary file systems over the network. With an appropriate setup, users always find themselves in the same environment independent of the terminal they currently use.

Like NIS, NFS is a client/server system. However, a machine can be both—it can supply file systems over the network (export) and mount file systems from other hosts (import).

IMPORTANT: Need for DNS

In principle, all exports can be made using IP addresses only. To avoid timeouts, you should have a working DNS system. This is necessary at least for logging purposes, because the mountd daemon does reverse lookups.

38.1 Installing the Required Software

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.

NFS server software is not part of the default installation. To install the NFS server software, start YaST and select *Software* > *Software Management*. Now choose *Filter* > *Patterns* and select *Misc. Server* or use the *Search* option and search for `NFS Server`. Confirm the installation of the packages to finish the installation process.

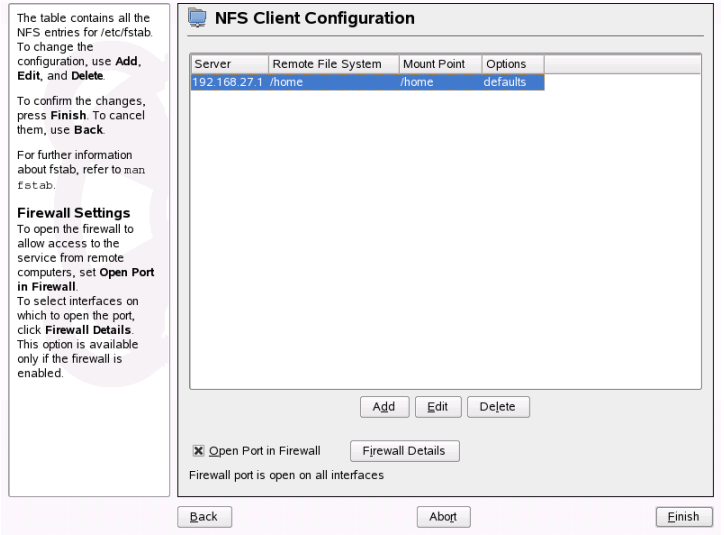
38.2 Importing File Systems with YaST

Users authorized to do so can mount NFS directories from an NFS server into their own file trees. This can be achieved 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. The changes will take effect after *Add* is clicked in the first dialog. Click *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. Clicking *Finish* to saves your changes. See Figure 38.1, “NFS Client Configuration with YaST” (page 721).

The configuration is written to `/etc/fstab` and the specified file systems are mounted. When you start the YaST configuration client at a later point in time, it also reads the existing configuration from this file.

An NFSv4 file system can currently only be imported manually. This is explained in Section 38.3, “Importing File Systems Manually” (page 721).

Figure 38.1 *NFS Client Configuration with YaST*



38.3 Importing File Systems Manually

File systems can also be imported manually from an NFS server. The prerequisite for this is a running RPC port mapper, which can be started 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 user directories from the machine sun, for example, should be imported, use the following command:

```
mount sun:/home /home
```

38.3.1 Importing NFSv4 File Systems

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 `rcidmapd start`. Use `rcidmapd status` to check the status of idmapd.

The idmapd services stores its parameters in the `/etc/idmapd.conf` file. Leave the value of the `Domain` parameter as `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 sun to `/local/home`, use the following command:

```
mount -t nfs4 sun:/ /local/home
```

The remote file system path that follows the server name and a colon is 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 38.4.1, "Exporting for NFSv4 Clients" (page 725).

38.3.2 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:/
```

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`.

38.3.3 Manually Editing /etc/fstab

A typical NFS mount entry in `/etc/fstab` looks like this:

```
host:/data /local/path nfs rw,noauto 0 0
```

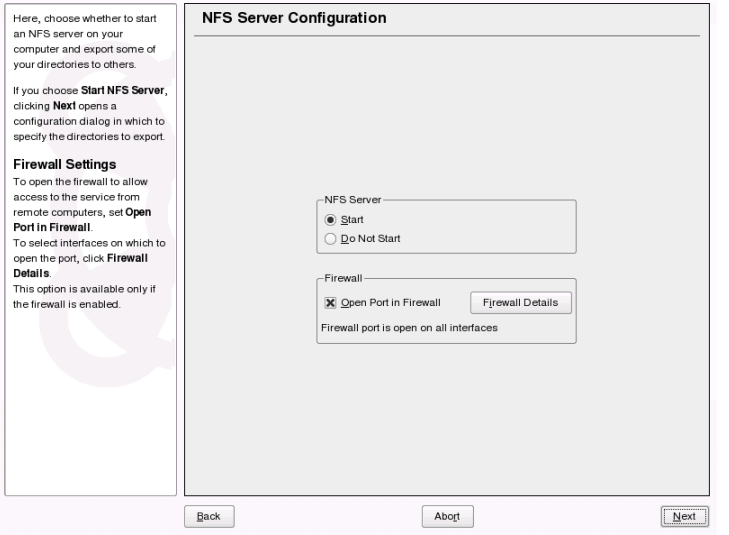
NFSv4 mounts may also be added to the `/etc/fstab` file manually. For these mounts, use `nfs4` instead of `nfs` in the third column and make sure that the remote file system is given as `/` after the *`host:`* in the first column. The advantage of saving this information in `/etc/fstab` is that commands for mounting can be shortened to just mentioning the local mount point alone, for example:

```
mount /local/path
```

38.4 Exporting File Systems with YaST

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 38.2, “NFS Server Configuration Tool” (page 724) opens.

Figure 38.2 *NFS Server Configuration Tool*

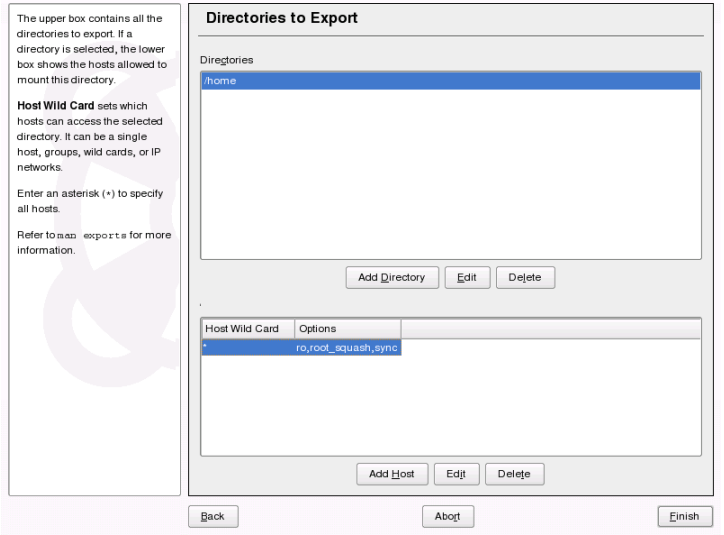


Next, activate *Start NFS Server* and enter the *NFSv4 domain name*.

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 them. This dialog is shown in Figure 38.3, “Configuring an NFS Server with YaST” (page 725). The figure shows the scenario where NFSv4 is enabled in the previous dialog. `Bindmount Targets` is shown in the right pane. For more details, refer to the help shown on the left pane. In the lower half of the dialog, there are four options that can be set for each host: `single host`, `netgroups`, `wildcards`, and `IP networks`. For a more thorough explanation of these options, refer to `exports` man page. Click *Finish* to complete the configuration.

Figure 38.3 *Configuring an NFS Server with YaST*



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.

38.4.1 Exporting for NFSv4 Clients

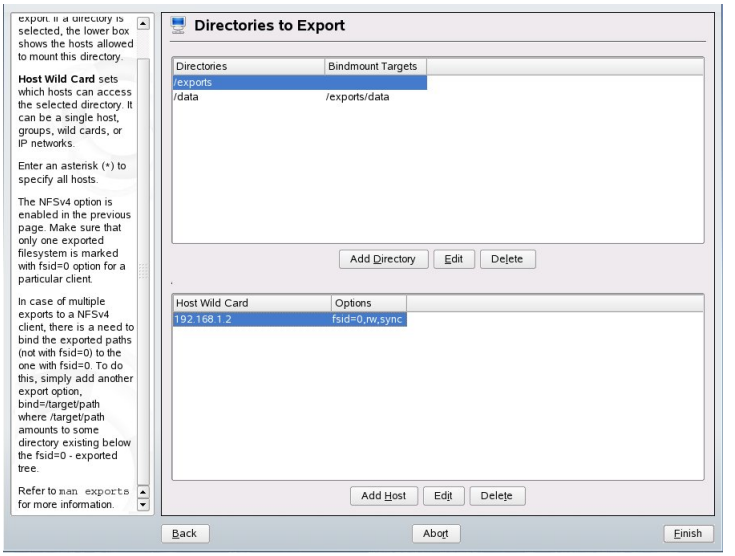
Activate *Enable NFSv4* to support NFSv4 clients. Clients with NFSv3 can still access the server's exported directories if they are exported appropriately. This is explained in detail in Section 38.4.3, “Coexisting v3 and v4 Exports” (page 728).

After activating NFSv4, enter an appropriate domain name. Make sure that the name entered is the same as the one present in the `/etc/idmapd.conf` file of any NFSv4 client that accesses this particular server. This parameter is for the idmapd service that is required for NFSv4 support (on both server and client). Leave it as `localdomain` (the default) if you do not have special requirements. For more information, see Section 38.7, “For More Information” (page 732).

Click *Next*. The dialog that follows has two sections. The upper half consists of two columns named *Directories* and *Bind mount targets*. *Directories* is a directly editable column that lists the directories to export.

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.

Figure 38.4 *Exporting Directories with NFSv4*



In the lower half of the dialog, enter the client (wild card) and export options for a particular directory. After adding a directory in the upper half, another dialog for entering the client and option information pops up automatically. After that, 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 column, instead summarizing directories and their nature. After the information is complete, click *Finish* to complete the configuration or *Start* to restart the service.

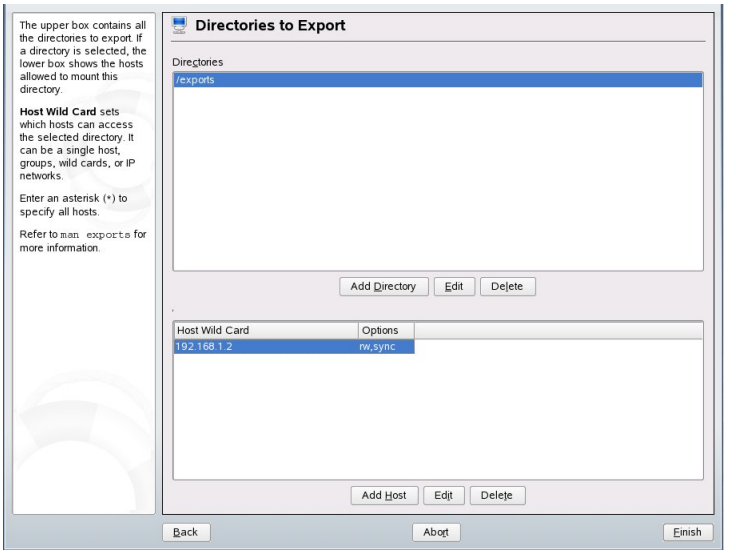
38.4.2 NFSv3 and NFSv2 Exports

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 38.4, “Exporting Directories with NFSv4” (page 726). Find a more thorough explanation of these options in `man exports`. Click *Finish* to complete the configuration.

Figure 38.5 *Exporting Directories with NFSv2 and v3*



38.4.3 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 38.4, “Exporting Directories with NFSv4” (page 726). 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.

38.5 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` and `rcidmapd restart`. The NFS server depends on a running RPC portmapper. Therefore, also start or restart the portmapper service with `rcportmap restart`.

38.5.1 Exporting File Systems with NFSv4

NFSv4 is the latest version of NFS protocol available on SUSE Linux Enterprise 10. Configuring the directories for export with NFSv4 differs slightly from the previous versions.

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, `/exports` 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 the services are running.

38.5.2 Exporting File Systems with NFSv2 and NFSv3

This is specific to NFSv3 and NFSv2 exports. Refer to Section 38.5.1, “Exporting File Systems with NFSv4” (page 729) 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 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`.

38.6 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. Additionally complete the following steps:

- 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 38.7, “For More Information” (page 732).

38.7 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` and these 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.

39 File Synchronization

Today, many people use several computers—one computer at home, one or several computers at the workplace, and possibly a laptop or PDA on the road. Many files are needed on all these computers. You may want to be able to work with all computers and modify the files and subsequently have the latest version of the data available on all computers.

39.1 Available Data Synchronization Software

Data synchronization is no problem for computers that are permanently linked by means of a fast network. In this case, use a network file system, like NFS, and store the files on a server, enabling all hosts to access the same data via the network. This approach is impossible if the network connection is poor or not permanent. When you are on the road with a laptop, copies of all needed files must be on the local hard disk. However, it is then necessary to synchronize modified files. When you modify a file on one computer, make sure a copy of the file is updated on all other computers. For occasional copies, this can be done manually with scp or rsync. However, if many files are involved, the procedure can be complicated and requires great care to avoid errors, such as overwriting a new file with an old file.

WARNING: Risk of Data Loss

Before you start managing your data with a synchronization system, you should be well acquainted with the program used and test its functionality. A backup is indispensable for important files.

The time-consuming and error-prone task of manually synchronizing data can be avoided by using one of the programs that use various methods to automate this job. The following summaries are merely intended to convey a general understanding of how these programs work and how they can be used. If you plan to use them, read the program documentation.

39.1.1 CVS

CVS, which is mostly used for managing program source versions, offers the possibility to keep copies of the files on multiple computers. Accordingly, it is also suitable for data synchronization. CVS maintains a central repository on the server in which the files and changes to files are saved. Changes that are performed locally are committed to the repository and can be retrieved from other computers by means of an update. Both procedures must be initiated by the user.

CVS is very resilient to errors when changes occur on several computers. The changes are merged and, if changes took place in the same lines, a conflict is reported. When a conflict occurs, the database remains in a consistent state. The conflict is only visible for resolution on the client host.

39.1.2 rsync

When no version control is needed but large directory structures need to be synchronized over slow network connections, the tool rsync offers well-developed mechanisms for transmitting only changes within files. This not only concerns text files, but also binary files. To detect the differences between files, rsync subdivides the files into blocks and computes checksums over them.

The effort put into the detection of the changes comes at a price. The systems to synchronize should be scaled generously for the usage of rsync. RAM is especially important.

39.2 Determining Factors for Selecting a Program

There are some important factors to consider when deciding which program to use.

39.2.1 Client-Server versus Peer-to-Peer

Two different models are commonly used for distributing data. In the first model, all clients synchronize their files with a central server. The server must be accessible by all clients at least occasionally. This model is used by CVS.

The other possibility is to let all networked hosts synchronize their data between each other as peers. rsync actually works in client mode, but any client can also act as a server.

39.2.2 Portability

CVS and rsync are also available for many other operating systems, including various Unix and Windows systems.

39.2.3 Interactive versus Automatic

In CVS, the data synchronization is started manually by the user. This allows fine control over the data to synchronize and easy conflict handling. However, if the synchronization intervals are too long, conflicts are more likely to occur.

39.2.4 Conflicts: Incidence and Solution

Conflicts only rarely occur in CVS, even when several people work on one large program project. This is because the documents are merged on the basis of individual lines. When a conflict occurs, only one client is affected. Usually conflicts in CVS can easily be resolved.

There is no conflict handling in rsync. The user is responsible for not accidentally overwriting files and manually resolving all possible conflicts. To be on safe side, a versioning system like RCS can be additionally employed.

39.2.5 Selecting and Adding Files

In CVS, new directories and files must be added explicitly using the command `cvs add`. This results in greater user control over the files to synchronize. On the other hand, new files are often overlooked, especially when the question marks in the output of `cvs update` are ignored due to the large number of files.

39.2.6 History

An additional feature of CVS is that old file versions can be reconstructed. A brief editing remark can be inserted for each change and the development of the files can easily be traced later based on the content and the remarks. This is a valuable aid for theses and program texts.

39.2.7 Data Volume and Hard Disk Requirements

A sufficient amount of free space for all distributed data is required on the hard disks of all involved hosts. CVS require additional space for the repository database on the server. The file history is also stored on the server, requiring even more space. When files in text format are changed, only the modified lines need to be saved. Binary files require additional space amounting to the size of the file every time the file is changed.

39.2.8 GUI

Experienced users normally run CVS from the command line. However, graphical user interfaces are available for Linux, such as cervisia, and for other operating systems, like wincvs. Many development tools, such as kdevelop, and text editors, such as Emacs, provide support for CVS. The resolution of conflicts is often much easier to perform with these front-ends.

39.2.9 User Friendliness

rsync is rather easy to use and is also suitable for newcomers. CVS is somewhat more difficult to operate. Users should understand the interaction between the repository and local data. Changes to the data should first be merged locally with the repository. This is done with the command `cvs update`. Then the data must be sent back to the repository with the command `cvs commit`. Once this procedure has been understood, newcomers are also able to use CVS with ease.

39.2.10 Security against Attacks

During transmission, the data should ideally be protected against interception and manipulation. CVS and rsync can easily be used via ssh (secure shell), providing security against attacks of this kind. Running CVS via rsh (remote shell) should be avoided. Accessing CVS with the *pserver* mechanism in insecure networks is likewise not advisable.

39.2.11 Protection against Data Loss

CVS has been used by developers for a long time to manage program projects and is extremely stable. Because the development history is saved, CVS even provides protection against certain user errors, such as unintentional deletion of a file.

Table 39.1 *Features of the File Synchronization Tools: -- = very poor, - = poor or not available, o = medium, + = good, ++ = excellent, x = available*

	CVS	**rsync**
Client/Server	C-S	C-S
Portability	Lin,Un*x,Win	Lin,Un*x,Win
Interactivity	x	x
Speed	o	+
Conflicts	++	o

	CVS	rsync
File Sel.	Sel./file, dir.	Dir.
History	x	-
Hard Disk Space	--	o
GUI	o	-
Difficulty	o	+
Attacks	+ (ssh)	+(ssh)
Data Loss	++	+

39.3 Introduction to CVS

CVS is suitable for synchronization purposes if individual files are edited frequently and are stored in a file format, such as ASCII text or program source text. The use of CVS for synchronizing data in other formats, such as JPEG files, is possible, but leads to large amounts of data, because all variants of a file are stored permanently on the CVS server. In such cases, most of the capabilities of CVS cannot be used. The use of CVS for synchronizing files is only possible if all workstations can access the same server.

39.3.1 Configuring a CVS Server

The *server* is the host on which all valid files are located, including the latest versions of all files. Any stationary workstation can be used as a server. If possible, the data of the CVS repository should be included in regular backups.

When configuring a CVS server, it might be a good idea to grant users access to the server via SSH. If the user is known to the server as `tux` and the CVS software is installed on the server as well as on the client, the following environment variables must be set on the client side:

```
CVS_RSH=ssh CVSROOT=tux@server:/serverdir
```

The command `cvs init` can be used to initialize the CVS server from the client side. This needs to be done only once.

Finally, the synchronization must be assigned a name. Select or create a directory on the client exclusively to contain files to manage with CVS (the directory can also be empty). The name of the directory is also the name of the synchronization. In this example, the directory is called `synchome`. Change to this directory and enter the following command to set the synchronization name to `synchome`:

```
cvs import synchome tux wilber
```

Many CVS commands require a comment. For this purpose, CVS starts an editor (the editor defined in the environment variable `$EDITOR` or vi if no editor was defined). The editor call can be circumvented by entering the comment in advance on the command line, such as in the following example:

```
cvs import -m 'this is a test' synchome tux wilber
```

39.3.2 Using CVS

The synchronization repository can now be checked out from all hosts with `cvs co synchome`. This creates a new subdirectory `synchome` on the client. To commit your changes to the server, change to the directory `synchome` (or one of its subdirectories) and enter `cvs commit`.

By default, all files (including subdirectories) are committed to the server. To commit only individual files or directories, specify them as in `cvs commit file1 directory1`. New files and directories must be added to the repository with a command like `cvs add file1 directory1` before they are committed to the server. Subsequently, commit the newly added files and directories with `cvs commit file1 directory1`.

If you change to another workstation, check out the synchronization repository if this has not been done during an earlier session at the same workstation.

Start the synchronization with the server with `cvs update`. Update individual files or directories as in `cvs update file1 directory1`. To see the difference between the current files and the versions stored on the server, use the command `cvs diff` or `cvs diff file1 directory1`. Use `cvs -nq update` to see which files would be affected by an update.

Here are some of the status symbols displayed during an update:

U
: The local version was updated. This affects all files that are provided by the server and missing on the local system.

M
: The local version was modified. If there were changes on the server, it was possible to merge the differences in the local copy.

P
: The local version was patched with the version on the server.

C
: The local file conflicts with current version in the repository.

?
: This file does not exist in CVS.

The status `M` indicates a locally modified file. Either commit the local copy to the server or remove the local file and run the update again. In this case, the missing file is retrieved from the server. If you commit a locally modified file and the file was changed in the same line and committed, you might get a conflict, indicated with `C`.

In this case, look at the conflict marks (»> and «<) in the file and decide between the two versions. As this can be a rather unpleasant job, you might decide to abandon your changes, delete the local file, and enter `cvs up` to retrieve the current version from the server.

39.3.3 For More Information

This section merely offers a brief introduction to the many possibilities of CVS. Extensive documentation is available at the following URLs:

- CVS: http://www.cvshome.org
- Rsync: http://www.gnu.org/manual

39.4 Introduction to rsync

rsync is useful when large amounts of data need to be transmitted regularly while not changing too much. This is, for example, often the case when creating backups. Another application concerns staging servers. These are servers that store complete directory trees of Web servers that are regularly mirrored onto a Web server in a DMZ.

39.4.1 Configuration and Operation

rsync can be operated in two different modes. It can be used to archive or copy data. To accomplish this, only a remote shell, like ssh, is required on the target system. However, rsync can also be used as a daemon to provide directories to the network.

The basic mode of operation of rsync does not require any special configuration. rsync directly allows mirroring complete directories onto another system. As an example, the following command creates a backup of the home directory of tux on a backup server named sun:

```
rsync -baz -e ssh /home/tux/ tux@sun:backup
```

The following command is used to play the directory back:

```
rsync -az -e ssh tux@sun:backup /home/tux/
```

Up to this point, the handling does not differ much from that of a regular copying tool, like scp.

rsync should be operated in “rsync” mode to make all its features fully available. This is done by starting the rsyncd daemon on one of the systems. Configure it in the file `/etc/rsyncd.conf`. For example, to make the directory `/srv/ftp` available with rsync, use the following configuration:

```
gid = nobody
uid = nobody
read only = true
use chroot = no
transfer logging = true
log format = %h %o %f %l %b
log file = /var/log/rsyncd.log

[FTP]
        path = /srv/ftp
        comment = An Example
```

Then start rsyncd with `rcrsyncd start`. rsyncd can also be started automatically during the boot process. Set this up by activating this service in the runlevel editor provided by YaST or by manually entering the command `insserv rsyncd`. rsyncd can alternatively be started by xinetd. This is, however, only recommended for servers that rarely use rsyncd.

The example also creates a log file listing all connections. This file is stored in `/var/log/rsyncd.log`.

It is then possible to test the transfer from a client system. Do this with the following command:

```
rsync -avz sun::FTP
```

This command lists all files present in the directory `/srv/ftp` of the server. This request is also logged in the log file `/var/log/rsyncd.log`. To start an actual transfer, provide a target directory. Use `.` for the current directory. For example:

```
rsync -avz sun::FTP .
```

By default, no files are deleted while synchronizing with rsync. If this should be forced, the additional option `--delete` must be stated. To ensure that no newer files are deleted, the option `--update` can be used instead. Any conflicts that arise must be resolved manually.

39.4.2 For More Information

Important information about rsync is provided in the man pages `man rsync` and `man rsyncd.conf`. A technical reference about the operating principles of rsync is featured in `/usr/share/doc/packages/rsync/tech_report.ps`. Find the latest news about rsync on the project Web site at http://rsync.samba.org/.

If you want Subversion or other tools, download the the SDK. Find it at http://developer.novell.com/wiki/index.php/SUSE_LINUX_SDK.

40 The Apache HTTP Server

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. SUSE® Linux Enterprise Server 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.

40.1 Quick Start

With the help of this section, quickly set up and start Apache. time. You must be `root` to install and configure Apache.

40.1.1 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 30, *Basic Networking* (page 547).

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 32, *Time Synchronization with NTP* (page 609) 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 43.4.1, “Configuring the Firewall with YaST” (page 829) gives details.

40.1.2 Installation

Apache on SUSE Linux Enterprise Server 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 *Primary 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 40.4, “Installing, Activating, and Configuring Modules” (page 763) for more information about modules.

40.1.3 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 SUSE Linux Enterprise Server and a description of the YaST runlevel editor, refer to Section 20.2.3, “Configuring System Services (Runlevel) with YaST” (page 400).

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 40.8, "Troubleshooting" (page 781).

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.

40.2 Configuring Apache

Apache in SUSE Linux Enterprise Server 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 40.3, "Starting and Stopping Apache" (page 761). Most configuration changes only require a reload with `rcapache2 reload`.

40.2.1 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 751) 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 40.4, “Installing, Activating, and Configuring Modules” (page 763) 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 40.4.4, “Multiprocessing Modules” (page 768)) 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 40.6, “Setting Up a Secure Web Server with SSL” (page 773) 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 750) 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 758)) or by manually editing a configuration file. By default, Apache in SUSE Linux Enterprise Server 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 40.1 *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 40.2 *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 40.3 *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 40.4 *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>
```

40.2.2 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 759) 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

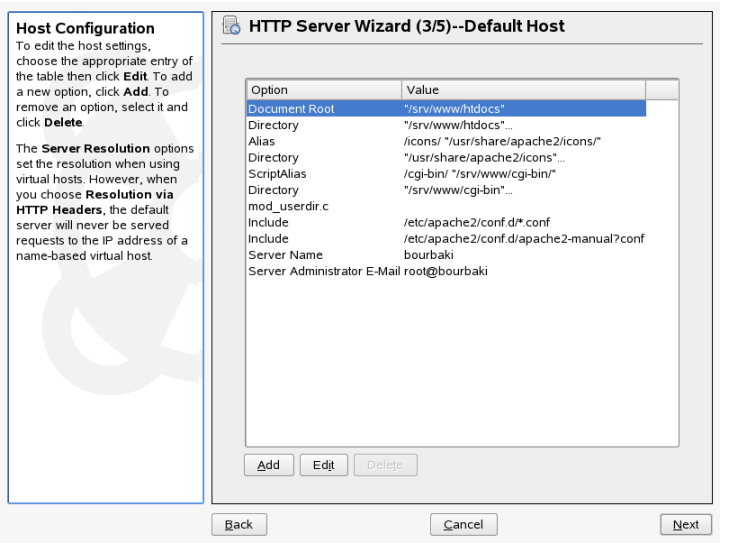
The *Modules* configuration option allows for the activation or deactivation of the script languages, the web server should support. For the activation or deactivation of other modules, refer to Section "Server Modules" (page 760). Click *Next* to advance to the next dialog.

Default Host

This option pertains to the default Web server. As explained in Section “Virtual Host Configuration” (page 750), 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 40.1 *HTTP Server Wizard: Default Host*



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 SUSE Linux Enterprise `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 750)). 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 40.6.2, "Configuring Apache with SSL" (page 778) 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`.

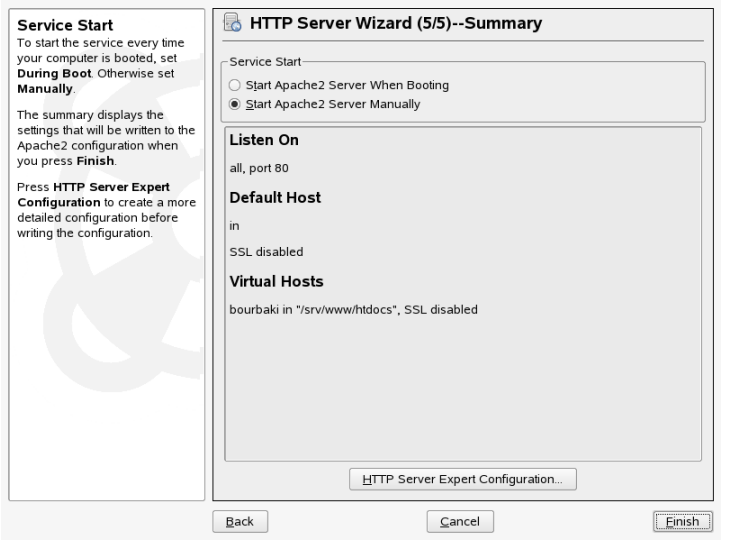
IMPORTANT: Creating Virtual Hosts

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 759).

Figure 40.2 *HTTP Server Wizard: Summary*



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.

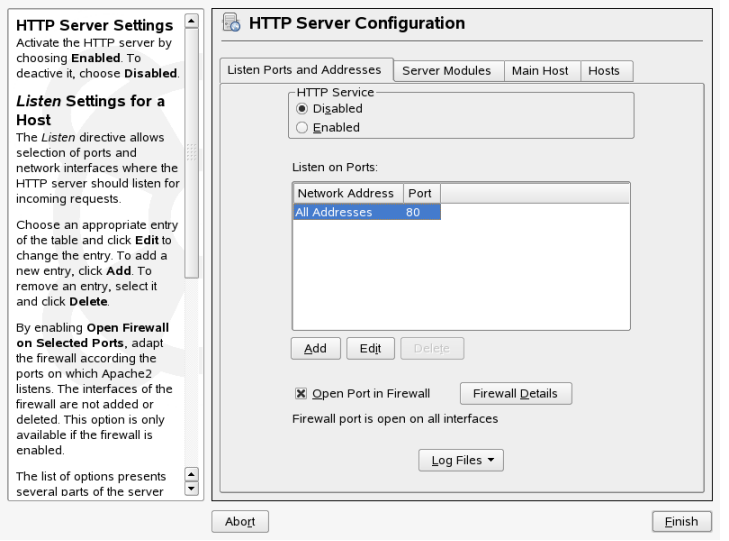
Listen Ports and Addresses

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 40.3, “Starting and Stopping Apache” (page 761) for details). These commands are effective immediately.

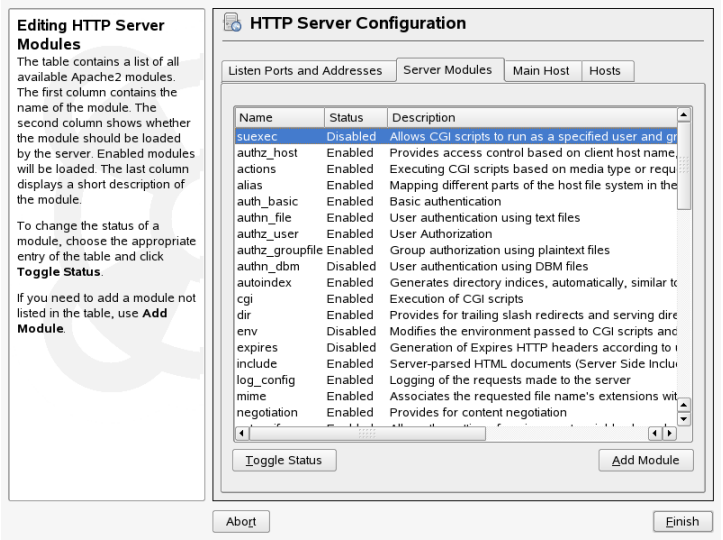
Figure 40.3 *HTTP Server Configuration: Listen Ports and Addresses*



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 40.4, “Installing, Activating, and Configuring Modules” (page 763).

Figure 40.4 *HTTP Server Configuration: Server Modules*



Main Host or Hosts

These dialogs are identical to the ones already described. Refer to Section “Default Host” (page 756) and Section “Virtual Hosts” (page 758).

40.3 Starting and Stopping Apache

If configured with YaST (see Section 40.2.2, “Configuring Apache with YaST” (page 755)), 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 20.2.2, “Init Scripts” (page 396) 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 40.6, “Setting Up a Secure Web Server with SSL” (page 773).

`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.

40.4 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 40.4.2, “Activation and Deactivation” (page 764) 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 SUSE Linux Enterprise Server, 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. SUSE Linux Enterprise Server 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.

40.4.1 Module Installation

If you have followed the default way of installing Apache (described in Section 40.1.2, "Installation" (page 746)), 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.

40.4.2 Activation and Deactivation

Using YaST, you can activate or deactivate the script language modules (PHP5, Perl, Python) with the module configuration described in Section "HTTP Server Wizard" (page 755). All other modules can be enabled or disabled as described in Section "Server Modules" (page 760).

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.

40.4.3 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 40.6, “Setting Up a Secure Web Server with SSL” (page 773) 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.

40.4.4 Multiprocessing Modules

SUSE Linux Enterprise Server 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.

40.4.5 External Modules

Find a list of all external modules shipped with SUSE Linux Enterprise Server 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_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`

40.4.6 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.

40.5 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 40.4.2, “Activation and Deactivation” (page 764) for details on activating modules.

WARNING: CGI Security

Allowing the server to execute CGI scripts is a potential security hole. Refer to Section 40.7, “Avoiding Security Problems” (page 779) for additional information.

40.5.1 Apache Configuration

In SUSE Linux Enterprise Server, 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 750)) and want to place your scripts in a host-specific directory, you must unlock and configure this directory.

Example 40.5 *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.

40.5.2 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 40.7, "Avoiding Security Problems" (page 779) 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".

40.5.3 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.

40.6 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 SUSE Linux Enterprise Server, 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://`.

40.6.1 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 40.1 *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`.

40.6.2 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 43.4.1, "Configuring the Firewall with YaST" (page 829).

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 750) for the general virtual host configuration.

To get started, copy the template to `/etc/apache2/vhosts.d/`*`mySSL-host`*`.conf` and edit it. Adjusting the values for the following directives should be sufficient:

- `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.

40.7 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.

40.7.1 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

40.7.2 DocumentRoot Permissions

By default in SUSE Linux Enterprise Server, 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`.

40.7.3 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 753) for details). In doing so, ensure that no critical files, such as password or system configuration files, can be read from the outside.

40.7.4 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 adminis-

trator 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 SUSE Linux Enterprise Server 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.

40.7.5 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 SUSE Linux Enterprise Server, `.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).

40.8 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 40.3, “Starting and Stopping Apache” (page 761)) 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 40.2.2, “Configuring Apache with YaST” (page 755)). 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.

40.9 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.*.

40.9.1 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.

40.9.2 Apache Modules

More information about external Apache modules from Section 40.4.5, “External Modules” (page 769) is available at the following locations:

mod-apparmor
http://en.opensuse.org/AppArmor

mod_perl
http://perl.apache.org/

mod_php5
http://www.php.net/manual/en/install.unix.apache2.php

mod_python
http://www.modpython.org/

40.9.3 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/

40.9.4 Miscellaneous Sources

If you experience difficulties specific to Apache in SUSE Linux Enterprise Server, take a look at the Technical Information Search at http://www.novell.com/support. The history of Apache is provided at http://httpd.apache.org/ABOUT_APACHE.html. This page also explains why the server is called Apache.

41 The Proxy Server Squid

Squid is a widely-used proxy cache for Linux and UNIX platforms. This means that it stores requested Internet objects, such as data on a Web or FTP server, on a machine that is closer to the requesting workstation than the server. It may be set up in multiple hierarchies to assure optimal response times and low bandwidth usage, even in modes that are transparent for the end user. Additional software like squidGuard may be used to filter Web contents.

Squid acts as a proxy cache. It redirects object requests from clients (in this case, from Web browsers) to the server. When the requested objects arrive from the server, it delivers the objects to the client and keeps a copy of them in the hard disk cache. One of the advantages of caching is that several clients requesting the same object can be served from the hard disk cache. This enables clients to receive the data much faster than from the Internet. This procedure also reduces the network traffic.

Along with the actual caching, Squid offers a wide range of features such as distributing the load over intercommunicating hierarchies of proxy servers, defining strict access control lists for all clients accessing the proxy, allowing or denying access to specific Web pages with the help of other applications, and generating statistics about frequently-visited Web pages for the assessment of the users' surfing habits. Squid is not a generic proxy. It normally proxies only HTTP connections. It does also support the protocols FTP, Gopher, SSL, and WAIS, but it does not support other Internet protocols, such as Real Audio, news, or video conferencing. Because Squid only supports the UDP protocol to provide communication between different caches, many other multimedia programs are not supported.

41.1 Some Facts about Proxy Caches

As a proxy cache, Squid can be used in several ways. When combined with a firewall, it can help with security. Multiple proxies can be used together. It can also determine what types of objects should be cached and for how long.

41.1.1 Squid and Security

It is possible to use Squid together with a firewall to secure internal networks from the outside using a proxy cache. The firewall denies all clients access to external services except Squid. All Web connections must be established by the proxy. With this configuration, Squid completely controls Web access.

If the firewall configuration includes a DMZ, the proxy should operate within this zone. Section 41.5, "Configuring a Transparent Proxy" (page 797) describes how to implement a *transparent* proxy. This simplifies the configuration of the clients, because in this case they do not need any information about the proxy.

41.1.2 Multiple Caches

Several instances of Squid can be configured to exchange objects between them. This reduces the total system load and increases the chances of finding an object already existing in the local network. It is also possible to configure cache hierarchies, so a cache is able to forward object requests to sibling caches or to a parent cache—causing it to get objects from another cache in the local network or directly from the source.

Choosing the appropriate topology for the cache hierarchy is very important, because it is not desirable to increase the overall traffic on the network. For a very large network, it would make sense to configure a proxy server for every subnetwork and connect them to a parent proxy, which in turn is connected to the proxy cache of the ISP.

All this communication is handled by ICP (Internet cache protocol) running on top of the UDP protocol. Data transfers between caches are handled using HTTP (hypertext transmission protocol) based on TCP.

To find the most appropriate server from which to get the objects, one cache sends an ICP request to all sibling proxies. These answer the requests via ICP responses with a

HIT code if the object was detected or a MISS if it was not. If multiple HIT responses were found, the proxy server decides from which server to download, depending on factors such as which cache sent the fastest answer or which one is closer. If no satisfactory responses are received, the request is sent to the parent cache.

TIP

To avoid duplication of objects in different caches in the network, other ICP protocols are used, such as CARP (cache array routing protocol) or HTCP (hypertext cache protocol). The more objects maintained in the network, the greater the possibility of finding the desired one.

41.1.3 Caching Internet Objects

Not all objects available in the network are static. There are a lot of dynamically generated CGI pages, visitor counters, and encrypted SSL content documents. Objects like this are not cached because they change each time they are accessed.

The question remains as to how long all the other objects stored in the cache should stay there. To determine this, all objects in the cache are assigned one of various possible states. Web and proxy servers find out the status of an object by adding headers to these objects, such as “Last modified” or “Expires” and the corresponding date. Other headers specifying that objects must not be cached are used as well.

Objects in the cache are normally replaced, due to a lack of free hard disk space, using algorithms such as LRU (last recently used). Basically this means that the proxy expunges the objects that have not been requested for the longest time.

41.2 System Requirements

The most important thing is to determine the maximum network load the system must bear. It is, therefore, important to pay more attention to the load peaks, because these might be more than four times the day's average. When in doubt, it would be better to overestimate the system's requirements, because having Squid working close to the limit of its capabilities could lead to a severe loss in the quality of the service. The following sections point to the system factors in order of significance.

41.2.1 Hard Disks

Speed plays an important role in the caching process, so this factor deserves special attention. For hard disks, this parameter is described as *random seek time*, measured in milliseconds. Because the data blocks that Squid reads from or writes to the hard disk tend to be rather small, the seek time of the hard disk is more important than its data throughput. For the purposes of a proxy, hard disks with high rotation speeds are probably the better choice, because they allow the read-write head to be positioned in the required spot more quickly. One possibility to speed up the system is to use a number of disks concurrently or to employ striping RAID arrays.

41.2.2 Size of the Disk Cache

In a small cache, the probability of a HIT (finding the requested object already located there) is small, because the cache is easily filled and the less requested objects are replaced by newer ones. If, for example, one GB is available for the cache and the users only surf ten MB per day, it would take more than one hundred days to fill the cache.

The easiest way to determine the needed cache size is to consider the maximum transfer rate of the connection. With a 1 Mbit/s connection, the maximum transfer rate is 125 KB/s. If all this traffic ends up in the cache, in one hour it would add up to 450 MB and, assuming that all this traffic is generated in only eight working hours, it would reach 3.6 GB in one day. Because the connection is normally not used to its upper volume limit, it can be assumed that the total data volume handled by the cache is approximately 2 GB. This is why 2 GB of disk space is required in the example for Squid to keep one day's worth of browsed data cached.

41.2.3 RAM

The amount of memory (RAM) required by Squid directly correlates to the number of objects in the cache. Squid also stores cache object references and frequently requested objects in the main memory to speed up retrieval of this data. Random access memory is much faster than a hard disk.

In addition to that, there is other data that Squid needs to keep in memory, such as a table with all the IP addresses handled, an exact domain name cache, the most frequently requested objects, access control lists, buffers, and more.

It is very important to have sufficient memory for the Squid process, because system performance is dramatically reduced if it must be swapped to disk. The cachemgr.cgi tool can be used for the cache memory management. This tool is introduced in Section 41.6, “cachemgr.cgi” (page 800). Sites with huge network traffic should consider using an AMD64 or Intel 64 system with more than 4 GB of memory.

41.2.4 CPU

Squid is not a program that requires intensive CPU usage. The load of the processor is only increased while the contents of the cache are loaded or checked. Using a multiprocessor machine does not increase the performance of the system. To increase efficiency, it is better to buy faster disks or add more memory.

41.3 Starting Squid

Squid is already preconfigured in SUSE® Linux Enterprise Server, you can start it right after the installation. To ensure a smooth start-up, the network should be configured in a way that at least one name server and the Internet can be reached. Problems can arise if a dial-up connection is used with a dynamic DNS configuration. In this case, at least the name server should be entered, because Squid does not start if it does not detect a DNS server in `/etc/resolv.conf`.

41.3.1 Commands for Starting and Stopping Squid

To start Squid, enter `rcsquid start` at the command line as `root`. In the initial start-up, the directory structure of the cache must first be defined in `/var/cache/squid`. This is done automatically by the start script `/etc/init.d/squid` and can take a few seconds or even minutes. If `done` appears to the right in green, Squid has been successfully loaded. To test the functionality of Squid on the local system, enter `localhost` as the proxy and `3128` as the port in the browser.

To allow users from the local system and other systems to access Squid and the Internet, change the entry in the configuration files `/etc/squid/squid.conf` from `http_access deny all` to `http_access allow all`. However, in doing

so, consider that Squid is made completely accessible to anyone by this action. Therefore, define ACLs that control access to the proxy. More information about this is available in Section 41.4.2, “Options for Access Controls” (page 794).

After modifying the configuration file `/etc/squid/squid.conf`, Squid must reload the configuration file. Do this with `rcsquid reload`. Alternatively, completely restart Squid with `rcsquid restart`.

The command `rcsquid status` can be used to check if the proxy is running. The command `rcsquid stop` causes Squid to shut down. This can take a while, because Squid waits up to half a minute (`shutdown_lifetime` option in `/etc/squid/squid.conf`) before dropping the connections to the clients and writing its data to the disk.

WARNING: Terminating Squid

Terminating Squid with `kill` or `killall` can damage the cache. To be able to restart Squid, the damaged cache must be deleted.

If Squid dies after a short period of time even though it was started successfully, check whether there is a faulty name server entry or whether the `/etc/resolv.conf` file is missing. Squid logs the cause of a start-up failure in the file `/var/log/squid/cache.log`. If Squid should be loaded automatically when the system boots, use the YaST runlevel editor to activate Squid for the desired runlevels. See Section 8.5.12, “System Services (Runlevel)” (page 162).

An uninstall of Squid does not remove the cache hierarchy or the log files. To remove these, delete the `/var/cache/squid` directory manually.

41.3.2 Local DNS Server

Setting up a local DNS server makes sense even if it does not manage its own domain. It then simply acts as a caching-only name server and is also able to resolve DNS requests via the root name servers without requiring any special configuration (see Section 33.3, “Starting the Name Server BIND” (page 626)). How this can be done depends on whether you chose dynamic DNS during the configuration of the Internet connection.

Dynamic DNS
: Normally, with dynamic DNS, the DNS server is set by the provider during the establishment of the Internet connection and the local file `/etc/resolv.conf` is adjusted automatically. This behavior is controlled in the file `/etc/sysconfig/network/config` with the sysconfig variable `MODIFY_RESOLV_CONF_DYNAMICALLY`, which is set to `"yes"`. Set this variable to `"no"` with the YaST sysconfig editor (see Section 20.3.1, “Changing the System Configuration Using the YaST sysconfig Editor” (page 402)). Then enter the local DNS server in the file `/etc/resolv.conf` with the IP address `127.0.0.1` for `localhost`. This way Squid can always find the local name server when it starts.

  To make the provider's name server accessible, enter it in the configuration file `/etc/named.conf` under `forwarders` along with its IP address. With dynamic DNS, this can be achieved automatically during connection establishment by setting the sysconfig variable `MODIFY_NAMED_CONF_DYNAMICALLY` to `YES`.

Static DNS
: With static DNS, no automatic DNS adjustments take place while establishing a connection, so there is no need to change any sysconfig variables. You must, however, enter the local DNS server in the file `/etc/resolv.conf` as described above. Additionally, the providers static name server must be entered manually in the file `/etc/named.conf` under `forwarders` along with its IP address.

TIP: DNS and Firewall

If you have a firewall running, make sure DNS requests can pass it.

41.4 The Configuration File /etc/squid/squid.conf

All Squid proxy server settings are made in the `/etc/squid/squid.conf` file. To start Squid for the first time, no changes are necessary in this file, but external clients are initially denied access. The proxy is available for `localhost`. The default port is `3128`. The preinstalled configuration file `/etc/squid/squid.conf` provides detailed information about the options and many examples. Nearly all entries begin

with # (the lines are commented) and the relevant specifications can be found at the end of the line. The given values almost always correlate with the default values, so removing the comment signs without changing any of the parameters actually has little effect in most cases. If possible, leave the sample as it is and insert the options along with the modified parameters in the line below. This way, the default values may easily be recovered and compared with the changes.

TIP: Adapting the Configuration File after an Update

If you have updated from an earlier Squid version, it is recommended to edit the new `/etc/squid/squid.conf` and only apply the changes made in the previous file. If you try to use the old `squid.conf`, risk that the configuration no longer works, because options are sometimes modified and new changes added.

41.4.1 General Configuration Options (Selection)

http_port 3128
: This is the port on which Squid listens for client requests. The default port is `3128`, but `8080` is also common. If desired, specify several port numbers separated by blank spaces.

cache_peer *`hostname type proxy-port icp-port`*
: Here, enter a parent proxy, for example, if you want to use the proxy of your ISP. As *`hostname`*, enter the name and IP address of the proxy to use and, as *`type`*, enter `parent`. For *`proxy-port`*, enter the port number that is also given by the operator of the parent for use in the browser, usually `8080`. Set the *`icp-port`* to `7` or `0` if the ICP port of the parent is not known and its use is irrelevant to the provider. In addition, `default` and `no-query` may be specified after the port numbers to prohibit the use of the ICP protocol. Squid then behaves like a normal browser as far as the provider's proxy is concerned.

cache_mem 8 MB
: This entry defines the amount of memory Squid can use for very popular replies. The default is `8 MB`. This does not specify the memory usage of Squid and may be exceeded.

cache_dir ufs /var/cache/squid/ 100 16 256

The entry *cache_dir* defines the directory where all the objects are stored on disk. The numbers at the end indicate the maximum disk space in MB to use and the number of directories in the first and second level. The `ufs` parameter should be left alone. The default is 100 MB occupied disk space in the `/var/cache/squid` directory and creation of 16 subdirectories inside it, each containing 256 more subdirectories. When specifying the disk space to use, leave sufficient reserve disk space. Values from a minimum of 50% to a maximum of 80% of the available disk space make the most sense here. The last two numbers for the directories should only be increased with caution, because too many directories can also lead to performance problems. If you have several disks that share the cache, enter several *cache_dir* lines.

cache_access_log /var/log/squid/access.log , cache_log /var/log/squid/cache.log , cache_store_log /var/log/squid/store.log

These three entries specify the paths where Squid logs all its actions. Normally, nothing is changed here. If Squid is experiencing a heavy usage burden, it might make sense to distribute the cache and the log files over several disks.

emulate_httpd_log off

If the entry is set to *on*, obtain readable log files. Some evaluation programs cannot interpret this, however.

client_netmask 255.255.255.255

With this entry, mask IP addresses of clients in the log files. The last digit of the IP address is set to zero if you enter `255.255.255.0` here. You may protect the privacy of your clients that way.

ftp_user Squid@

With this, set the password Squid should use for the anonymous FTP login. It can make sense to specify a valid e-mail address here, because some FTP servers check these for validity.

cache_mgr webmaster

An e-mail address to which Squid sends a message if it unexpectedly crashes. The default is *webmaster*.

logfile_rotate 0

If you run `squid -k rotate`, `Squid` can rotate secured log files. The files are numbered in this process and, after reaching the specified value, the oldest file is

overwritten. The default value is `0` because archiving and deleting log files in SUSE Linux Enterprise Server is carried out by a cron job set in the configuration file `/etc/logrotate/squid`.

append_domain <domain>
: With *append_domain*, specify which domain to append automatically when none is given. Usually, your own domain is entered here, so entering *www* in the browser accesses your own Web server.

forwarded_for on
: If you set the entry to *off*, Squid removes the IP address and the system name of the client from HTTP requests. Otherwise it adds a line to the header like

```
X-Forwarded-For: 192.168.0.1
```

negative_ttl 5 minutes; negative_dns_ttl 5 minutes
: Normally, you do not need to change these values. If you have a dial-up connection, however, the Internet may, at times, not be accessible. Squid makes a note of the failed requests then refuses to issue new ones, although the Internet connection has been reestablished. In a case such as this, change the *minutes* to *seconds* then, after clicking *Reload* in the browser, the dial-up process should be reengaged after a few seconds.

never_direct allow *`acl_name`*
: To prevent Squid from taking requests directly from the Internet, use the above command to force connection to another proxy. This must have previously been entered in *cache_peer*. If `all` is specified as the *`acl_name`*, force all requests to be forwarded directly to the *parent*. This might be necessary, for example, if you are using a provider that strictly stipulates the use of its proxies or denies its firewall direct Internet access.

41.4.2 Options for Access Controls

Squid provides a detailed system for controlling the access to the proxy. By implementing ACLs, it can be configured easily and comprehensively. This involves lists with rules that are processed sequentially. ACLs must be defined before they can be used. Some default ACLs, such as *all* and *localhost*, already exist. However, the mere definition of an ACL does not mean that it is actually applied. This only happens in conjunction with *http_access* rules.

acl <acl_name> <type> <data>

An ACL requires at least three specifications to define it. The name *<acl_name>* can be chosen arbitrarily. For *<type>*, select from a variety of different options, which can be found in the *ACCESS CONTROLS* section in the `/etc/squid/squid.conf` file. The specification for *<data>* depends on the individual ACL type and can also be read from a file, for example, via hostnames, IP addresses, or URLs. The following are some simple examples:

```
acl mysurfers srcdomain .my-domain.com
acl teachers src 192.168.1.0/255.255.255.0
acl students src 192.168.7.0-192.168.9.0/255.255.255.0
acl lunch time MTWHF 12:00-15:00
```

http_access allow <acl_name>

http_access defines who is allowed to use the proxy and who can access what on the Internet. For this, ACLs must be given. *localhost* and *all* have already been defined above, which can deny or allow access via *deny* or *allow*. A list containing any number of *http_access* entries can be created, processed from top to bottom, and, depending on which occurs first, access is allowed or denied to the respective URL. The last entry should always be *http_access deny all*. In the following example, the *localhost* has free access to everything while all other hosts are denied access completely.

```
http_access allow localhost
http_access deny all
```

In another example using these rules, the group `teachers` always has access to the Internet. The group `students` only gets access Monday to Friday during lunch time.

```
http_access deny localhost
http_access allow teachers
http_access allow students lunch time
http_access deny all
```

The list with the *http_access* entries should only be entered, for the sake of readability, at the designated position in the `/etc/squid/squid.conf` file. That is, between the text

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR
# CLIENTS
```

and the last

```
http_access deny all
```

redirect_program /usr/bin/squidGuard
: With this option, specify a redirector such as squidGuard, which allows blocking unwanted URLs. Internet access can be individually controlled for various user groups with the help of proxy authentication and the appropriate ACLs. squidGuard is a separate package that can be installed and configured.

auth_param basic program /usr/sbin/pam_auth
: If users must be authenticated on the proxy, set a corresponding program, such as pam_auth. When accessing pam_auth for the first time, the user sees a login window in which to enter the username and password. In addition, an ACL is still required, so only clients with a valid login can use the Internet:

```
acl password proxy_auth REQUIRED

http_access allow password
http_access deny all
```

The *REQUIRED* after *proxy_auth* can be replaced with a list of permitted usernames or with the path to such a list.

ident_lookup_access allow <acl_name>
: With this, have an ident request run for all ACL-defined clients to find each user's identity. If you apply *all* to the *<acl_name>*, this is valid for all clients. Also, an ident daemon must be running on all clients. For Linux, install the pidentd package for this purpose. For Microsoft Windows, free software is available for download from the Internet. To ensure that only clients with a successful ident lookup are permitted, define a corresponding ACL here:

```
acl identhosts ident REQUIRED

http_access allow identhosts
http_access deny all
```

Here, too, replace *REQUIRED* with a list of permitted usernames. Using *ident* can slow down the access time quite a bit, because ident lookups are repeated for each request.

41.5 Configuring a Transparent Proxy

The usual way of working with proxy servers is the following: the Web browser sends requests to a certain port in the proxy server and the proxy provides these required objects, whether they are in its cache or not. When working in a network, several situations may arise:

- For security reasons, it is recommended that all clients use a proxy to surf the Internet.
- All clients must use a proxy, regardless of whether they are aware of it.
- The proxy in a network is moved, but the existing clients should retain their old configuration.

In all these cases, a transparent proxy may be used. The principle is very easy: the proxy intercepts and answers the requests of the Web browser, so the Web browser receives the requested pages without knowing from where they are coming. As the name indicates, the entire process is done transparently.

41.5.1 Configuration Options in /etc/squid/squid.conf

The options to activate in the `/etc/squid/squid.conf` file to get the transparent proxy up and running are:

- `httpd_accel_host` virtual
- `httpd_accel_port` 80

  The port number where the actual HTTP server is located
- `httpd_accel_with_proxy` on
- `httpd_accel_uses_host_header` on

41.5.2 Firewall Configuration with SuSEfirewall2

Now redirect all incoming requests via the firewall with help of a port forwarding rule to the Squid port. To do this, use the enclosed tool SuSEfirewall2, described in Section 43.4.1, “Configuring the Firewall with YaST” (page 829). Its configuration file can be found in `/etc/sysconfig/SuSEfirewall2`. The configuration file consists of well-documented entries. To set a transparent proxy, you must configure several firewall options:

- Device pointing to the Internet: `FW_DEV_EXT`="eth1"
- Device pointing to the network: `FW_DEV_INT`="eth0"

Define ports and services (see `/etc/services`) on the firewall that are accessed from untrusted (external) networks such as the Internet. In this example, only Web services are offered to the outside:

```
FW_SERVICES_EXT_TCP="www"
```

Define ports or services (see `/etc/services`) on the firewall that are accessed from the secure (internal) network, both via TCP and UDP:

```
FW_SERVICES_INT_TCP="domain www 3128"
FW_SERVICES_INT_UDP="domain"
```

This allows accessing Web services and Squid (whose default port is `3128`). The service “domain” stands for DNS (domain name service). This service is commonly used. Otherwise, simply take it out of the above entries and set the following option to `no`:

```
FW_SERVICE_DNS="yes"
```

The most important option is option number `15`:

Example 41.1 *Firewall Configuration: Option 15*

```
# 15.)
# Which accesses to services should be redirected to a local port
# on the firewall machine?
#
# This can be used to force all internal users to surf via your
# Squid proxy, or transparently redirect incoming Web traffic to
# a secure Web server.
#
# Choice: leave empty or use the following explained syntax of
# redirecting rules, separated with spaces.
# A redirecting rule consists of 1) source IP/net,
# 2) destination IP/net, 3) original destination port and
# 4) local port to redirect the traffic to, separated by a colon,
# e.g. "10.0.0.0/8,0/0,80,3128 0/0,172.20.1.1,80,8080"
```

The comments above show the syntax to follow. First, enter the IP address and the netmask of the internal networks accessing the proxy firewall. Second, enter the IP address and the netmask to which these clients send their requests. In the case of Web browsers, specify the networks `0/0`, a wild card that means "to everywhere." After that, enter the original port to which these requests are sent and, finally, the port to which all these requests are redirected. Because Squid supports protocols other than HTTP, redirect requests from other ports to the proxy, such as FTP (port 21), HTTPS, or SSL (port 443). In this example, Web services (port `80`) are redirected to the proxy port (port `3128`). If there are more networks or services to add, they must be separated by a blank space in the respective entry.

```
FW_REDIRECT="192.168.0.0/16,0/0,tcp,80,3128 192.168.0.0/16,0/0,tcp,21,3128"
FW_REDIRECT="192.168.0.0/16,0/0,udp,80,3128 192.168.0.0/16,0/0,udp,21,3128"
```

To start the firewall and the new configuration with it, change an entry in the `/etc/sysconfig/SuSEfirewall2` file. The entry `START_FW` must be set to `"yes"`.

Start Squid as shown in Section 41.3, "Starting Squid" (page 789). To check if everything is working properly, check the Squid logs in `/var/log/squid/access.log`. To verify that all ports are correctly configured, perform a port scan on the machine from any computer outside your network. Only the Web services (port 80) should be open. To scan the ports with nmap, the command syntax is `nmap -O IP_address`.

41.6 cachemgr.cgi

The cache manager (cachemgr.cgi) is a CGI utility for displaying statistics about the memory usage of a running Squid process. It is also a more convenient way to manage the cache and view statistics without logging the server.

41.6.1 Setup

First, a running Web server on your system is required. Configure Apache as described in Chapter 40, *The Apache HTTP Server* (page 745). To check if Apache is already running, as `root` enter the command `rcapache status`. If a message like this appears:

```
Checking for service httpd: OK
Server uptime: 1 day 18 hours 29 minutes 39 seconds
```

Apache is running on the machine. Otherwise, enter `rcapache start` to start Apache with the SUSE Linux Enterprise Server default settings. The last step to set it up is to copy the file `cachemgr.cgi` to the Apache directory `cgi-bin`:

```
cp /usr/share/doc/packages/squid/scripts/cachemgr.cgi /srv/www/cgi-bin/
```

41.6.2 Cache Manager ACLs in /etc/squid/squid.conf

There are some default settings in the original file required for the cache manager. First, two ACLs are defined then http_access options use these ACLs to grant access from the CGI script to Squid. The first ACL is the most important, because the cache manager tries to communicate with Squid over the cache_object protocol.

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
```

The following rules give Apache the access rights to Squid:

```
http_access allow manager localhost
http_access deny manager
```

These rules assume that the Web server and Squid are running on the same machine. If the communication between the cache manager and Squid originates at the Web server on another computer, include an extra ACL as in Example 41.2, “Access Rules” (page 801).

Example 41.2 *Access Rules*

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl webserver src 192.168.1.7/255.255.255.255 # webserver IP
```

Then add the rules in Example 41.3, “Access Rules” (page 801) to permit access from the Web server.

Example 41.3 *Access Rules*

```
http_access allow manager localhost
http_access allow manager webserver
http_access deny manager
```

Configure a password for the manager for access to more options, like closing the cache remotely or viewing more information about the cache. For this, configure the entry `cachemgr_passwd` with a password for the manager and the list of options to view. This list appears as a part of the entry comments in `/etc/squid/squid.conf`.

Restart Squid every time the configuration file is changed. Do this easily with `rcsquid reload`.

41.6.3 Viewing the Statistics

Go to the corresponding Web site—`http://webserver.example.org/cgi-bin/cachemgr.cgi`. Press *continue* and browse through the different statistics. More details for each entry shown by the cache manager is in the Squid FAQ at `http://www.squid-cache.org/Doc/FAQ/FAQ-9.html`.

41.7 squidGuard

This section is not intended to explain an extensive configuration of squidGuard, only to introduce it and give some advice for using it. For more in-depth configuration issues, refer to the squidGuard Web site at http://www.squidguard.org.

squidGuard is a free (GPL), flexible, and fast filter, redirector, and access controller plug-in for Squid. It lets you define multiple access rules with different restrictions for different user groups on a Squid cache. squidGuard uses Squid's standard redirector interface. squidGuard can do the following:

- Limit the Web access for some users to a list of accepted or well-known Web servers or URLs.
- Block access to some listed or blacklisted Web servers or URLs for some users.
- Block access to URLs matching a list of regular expressions or words for some users.
- Redirect blocked URLs to an “intelligent” CGI-based information page.
- Redirect unregistered users to a registration form.
- Redirect banners to an empty GIF.
- Use different access rules based on time of day, day of the week, date, etc.
- Use different rules for different user groups.

squidGuard and Squid cannot be used to:

- Edit, filter, or censor text inside documents.
- Edit, filter, or censor HTML-embedded script languages, such as JavaScript or VBscript.

Before it can be used, install `squidGuard`. Provide a minimal configuration file as `/etc/squidguard.conf`. Find configuration examples in `http://www.squidguard.org/config/`. Experiment later with more complicated configuration settings.

Next, create a dummy “access denied” page or a more or less complex CGI page to redirect Squid if the client requests a blacklisted Web site. Using Apache is strongly recommended.

Now, configure Squid to use squidGuard. Use the following entry in the `/etc/squid/squid.conf` file:

```
redirect_program /usr/bin/squidGuard
```

Another option called `redirect_children` configures the number of “redirect” (in this case squidGuard) processes running on the machine. squidGuard is fast enough to handle many requests: on a 500 MHz Pentium with 5,900 domains and 7,880 URLs (totaling 13,780), 100,000 requests can be processed within 10 seconds. Therefore, it is not recommended to set more than four processes, because the allocation of these processes would consume an excessive amount of memory

```
redirect_children 4
```

Last, have Squid load the new configuration by running `rcsquid reload`. Now, test your settings with a browser.

41.8 Cache Report Generation with Calamaris

Calamaris is a Perl script used to generate reports of cache activity in ASCII or HTML format. It works with native Squid access log files. The Calamaris home page is located at http://Calamaris.Cord.de/. The program is quite easy to use.

Log in as `root` then enter `cat access.log.files | calamaris` *options* `> reportfile`. It is important when piping more than one log file that the log files are chronologically ordered with older files first. These are some options of the program:

-a
: output all available reports

-w
: output as HTML report

-l
include a message or logo in report header

More information about the various options can be found in the program's manual page with `man calamaris`.

A typical example is:

```
cat access.log.2 access.log.1 access.log | calamaris -a -w \
> /usr/local/httpd/htdocs/Squid/squidreport.html
```

This puts the report in the directory of the Web server. Apache is required to view the reports.

Another powerful cache report generator tool is SARG (Squid Analysis Report Generator). More information about this is available at: http://sarg.sourceforge.net/.

41.9 For More Information

Visit the home page of Squid at http://www.squid-cache.org/. Here, find the “Squid User Guide” and a very extensive collection of FAQs on Squid.

Following the installation, a small HOWTO about transparent proxies is available in `howtoenh` as `/usr/share/doc/howto/en/txt/TransparentProxy.gz`. In addition, mailing lists are available for Squid at squid-users@squid-cache.org. The archive for this is located at http://www.squid-cache.org/mail-archive/squid-users/.

Part V. Security

42 Managing X.509 Certification

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.

42.1 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.

42.1.1 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.

42.1.2 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 42.1 shows the fields of a basic X.509 certificate in version 3.

Table 42.1 *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”

42.1.3 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 42.2, “X.509 Certificate Revocation List (CRL)” (page 810) shows the basic parts of a X.509 CRL.

Table 42.2 *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

42.1.4 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 36, *LDAP—A Directory Service* (page 667). Chapter 40, *The Apache HTTP Server* (page 745) contains information about the HTTP server.

42.1.5 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.

42.2 YaST Modules for CA Management

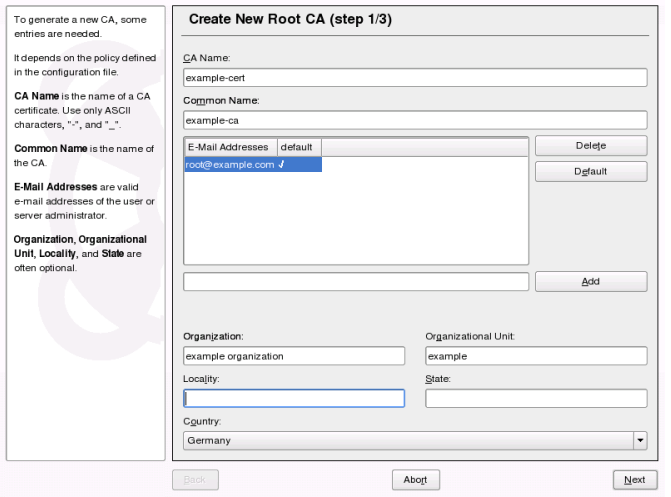
YaST provides two modules for basic CA management. The primary management tasks with these modules are explained here.

42.2.1 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 42.1, “YaST CA Module—Basic Data for a Root CA” (page 812). The text fields have the following meanings:

Figure 42.1 *YaST CA Module—Basic Data for a Root CA*



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 42.4, “YaST CA Module—Extended Settings” (page 818)). 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.

42.2.2 Creating or Revoking a Sub-CA

A sub-CA is created in exactly the same way as a root CA. Do the following:

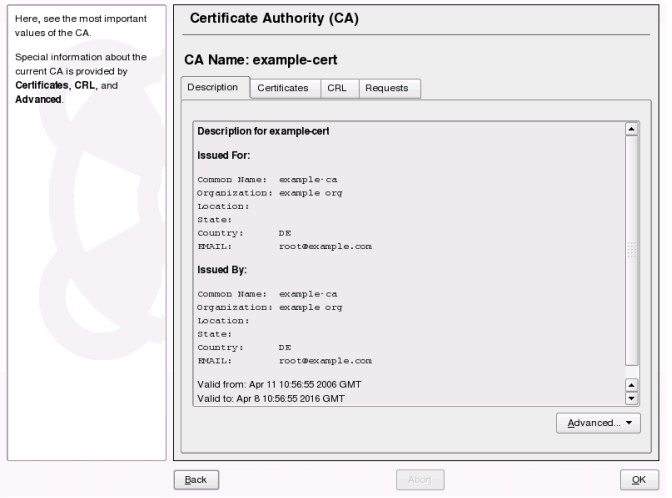
1. Start YaST and open the CA module.
2. Select the required CA and click *Enter CA*.

   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 42.2).

Figure 42.2 *YaST CA Module—Using a CA*



4 Click *Advanced* and select *Create SubCA*. This opens the same dialog as for creating a root CA.

5 Proceed as described in Section 42.2.1, “Creating a Root CA” (page 812).

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 42.2.5, “Creating CRLs ” (page 819).

7 Finish with *Ok*

42.2.3 Creating or Revoking User Certificates

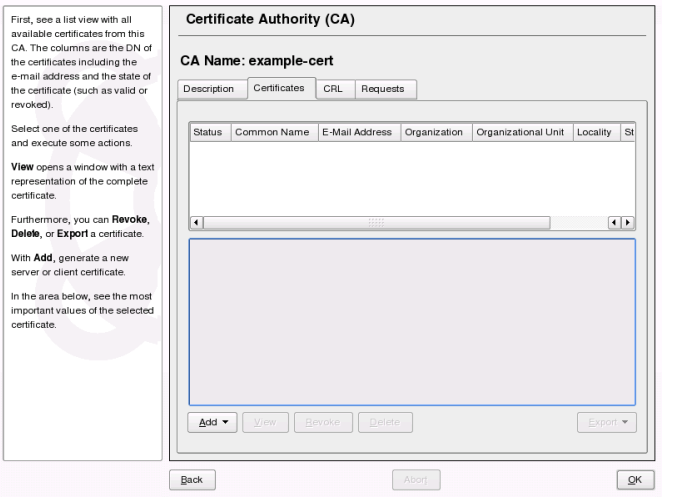
Creating client and server certificates is very similar to the one for creating CAs in Section 42.2.1, “Creating a Root CA” (page 812). The same principles apply here. In certificates intended for e-mail signature, the e-mail address of the sender (the private key owner) should be contained in the certificate 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 42.3, “Certificates of a CA” (page 816)).

Figure 42.3 *Certificates of a CA*



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 42.2.2, “Creating or Revoking a Sub-CA” (page 814).)
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 42.2.5, “Creating CRLs ” (page 819) explains how to create CRLs. Revoked certificates can be completely removed after publication in a CRL with *Delete*.

42.2.4 Changing Default Values

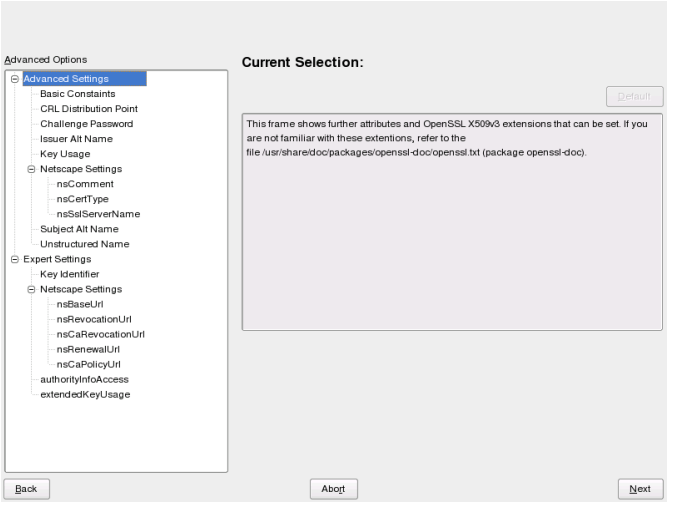
The previous sections explained how to create sub-CAs, client certificates, and server certificates. Special settings are used in the extensions of the X.509 certificate. These settings have been given rational defaults for every certificate type and do not normally need to be changed. However, it may 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 42.2.2, “Creating or Revoking a Sub-CA” (page 814).

3 Click *Advanced* > *Edit Defaults*.

4 Choose the type the settings to change. The dialog for changing the defaults, shown in Figure 42.4, “YaST CA Module—Extended Settings” (page 818), then opens.

Figure 42.4 *YaST CA Module—Extended Settings*



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.

42.2.5 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 42.2.2, "Creating or Revoking a Sub-CA" (page 814) (for sub-CAs) and Section 42.2.3, "Creating or Revoking User Certificates" (page 815) (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 42.2.2, "Creating or Revoking a Sub-CA" (page 814).
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.

42.2.6 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 42.3, “Passwords during LDAP Export” (page 820)).

Table 42.3 *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 42.2.2, “Creating or Revoking a Sub-CA” (page 814). 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, create a new CRL and click *Export*. The dialog that opens displays the export parameters. You can export the CRL for this CA either once or in periodical time intervals. Activate the export by selecting *Export to LDAP* and enter the respective LDAP data. To do this at regular intervals, select the *Repeated recreation and export* radio button and change the interval, if appropriate.

42.2.7 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 as with LDAP, described in Section 42.2.6, "Exporting CA Objects to LDAP " (page 819), 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 clicking *OK*.

For CRLs click *Export*, select *Export to file*, choose the export format (PEM or DER) and enter the path. Proceed with *Ok* to save it to the respective location.

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.

42.2.8 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*.

43 Masquerading and Firewalls

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.

43.1 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.

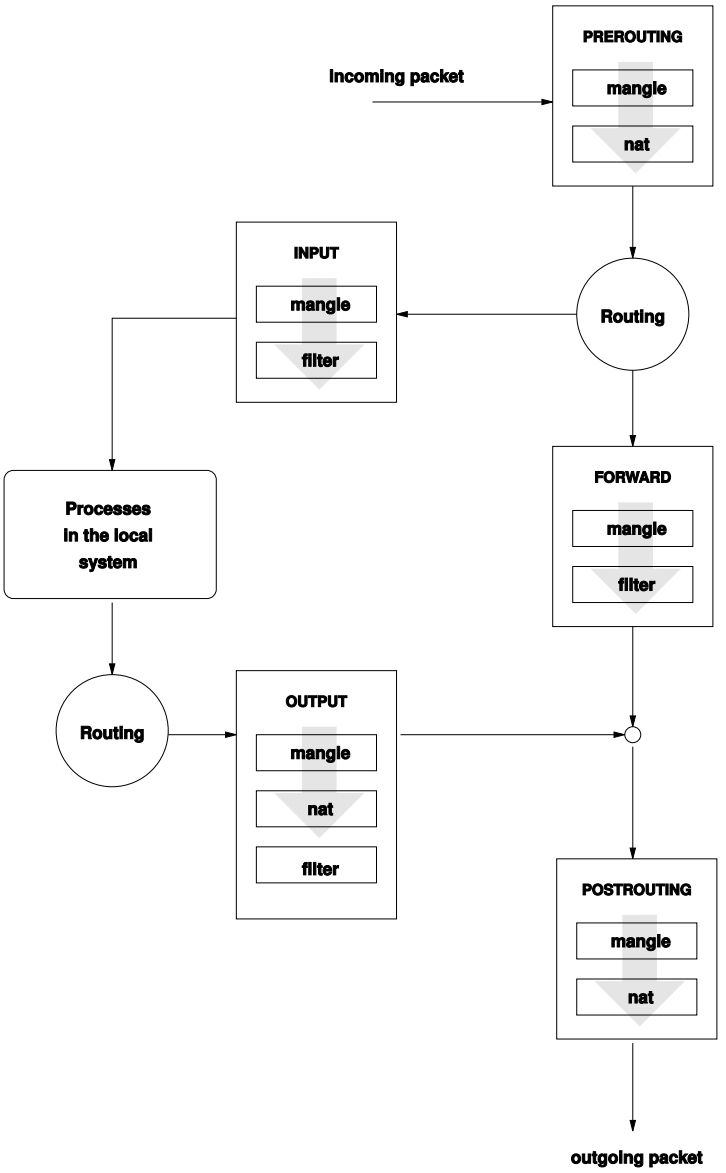
The Linux kernel maintains three tables, each for a particular category of functions of the packet filter:

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).

Figure 43.1 *iptables: A Packet's Possible Paths*



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 43.1, “iptables: A Packet's Possible Paths” (page 825) 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.

43.2 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 30.1.2, “Netmasks and Routing” (page 551)) 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.

43.3 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 SUSE Linux Enterprise. 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
```

43.4 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 30.1.2, "Netmasks and Routing" (page 551)), 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 43.4.1, "Configuring the Firewall with YaST" (page 829)). 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`.

43.4.1 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*.

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.

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.

43.4.2 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. Use port range (e.g., `500:510`) whenever appropriate. 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 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.

43.5 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

44

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 SUSE Linux Enterprise is OpenSSH.

44.1 The OpenSSH Package

SUSE Linux Enterprise installs the package OpenSSH by default. The programs ssh, scp, and sftp are then available as alternatives to telnet, rlogin, rsh, rcp, and ftp. In the default configuration, system access of a SUSE Linux Enterprise system is only possible with the OpenSSH utilities and only if the firewall permits access.

44.2 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 sun with the command `ssh sun`. The host then prompts for the password on sun.

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 sun` or `ssh augustine@sun`.

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 sun and create a directory with the name `tmp`. The program output is displayed on the local terminal of the host earth.

```
ssh otherplanet "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 sun.

44.3 scp—Secure Copy

scp copies files to a remote machine. It is a secure and encrypted substitute for rcp. For example, `scp MyLetter.tex sun:` copies the file `MyLetter.tex` from the host earth to the host sun. If the username on earth is different than the username on sun, specify the latter using the `username@host` format. There is no `-l` option for this command.

After the correct password is entered, scp starts the data transfer and shows a growing row of asterisks to simulate a progress bar. In addition, 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/ sun:backup/` copies the entire contents of the directory `src` including all subdirectories to the `backup` directory on the host sun. 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.

44.4 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.

44.5 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 consist 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. In 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`.

44.6 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`.

44.7 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:sun:25 earth
```

With this command, any connection directed to earth port 25 (SMTP) is redirected to the SMTP port on sun 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 earth can be forwarded to the POP3 port of sun with this command:

```
ssh -L 110:sun:110 earth
```

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`.

Network Authentication—Kerberos

45

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.

45.1 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*.

45.2 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.

45.2.1 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.

45.2.2 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.

45.2.3 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.

45.2.4 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.

45.2.5 Compatibility to Windows 2000

Windows 2000 contains a Microsoft implementation of Kerberos 5. Because SUSE Linux Enterprise® uses the MIT implementation of Kerberos 5, find useful information and guidance in the MIT documentation. See Section 45.4, “For More Information” (page 847).

45.3 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`

45.4 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

46

This section covers the installation of the MIT Kerberos implementation as well as some aspects of administration. This section assumes you are familiar with the basic concepts of Kerberos (see also Chapter 45, *Network Authentication—Kerberos* (page 841)).

46.1 Choosing the Kerberos Realms

The domain of a Kerberos installation is called a realm and is identified by a name, such as `FOOBAR.COM` or simply `ACCOUNTING`. Kerberos is case-sensitive, so `foobar.com` is actually a different realm than `FOOBAR.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.FOOBAR.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 `FOOBAR.COM`, have two "subrealms" named `DEVELOPMENT` and `ACCOUNTING` underneath it, and expect the two subordinate realms to somehow inherit principals from `FOOBAR.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.

46.2 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`, `shadow`, and `gshadow` 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.

46.3 Clock 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 32.1, “Configuring an NTP Client with YaST” (page 609).

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 46.5.3, “Adjusting the Clock Skew” (page 857).

46.4 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. See Section 46.4.1, “Installing the RPMs” (page 852) for details.

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.

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.

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.

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.

7 **Create a Principal for Yourself**

46.4.1 Installing the RPMs

Before you can start, install the Kerberos software. On the KDC, install the packages `krb5`, `krb5-server`, and `krb5-client`.

46.4.2 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.

46.4.3 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.

46.4.4 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`.

46.5 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.

46.5.1 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.

46.5.2 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"
```

46.5.3 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:

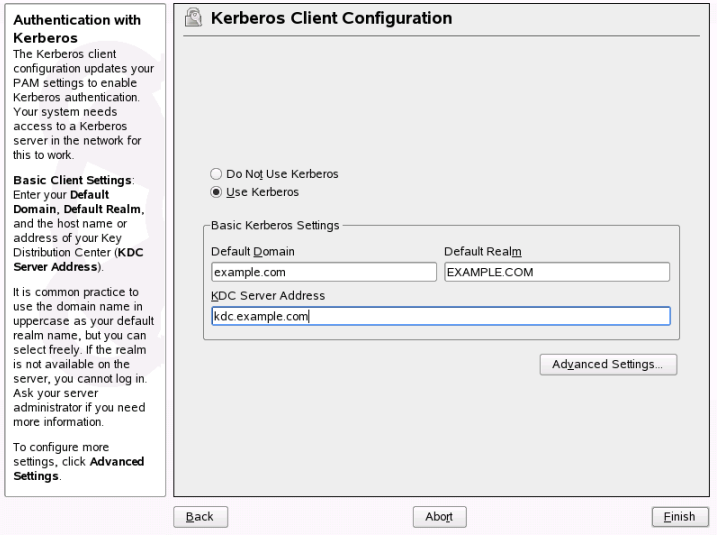
```
[libdefaults]
        clockskew = 120
```

46.6 Configuring a Kerberos Client with YaST

As an alternative to the manual configuration described above, use YaST to configure a Kerberos client. Proceed as follows:

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 46.1 *YaST: Basic Configuration of a Kerberos Client*



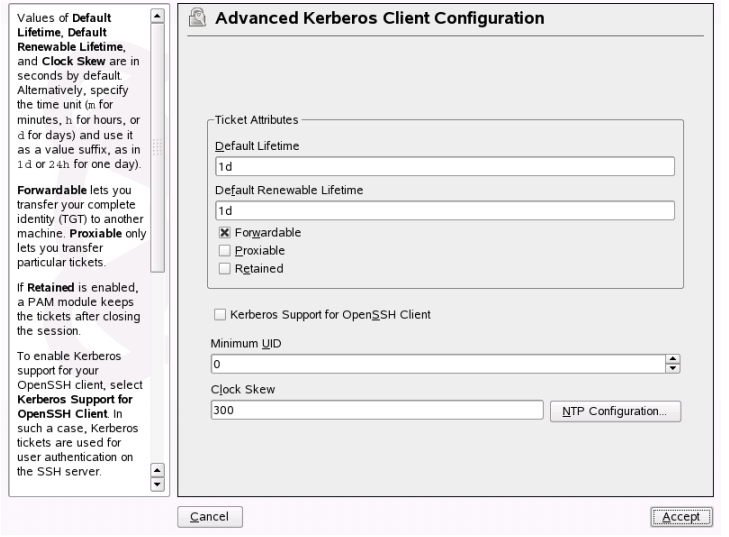
To configure ticket-related options in the *Advanced Settings* dialog, choose from the following options:

- 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 32.1, “Configuring an NTP Client with YaST” (page 609). After finishing the configuration, YaST performs all the necessary changes and the Kerberos client is ready for use.

Figure 46.2 *YaST: Advanced Configuration of a Kerberos Client*



46.7 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.

46.7.1 Using kadmin for Remote Administration

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`.

46.8 Creating Kerberos Host Principals

In addition to making sure every machine on your network knows which Kerberos realm it is in and what KDC to contact, create a *host principal* for it. So far, only user credentials have been discussed. However, Kerberos-compatible services usually need to authenticate themselves to the client user, too. Therefore, special host principals must be present in the Kerberos database for each host in the realm.

The naming convention for host principals is `host/<hostname>@<REALM>`, where `hostname` is the host's fully qualified hostname. Host principals are similar to user principals, but have significant differences. The main difference between a user principal and a host 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 host 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 principal for `test.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/test.example.com
WARNING: no policy specified for host/test.example.com@EXAMPLE.COM;
defaulting
to no policy
Principal "host/test.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/test.example.com
Entry for principal host/test.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/test.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`.

46.9 Enabling PAM Support for Kerberos

SUSE Linux Enterprise® 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.

The `pam_unix2` module also supports Kerberos authentication and password update. To enable Kerberos support in `pam_unix2`, edit the file `/etc/security/pam_unix2.conf` so it contains the following lines:

```
auth:       use_krb5 nullok
account:    use_krb5
password:   use_krb5 nullok
session:    none
```

After that, all programs evaluating the entries in this file use Kerberos for user authentication. For a user that does not have a Kerberos principal, `pam_unix2` falls back on the normal password authentication mechanism. For those users who have a principal, it should now be possible to change their Kerberos passwords transparently using the `passwd` command.

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.

46.10 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.

46.11 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`.

46.11.1 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.

46.11.2 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.

47 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 47.1.1, “Creating an Encrypted Partition during Installation” (page 871) and Section 47.1.2, “Creating an Encrypted Partition on a Running System” (page 872) for details. This option can also be used for removable media, such as external hard disks, as described in Section 47.1.4, “Encrypting the Content of Removable Media” (page 873).

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 47.1.3, “Creating an Encrypted File as a Container” (page 872).

Encrypting Home Directories
: With SUSE Linux Enterprise, 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 47.2, “Using Encrypted Home Directories” (page 873) for more information.

Encrypting Single ASCII Text Files
: If you only have a small number of ASCII text 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 47.3, “Using vi to Encrypt Single ASCII Text Files” (page 874) 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.

47.1 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 include an encrypted partition, by default. Add it manually in the partitioning dialog.

47.1.1 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* check box.
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 47.1.2, "Creating an Encrypted Partition on a Running System" (page 872) and be aware that this action destroys all data on the existing partition to encrypt.

47.1.2 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 47.1.1, "Creating an Encrypted Partition during Installation" (page 871).

47.1.3 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 or change 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.

47.1.4 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 YaST, the KDE and GNOME desktops automatically recognize the encrypted partition and prompt for the password when the device is detected. If you plug in a FAT formatted removable device while running KDE or GNOME, the desktop user entering the 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.

47.2 Using Encrypted Home Directories

To protect data in home directories against theft and hard disk removal, use the YaST user management module to enable encryption of home directories. You can create encrypted home directories for new or existing users. To encrypt or decrypt home directories of already existing users, you need to know their login password. See for instructions.

Encrypted home partitions are created within a file container as described in Section 47.1.3, “Creating an Encrypted File as a Container” (page 872). Two files are created under `/home` for each encrypted home directory:

`LOGIN``.img`
: The image holding the directory

LOGIN.key
: The image key, protected with the user's login password.

On login the home directory automatically gets decrypted. Internally, it 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 have to add this module to the respective configuration file in /etc/pam.d/. For more information see also Chapter 27, *Authentication with PAM* (page 497) and the man page of pam_mount.

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 47.1.1, “Creating an Encrypted Partition during Installation” (page 871) or Section 47.1.3, “Creating an Encrypted File as a Container” (page 872).

47.3 Using vi to Encrypt Single ASCII Text 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

48

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*.

48.1 Installing Novell AppArmor

Novell AppArmor is installed and running by default on any installation of SUSE Linux Enterprise® 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`

48.2 Enabling and Disabling Novell AppArmor

Novell AppArmor is configured to run by default on any fresh installation of SUSE Linux Enterprise. 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*.

48.3 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 48.3.1, “Choosing the Applications to Profile” (page 878).
2. Build the needed profiles as roughly outlined in Section 48.3.2, “Building and Modifying Profiles” (page 879). 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 48.3.3, “Configuring Novell AppArmor Event Notification and Reports” (page 882).
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 48.3.4, “Updating Your Profiles” (page 883).

48.3.1 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 48.1 *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*).

48.3.2 Building and Modifying Profiles

Novell AppArmor on SUSE Linux Enterprise 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 48.3.1, "Choosing the Applications to Profile" (page 878) 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 880) 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 880) and Step 3 (page 880). 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 880). 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*).

48.3.3 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*).

48.3.4 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*).

49 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.

49.1 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.

49.1.1 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.

49.1.2 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.

49.1.3 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 21, *The Boot Loader* (page 405)). 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.

49.1.4 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 SUSE Linux Enterprise 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 SUSE Linux Enterprise 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 SUSE Linux Enterprise'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`).

49.1.5 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 49.1.4, “File Permissions” (page 888)).

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 49.1.4, “File Permissions” (page 888)).

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.

49.1.6 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. SUSE Linux Enterprise'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.

49.1.7 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.

49.1.8 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. Read more about X Window System security mechanisms in the man page of Xsecurity (`man Xsecurity`).

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 44, *SSH: Secure Network Operations* (page 835).

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.

49.1.9 Buffer Overflows and Format String Bugs

As discussed in Section 49.1.5, “Buffer Overflows and Format String Bugs” (page 889), 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 (SUSE Linux Enterprise 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.

49.1.10 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.

49.1.11 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.

49.1.12 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.

49.1.13 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.

49.2 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 a mailing list to which you can subscribe by following the link `http://en.opensuse.org/Communicate/Mailinglists`. The list `opensuse-security-announce@opensuse.org` is a first-hand source of information regarding updated packages and includes members of SUSE's security team among its active contributors.

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 SUSE Linux Enterprise. 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.

49.3 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`.

Part VI. Troubleshooting

50 Help and Documentation

SUSE Linux Enterprise® comes with various sources of information and documentation. The SUSE Help Center provides central access to the most important documentation resources on your system in searchable form. These resources include online help for installed applications, manual pages, info pages, databases on hardware and software topics, and all manuals delivered with your product.

50.1 Using the SUSE Help Center

When you start the SUSE Help Center for the first time from the main menu (*SuSE Help Center*) or with the command `susehelp` in the shell, a window as shown in Figure 50.1, “The Main Window of the SUSE Help Center” (page 902) is displayed. The dialog window consists of three main areas:

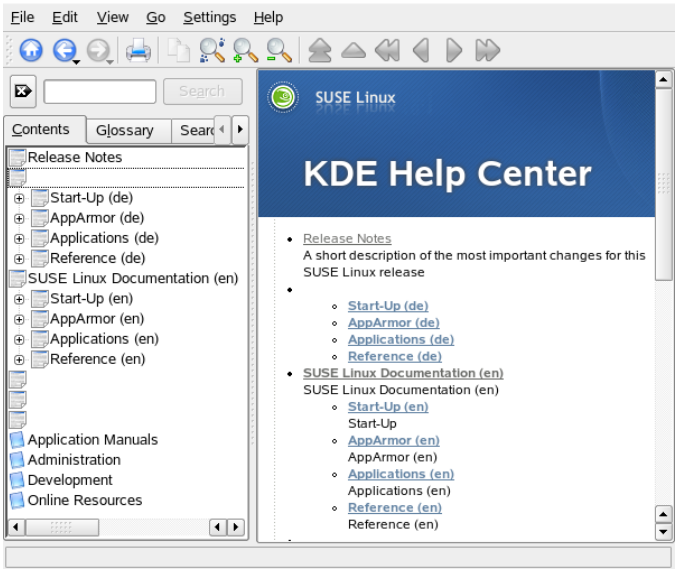
Menu Bar and Toolbar
: The menu bar provides the main editing, navigation, and configuration options. *File* contains the option for printing the currently displayed content. Under *Edit*, access the search function. *Go* contains all navigation possibilities: *Table of Contents* (home page of the Help Center), *Back*, *Forward*, and *Last Search Result*. With *Settings* > *Build Search Index*, generate a search index for all selected information sources. The toolbar contains three navigation icons (forward, back, home) and a printer icon for printing the current contents.

Navigation Area with Tabs
: The navigation area in the left part of the window provides an input field for a quick search in selected information sources. Details regarding the search and the

configuration of the search function in the *Search* tab are presented in Section 50.1.2, “The Search Function” (page 903). The *Contents* tab presents a tree view of all available and currently installed information sources. Click the book icons to open and browse the individual categories.

View Window
: The view window always displays the currently selected contents, such as online manuals, search results, or Web pages.

Figure 50.1 *The Main Window of the SUSE Help Center*



NOTE: Language Selects View

The documentation available in the SUSE Help Center depends on the current language. Changing your language changes the tree view.

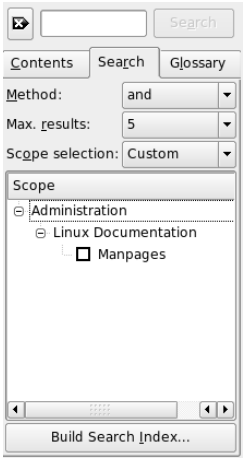
50.1.1 Contents

The SUSE Help Center provides access to useful information from various sources. It contains special documentation for SUSE Linux Enterprise (*Start-Up*, *KDE User Guide*, *GNOME User Guide*, and *Reference*), all available information sources for your workstation environment, online help for the installed programs, and help texts for other applications. Furthermore, the SUSE Help Center provides access to SUSE's online databases that cover special hardware and software issues for SUSE Linux Enterprise. All these sources can be searched comfortably once a search index has been generated.

50.1.2 The Search Function

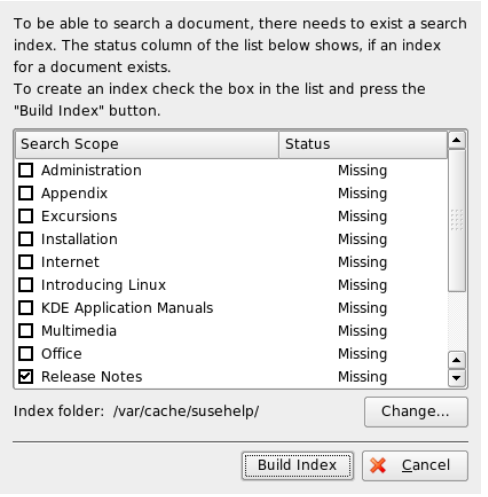
To search all installed information sources of SUSE Linux Enterprise, generate a search index and set a number of search parameters. To do this, use the *Search* tab, shown in Figure 50.2, “Configuring the Search Function” (page 903).

Figure 50.2 *Configuring the Search Function*



If no search index has been generated, the system automatically prompts you to do so when you click the *Search* tab or enter a search string then click *Search*. In the window for generating the search index, shown in Figure 50.3, “Generating a Search Index” (page 904), use the check boxes to determine the information sources to index. The index is generated when you exit the dialog with *Build Index*.

Figure 50.3 *Generating a Search Index*



To limit the search base and the hit list as precisely as possible, use the three drop-down menus to determine the number of displayed hits and the selection area of sources to search. The following options are available for determining the selection area:

Default
: A predefined selection of sources is searched.

All
: All sources are searched.

None
: No sources selected for the search.

Custom
: Determine the sources to search by activating the respective check boxes in the overview.

When you have completed the search configuration, click *Search*. The relevant items are then displayed in the view window and can easily be navigated with mouse clicks.

50.2 Man Pages

Man pages are an essential part of any Linux system. They explain the usage of a command and all available options and parameters. Man pages are sorted in categories as shown in Table 50.1, "Man Pages—Categories and Descriptions" (page 905) (taken from the man page for man itself).

Table 50.1 *Man Pages—Categories and Descriptions*

Number	Description
1	Executable programs or shell commands
2	System calls (functions provided by the kernel)
3	Library calls (functions within program libraries)
4	Special files (usually found in /dev)
5	File formats and conventions (`/etc/fstab`)
6	Games
7	Miscellaneous (including macro packages and conventions), for example, man(7), groff(7)
8	System administration commands (usually only for root)
9	Kernel routines (nonstandard)

Generally, man pages are delivered with the associated command. They can be browsed in the help center or directly in a shell. To display a man page in a shell, use the `man` command. For example, to display the man page for `ls` enter `man ls`. Each man page consists of several parts labeled *NAME*, *SYNOPSIS*, *DESCRIPTION*, *SEE ALSO*, *LICENSING*, and *AUTHOR*. There may be additional sections available depending on the type of command. With Q, exit the man page viewer.

Another possibility to display a man page is to use Konqueror. Start Konqueror and type, for example, `man:/ls`. If there are different categories for a command, Konqueror displays them as links.

50.3 Info Pages

Info pages are another important source of information on your system. Usually they are more verbose than man pages. You can browse an info page with an info viewer and display the different sections, called "nodes." Use the command `info` for this task. For example, to view the info page of `info` itself, type `info info` in the shell.

For more convenience, use the Help Center or Konqueror. Start Konqueror and type `info:/` to view the top level. To display the info page for `grep`, type `info:/grep`.

50.4 The Linux Documentation Project

The Linux Documentation Project (TLDP) is run by a team of volunteers who write Linux and Linux-related documentation (see http://www.tldp.org). The set of documents contains tutorials for beginners, but is mainly focused on experienced users and professional system administrators. TLDP publishes HOWTOs, FAQs, and guides (handbooks) under a free license.

50.4.1 HOWTOs

HOWTOs are usually a short, informal, step-by-step guide to accomplishing a specific task. It is written by experts for nonexperts in a procedural manner. For example, how to configure a DHCP server. HOWTOs can be found in the package `howto` and are installed under `/usr/share/doc/howto`

50.4.2 Frequently Asked Questions

FAQs (frequently asked questions) are a series of questions and answers. They originate from Usenet newsgroups where the purpose was to reduce continuous reposting of the same basic questions.

50.5 Wikipedia: The Free Online Encyclopedia

Wikipedia is "a multilingual encyclopedia designed to be read and edited by anyone" (see http://en.wikipedia.org). The content of Wikipedia is created by its users and is published under a free license (GFDL). Any visitors can edit articles, which gives the danger of vandalism, but this does not repel visitors. With over four hundred thousand articles, find an answer for nearly every topic.

50.6 Guides and Books

A broad range of guides and books are available for Linux topics.

50.6.1 SUSE Books

SUSE provides detailed and informative books. We provide HTML and PDF versions of our books in different languages. The PDF file is available on the DVD in the directory `docu`. For HTML, install the package `opensuse-manual_LANG` (replace *`LANG`* with your preferred language.) After the installation, find them in the SUSE Help Center.

50.6.2 Other Manuals

The SUSE help center offers additional manuals and guides for various topics or programs. More can be found at http://www.tldp.org/guides.html. They range from *Bash Guide for Beginners* to *Linux Filesystem Hierarchy* to *Linux Administrator's Security Guide*. Generally, guides are more detailed and exhaustive than a HOWTO or FAQ. They are usually written by experts for experts. Some of these books are old but still valid. Install books and guides with YaST.

50.7 Package Documentation

If you install a package in your system, a directory `/usr/share/doc/packages/`*`packagename`* is created. You can find files from the package maintainer as well as additional information from SUSE. Sometimes there are also examples, configuration files, additional scripts, or other things available. Usually you can find the following files, but they are not standard and sometimes not all files are available.

`AUTHORS`
: The list of the main developers of this package and usually their tasks.

`BUGS`
: Known bugs or malfunctions of this package. Usually also a link to a Bugzilla Web page where you can search all bugs.

`CHANGES , ChangeLog`
: Summary of changes from version to version. Usually interesting for developers, because it is very detailed.

`COPYING , LICENSE`
: Licensing information.

`FAQ`
: Question and answers collected from mailing lists or newsgroups.

`INSTALL`
: Procedures for installing this package in your system. Normally you do not need it, because you have the package installed already.

`README , README.*`
: General information such as how to use it and what you can do with this package.

`TODO`
: Things that are not implemented yet, but probably will be in the future.

`MANIFEST`
: List of files with a brief summary.

NEWS
: Description of what is new in this version.

50.8 Usenet

Created in 1979 before the rise of the Internet, Usenet is one of the oldest computer networks and still in active use. The format and transmission of Usenet articles is very similar to e-mail, but is developed for a many-to-many communication.

Usenet is organized into seven topical categories: `comp.*` for computer-related discussions, `misc.*` for miscellaneous topics, `news.*` for newsgroup-related matters, `rec.*` for recreation and entertainment, `sci.*` for science-related discussions, `soc.*` for social discussions, and `talk.*` for various controversial topics. The top levels are split in subgroups. For instance, `comp.os.linux.hardware` is a newsgroup for Linux-specific hardware issues.

Before you can post an article, have your client connect to a news server and subscribe to a specific newsgroup. News clients include Knode or Evolution. Each news server communicates to other news servers and exchanges articles with them. Not all newsgroups may be available on your news server.

Interesting newsgroups for Linux users are `comp.os.linux.apps`, `comp.os.linux.questions`, and `comp.os.linux.hardware`. If you cannot find a specific newsgroup, go to `http://www.linux.org/docs/usenetlinux.html`. Follow the general Usenet rules available online at `http://www.faqs.org/faqs/usenet/posting-rules/part1/`.

50.9 Standards and Specifications

There are various sources that provide information about standards or specifications.

`http://www.linuxbase.org`
: The Free Standards Group is an independent nonprofit organization that promotes the distribution of free software and open source software. The organization endeavors to achieve this by defining distribution-independent standards. The maintenance

of several standards, such as the important LSB (Linux Standard Base), is supervised by this organization.

http://www.w3.org
: The World Wide Web Consortium (W3C) is certainly one of the best-known standards organizations. It was founded in October 1994 by Tim Berners-Lee and concentrates on standardizing Web technologies. W3C promotes the dissemination of open, license-free, and manufacturer-independent specifications, such as HTML, XHTML, and XML. These Web standards are developed in a four-stage process in *working groups* and are presented to the public as *W3C recommendations* (REC).

http://www.oasis-open.org
: OASIS (Organization for the Advancement of Structured Information Standards) is an international consortium specializing in the development of standards for Web security, e-business, business transactions, logistics, and interoperability between various markets.

http://www.ietf.org
: The Internet Engineering Task Force (IETF) is an internationally active cooperative of researchers, network designers, suppliers, and users. It concentrates on the development of Internet architecture and the smooth operation of the Internet by means of protocols.

  Every IETF standard is published as an RFC (Request for Comments) and is available free-of-charge. There are six types of RFC: proposed standards, draft standards, Internet standards, experimental protocols, information documents, and historic standards. Only the first three (proposed, draft, and full) are IETF standards in the narrower sense (see http://www.ietf.org/rfc/rfc1796.txt).

http://www.ieee.org
: The Institute of Electrical and Electronics Engineers (IEEE) is an organization that draws up standards in the areas of information technology, telecommunication, medicine and health care, transport, and others. IEEE standards are subject to a fee.

http://www.iso.org
: The ISO Committee (International Organization for Standards) is the world's largest developer of standards and maintains a network of national standardization institutes in over 140 countries. ISO standards are subject to a fee.

http://www.din.de , http://www.din.com

The Deutsches Institut für Normung (DIN) is a registered technical and scientific association. It was founded in 1917. According to DIN, the organization is “the institution responsible for standards in Germany and represents German interests in worldwide and European standards organizations.”

The association brings together manufacturers, consumers, trade professionals, service companies, scientists and others who have an interest in the establishment of standards. The standards are subject to a fee and can be ordered using the DIN home page.

Common Problems and Their Solutions

51

This chapter offers a range of common problems that can arise with an intention of covering as many of the various types of potential problems as possible. That way, even if your precise situation is not listed here, there might be one similar enough to offer hints as to the solution.

51.1 Finding and Gathering Information

Linux logs things in a fair amount of detail. There are several places to look when you have problems with your system, most of which are standard to Linux systems in general and some of which are peculiar to SUSE Linux Enterprise systems. Most log files can also be viewed with YaST (*Miscellaneous* > *Start-Up Log*).

YaST offers the possibility to collect all system information needed by the support team. Use *Miscellaneous* > *Support Query*. Select the problem category. When all information is gathered, attach it to your support request.

The following is a list of the most commonly checked log files and what they typically contain.

Table 51.1 *Log Files*

Log File	Description
`/var/log/boot.msg`	Messages from the kernel during the boot process.
`/var/log/mail.*`	Messages from the mail system.
`/var/log/messages`	Ongoing messages from the kernel and system log daemon when running.
`/var/log/NetworkManager`	Log file from NetworkManager to collect problems with network connectivity
`/var/log/SaX.log`	Hardware messages from the SaX display and KVM system.
`/home/`*`user`*`/.xsession-errors`	Messages from the desktop applications currently running. Replace *`user`* with the actual username.
`/var/log/warn`	All messages from the kernel and system log daemon assigned WARNING level or higher.
`/var/log/wtmp`	Binary file containing user login records for the current machine session. View it with `last`.
`/var/log/Xorg.*.log`	Various start-up and runtime logs from the X Window system. It is useful for debugging failed X start-ups.
`/var/log/YaST2/`	Directory containing YaST's actions and their results.
`/var/log/samba/`	Directory containing Samba server and client log messages.

Apart from log files, your machine also supplies you with information about the running system. See Table 51.2: System Information.

Table 51.2 *System Information*

File	Description
`/proc/cpuinfo`	This displays processor information, including its type, make, model, and performance.
`/proc/dma`	This shows which DMA channels are currently being used.
`/proc/interrupts`	This shows which interrupts are in use and how many of each have been in use.
`/proc/iomem`	This displays the status of I/O (input/output) memory.
`/proc/ioports`	This shows which I/O ports are in use at the moment.
`/proc/meminfo`	This displays memory status.
`/proc/modules`	This displays the individual modules.
`/proc/mounts`	This displays devices currently mounted.
`/proc/partitions`	This shows the partitioning of all hard disks.
`/proc/version`	This displays the current version of Linux.

Linux comes with a number of tools for system analysis and monitoring. See Chapter 17, *System Monitoring Utilities* (page 327) for a selection of the most important ones used in system diagnostics.

Each scenario included in the following begins with a header describing the problem followed by a paragraph or two offering suggested solutions, available references for more detailed solutions, and cross-references to other scenarios that might be related.

51.2 Installation Problems

Installation problems are situations when a machine fails to install. It may fail entirely or it may not be able to start the graphical installer. This section highlights some of the typical problems you might run into and offers possible solutions or workarounds for this kind of situations.

51.2.1 Checking Media

If you encounter any problems using the SUSE Linux Enterprise installation media, you can check the integrity of your installation media with *Software > Media Check*. Media problems are more likely to occur with media you burn yourself. To check a SUSE Linux Enterprise CD or DVD, insert the medium into the drive and click *Start* for YaST to check the MD5 checksum of the medium. This may take several minutes. If errors are detected, do not use this medium for installation.

51.2.2 Hardware Information

Display detected hardware and technical data using *Hardware > Hardware Information*. Click any node of the tree for more information about a device. This module is especially useful, for example, when submitting a support request for which you need information about your hardware.

Save the hardware information displayed to a file by clicking *Save to File*. Select the desired directory and filename then click *Save* to create the file.

51.2.3 No Bootable CD-ROM Drive Available

If your computer does not contain a bootable CD or DVD-ROM drive or if the one you have is not supported by Linux, there are several options for installing your machine without a need for a built-in CD or DVD drive:

Booting from a Floppy Disk
: Create a boot floppy and boot from floppy disk instead of CD or DVD.

Using an External Boot Device
: If it is supported by the machine's BIOS and the installation kernel, boot for installation from external CD or DVD drives.

Network Boot via PXE
: If a machines lacks a CD or DVD drive, but provides a working ethernet connection, perform a completely network-based installation. See Section 4.1.3, “Remote Installation via VNC—PXE Boot and Wake on LAN” (page 51) and Section 4.1.6, “Remote Installation via SSH—PXE Boot and Wake on LAN” (page 55) for details.

Booting from a Floppy Disk (SYSLINUX)

On some older computers, there is no bootable CD-ROM drive available, but a floppy disk drive. To install on such a system, create boot disks and boot your system with them.

The boot disks include the loader SYSLINUX and the program linuxrc. SYSLINUX enables the selection of a kernel during the boot procedure and the specification of any parameters needed for the hardware used. The program linuxrc supports the loading of kernel modules for your hardware and subsequently starts the installation.

When booting from a boot disk, the boot procedure is initiated by the boot loader SYSLINUX (package `syslinux`). When the system is booted, SYSLINUX runs a minimum hardware detection that mainly consists of the following steps:

1. The program checks if the BIOS provides VESA 2.0–compliant framebuffer support and boots the kernel accordingly.

2. The monitor data (DDC info) is read.

3. The first block of the first hard disk (MBR) is read to map BIOS IDs to Linux device names during the boot loader configuration. The program attempts to read the block by means of the the lba32 functions of the BIOS to determine if the BIOS supports these functions.

If you keep Shift pressed when SYSLINUX starts, all these steps are skipped. For troubleshooting purposes, insert the line

```
verbose 1
```

in `syslinux.cfg` for the boot loader to display which action is currently being performed.

If the machine does not boot from the floppy disk, you may need to change the boot sequence in the BIOS to `A,C,CDROM`.

External Boot Devices

Most CD-ROM drives are supported. If problems arise when booting from the CD-ROM drive, try booting CD 2 of the CD set.

If the system does not have a CD-ROM or floppy disk, it is still possible that an external CD-ROM, connected with USB, FireWire, or SCSI, can be used to boot the system. This depends largely on the interaction of the BIOS and the hardware used. Sometimes a BIOS update may help if you encounter problems.

51.2.4 Booting from Installation Media Fails

There are two possible reasons for a machine not to boot for installation:

CD or DVD-ROM Drive Unable to Read the Boot Image
: Your CD-ROM drive might not be able to read the boot image on CD 1. In this case, use CD 2 to boot the system. CD 2 contains a conventional 2.88 MB boot image that can be read even by unsupported drives and allows you to perform the installation over the network as described in Chapter 4, *Remote Installation* (page 47).

Incorrect Boot Sequence in BIOS
: The BIOS boot sequence must have CD-ROM set as the first entry for booting. Otherwise the machine would try to boot from another medium, typically the hard disk. Guidance for changing the BIOS boot sequence can be found the documentation provided with your motherboard or in the following paragraphs.

The BIOS is the software that enables the very basic functions of a computer. Motherboard vendors provide a BIOS specifically made for their hardware. Normally, the BIOS setup can only be accessed at a specific time—when the machine is booting. During this initialization phase, the machine performs a number of diagnostic hardware tests. One of them is a memory check, indicated by a memory counter. When the counter

appears, look for a line, usually below the counter or somewhere at the bottom, mentioning the key to press to access the BIOS setup. Usually the key to press is Del, F1, or Esc. Press this key until the BIOS setup screen appears.

Procedure 51.1 *Changing the BIOS Boot Sequence*

1. Enter the BIOS using the proper key as announced by the boot routines and wait for the BIOS screen to appear.

2. To change the boot sequence in an AWARD BIOS, look for the *BIOS FEATURES SETUP* entry. Other manufacturers may have a different name for this, such as *ADVANCED CMOS SETUP*. When you have found the entry, select it and confirm with Enter.

3. In the screen that opens, look for a subentry called *BOOT SEQUENCE*. The boot sequence is often set to something like `C,A` or `A,C`. In the former case, the machine first searches the hard disk (C) then the floppy drive (A) to find a bootable medium. Change the settings by pressing PgUp or PgDown until the sequence is `A,CDROM,C`.

4. Leave the BIOS setup screen by pressing Esc. To save the changes, select *SAVE & EXIT SETUP* or press F10. To confirm that your settings should be saved, press Y.

Procedure 51.2 *Changing the Boot Sequence in a SCSI BIOS (Adaptec Host Adapter)*

1. Open the setup by pressing Ctrl + A.

2. Select *Disk Utilities*, which displays the connected hardware components.

   Make note of the SCSI ID of your CD-ROM drive.

3. Exit the menu with Esc.

4. Open *Configure Adapter Settings*. Under *Additional Options*, select *Boot Device Options* and press Enter.

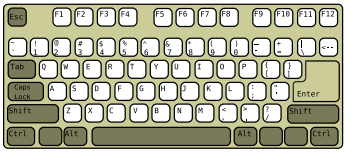
5. Enter the ID of the CD-ROM drive and press Enter again.

6. Press Esc twice to return to the start screen of the SCSI BIOS.

7 Exit this screen and confirm with *Yes* to boot the computer.

Regardless of what language and keyboard layout your final installation will be using, most BIOS configurations use the US keyboard layout as depicted in the following figure:

Figure 51.1 *US Keyboard Layout*



51.2.5 Fails to Boot

Some hardware types, mainly fairly old or very recent ones, fail to install. In many cases, this might happen because support for this type of hardware is missing from the installation kernel or due to certain functionality included in this kernel, such as ACPI, that still cause problems on some hardware.

If your system fails to install using the standard *Installation* mode from the first installation boot screen, try the following:

1 With the first CD or DVD still in the CD-ROM drive, reboot the machine with Ctrl + Alt + Del or using the hardware reset button.

2 When the boot screen appears, use the arrow keys of your keyboard to navigate to *Installation--ACPI Disabled* and press Enter to launch the boot and installation process. This option disables the support for ACPI power management techniques.

3 Proceed with the installation as described in Chapter 3, *Installation with YaST* (page 17).

If this fails, proceed as above, but choose *Installation--Safe Settings* instead. This option disables ACPI and DMA support. Most hardware should boot with this option.

If both of these options fail, use the boot options prompt to pass any additional parameters needed to support this type of hardware to the installation kernel. For more information about the parameters available as boot options, refer to the kernel documentation located in `/usr/src/linux/Documentation/kernel-parameters.txt`.

TIP: Obtaining Kernel Documentation

Install the `kernel-source` package to view the kernel documentation.

There are various other ACPI-related kernel parameters that can be entered at the boot prompt prior to booting for installation:

`acpi=off`
: This parameter disables the complete ACPI subsystem on your computer. This may be useful if your computer cannot handle ACPI at all or if you think ACPI in your computer causes trouble.

`acpi=force`
: Always enable ACPI even if your computer has an old BIOS dated before the year 2000. This parameter also enables ACPI if it is set in addition to `acpi=off`.

`acpi=noirq`
: Do not use ACPI for IRQ routing.

`acpi=ht`
: Run only enough ACPI to enable hyper-threading.

`acpi=strict`
: Be less tolerant of platforms that are not strictly ACPI specification compliant.

`pci=noacpi`
: Disable PCI IRQ routing of the new ACPI system.

`pnpacpi=off`
: This option is for seriell or parallel problems when your BIOS setup contains wrong interrupts or ports.

`notsc`
: Disable the time stamp counter. This option can be used to work around timing problems on your systems. It is a new feature, if you see regressions on your machine, especially time related or even total hangs, this option is worth a try.

`nohz=off`
: Disable the nohz feature. If your machine hangs, this option might help. Generally, you do not need it.

Once you have determined the right parameter combination, YaST automatically writes them to the boot loader configuration to make sure that the system boots properly next time.

If unexplainable errors occur when the kernel is loaded or during the installation, select *Memory Test* in the boot menu to check the memory. If *Memory Test* returns an error, it is usually a hardware error.

51.2.6 Fails to Launch Graphical Installer

After you insert the first CD or DVD into your drive and reboot your machine, the installation screen comes up, but after you select *Installation*, the graphical installer does not start.

There are several ways to deal with this situation:

- Try to select another screen resolution for the installation dialogs.
- Select *Text Mode* for installation.
- Do a remote installation via VNC using the graphical installer.

To change to another screen resolution for installation, proceed as follows:

1. Boot for installation.
2. Press F3 to open a menu from which to select a lower resolution for installation purposes.
3. Select *Installation* and proceed with the installation as described in Chapter 3, *Installation with YaST* (page 17).

To perform an installation in text mode, proceed as follows:

1. Boot for installation.
2. Press F3 and select *Text Mode*.
3. Select *Installation* and proceed with the installation as described in Chapter 3, *Installation with YaST* (page 17).

To perform a VNC installation, proceed as follows:

1. Boot for installation.
2. Enter the following text at the boot options prompt:

   ```
   vnc=1 vncpassword=some_password
   ```

   Replace `some_password` with the password to use for installation.
3. Select *Installation* then press Enter to start the installation.

   Instead of starting right into the graphical installation routine, the system continues to run in text mode then halts, displaying a message containing the IP address and port number at which the installer can be reached via a browser interface or a VNC viewer application.
4. If using a browser to access the installer, launch the browser and enter the address information provided by the installation routines on the future SUSE Linux Enterprise machine and hit Enter:

   ```
   http://ip_address_of_machine:5801
   ```

   A dialog opens in the browser window prompting you for the VNC password. Enter it and proceed with the installation as described in Chapter 3, *Installation with YaST* (page 17).

IMPORTANT

Installation via VNC works with any browser under any operating system, provided Java support is enabled.

If you use any kind of VNC viewer on your preferred operating system, enter the IP address and password when prompted to do so. A window opens, displaying the installation dialogs. Proceed with the installation as usual.

51.2.7 Only Minimalistic Boot Screen Started

You inserted the first CD or DVD into the drive, the BIOS routines are finished, but the system does not start with the graphical boot screen. Instead it launches a very minimalistic text-based interface. This might happen on any machine not providing sufficient graphics memory for rendering a graphical boot screen.

Although the text boot screen looks minimalistic, it provides nearly the same functionality as the graphical one:

Boot Options
: Unlike the graphical interface, the different boot options cannot be selected using the cursor keys of your keyboard. The boot menu of the text mode boot screen offers some keywords to enter at the boot prompt. These keywords map to the options offered in the graphical version. Enter your choice and hit `Enter` to launch the boot process.

Custom Boot Options
: After selecting a boot option, enter the appropriate keyword at the boot prompt or enter some custom boot options as described in Section 51.2.5, “Fails to Boot” (page 920). To launch the installation process, press `Enter`.

Screen Resolutions
: Use the F keys to determine the screen resolution for installation. If you need to boot in text mode, choose `F3`.

51.3 Boot Problems

Boot problems are situations when your system does not boot properly (does not boot to the expected runlevel and login screen).

51.3.1 Fails to Load the GRUB Boot Loader

If the hardware is functioning properly, it is possible that the boot loader has become corrupted and Linux cannot start on the machine. In this case, it is necessary to reinstall the boot loader. To reinstall the boot loader, proceed as follows:

1. Insert the installation media into the drive.
2. Reboot the machine.
3. Select *Installation* from the boot menu.
4. Select a language.
5. Accept the license agreement.
6. In the *Installation Mode* screen, select *Other* and set the installation mode to *Repair Installed System*.
7. Once in the YaST System Repair module, select *Expert Tools* then select *Install New Boot Loader*.
8. Restore the original settings and reinstall the boot loader.
9. Leave YaST System Repair and reboot the system.

If, for some reason, the graphical interface does not come up or you prefer to repair the system manually, refer to Section “Using the Rescue System” (page 946) for instructions.

Other reasons for the machine not booting may be BIOS-related:

BIOS Settings
: Check your BIOS for references to your hard drive. GRUB might simply not be started if the hard drive itself cannot be found with the current BIOS settings.

BIOS Boot Order
: Check whether your system's boot order includes the hard disk. If the hard disk option was not enabled, your system might install properly, but fail to boot when access to the hard disk is required.

51.3.2 No Graphical Login

If the machine comes up, but does not boot into the graphical login manager, anticipate problems either with the choice of the default runlevel or the configuration of the X Window System. To check the runlevel configuration, log in as the `root` user and check whether the machine is configured to boot into runlevel 5 (graphical desktop). A quick way to check this is to examine the contents of `/etc/inittab`, as follows:

```
nld-machine:~ # grep "id:" /etc/inittab
id:5:initdefault:
nld-machine:~ #
```

The returned line indicates that the machine's default runlevel (`initdefault`) is set to `5` and that it should boot to the graphical desktop. If the runlevel is set to any other number, use the YaST Runlevel Editor module to set it to `5`.

IMPORTANT

Do not edit the runlevel configuration manually. Otherwise SuSEconfig (run by YaST) will overwrite these changes on its next run. If you need to make manual changes here, disable future SuSEconfig changes by setting *`CHECK_INITTAB`* in `/etc/sysconfig/suseconfig` to `no`.

If the runlevel is set to `5`, you might have corruption problems with your desktop or X Windows software. Examine the log files at `/var/log/Xorg.*.log` for detailed messages from the X server as it attempted to start. If the desktop fails during start, it might log error messages to `/var/log/messages`. If these error messages hint at a configuration problem in the X server, try to fix these issues. If the graphical system still does not come up, consider reinstalling the graphical desktop.

One quick test: the `startx` command should force the X Window System to start with the configured defaults if the user is currently logged in on the console. If that does not work, it should log errors to the console. For more information about the X Window system configuration, refer to Chapter 26, *The X Window System* (page 483).

51.4 Login Problems

Login problems are those where your machine does, in fact, boot to the expected welcome screen or login prompt, but refuses to accept the username and password or accepts them but then does not behave properly (fails to start the graphic desktop, produces errors, drops to a command line, etc.).

51.4.1 Valid Username and Password Combinations Fail

This usually occurs when the system is configured to use network authentication or directory services and, for some reason, is unable to retrieve results from its configured servers. The `root` user, as the only local user, is the only user that can still log in to these machines. The following are some common reasons why a machine might appear functional but be unable to process logins correctly:

- The network is not working. For further directions on this, turn to Section 51.5, “Network Problems” (page 933).

- DNS is not working at the moment (which prevents GNOME or KDE from working and the system from making validated requests to secure servers). One indication that this is the case is that the machine takes an extremely long time to respond to any action. Find more information about this topic in Section 51.5, “Network Problems” (page 933).

- If the system is configured to use Kerberos, the system's local time might have drifted past the accepted variance with the Kerberos server time (this is typically 300 seconds). If NTP (network time protocol) is not working properly or local NTP servers are not working, Kerberos authentication ceases to function because it depends on common clock synchronization across the network.

- The system's authentication configuration is misconfigured. Check the PAM configuration files involved for any typographical errors or misordering of directives. For additional background information about PAM and the syntax of the configuration files involved, refer to Chapter 27, *Authentication with PAM* (page 497).

In all cases that do not involve external network problems, the solution is to reboot the system into single-user mode and repair the configuration before booting again into operating mode and attempting to log in again. To boot into single-user mode:

1. Reboot the system. The boot screen appears, offering a prompt.
2. Enter `1` at the boot prompt to make the system boot into single-user mode.
3. Enter the username and password for `root`.
4. Make all the necessary changes.
5. Boot into the full multiuser and network mode by entering `telinit 5` at the command line.

51.4.2 Valid Username and Password Not Accepted

This is by far the most common problem users encounter, because there are many reasons this can occur. Depending on whether you use local user management and authentication or network authentication, login failures occur for different reasons.

Local user management can fail for the following reasons:

- The user might have entered the wrong password.
- The user's home directory containing the desktop configuration files is corrupted or write protected.
- There might be problems with the X Window System authenticating this particular user, especially if the user's home directory has been used with another Linux distribution prior to installing the current one.

To locate the reason for a local login failure, proceed as follows:

1. Check whether the user remembered his password correctly before you start debugging the whole authentication mechanism. If the user might not remember his password correctly, use the YaST User Management module to change the user's password.

2 Log in as `root` and check `/var/log/messages` for error messages of the login process and of PAM.

3 Try to log in from a console (using Ctrl + Alt + F1). If this is successful, the blame cannot be put on PAM, because it is possible to authenticate this user on this machine. Try to locate any problems with the X Window System or the desktop (GNOME or KDE). For more information, refer to Section 51.4.3, “Login Successful but GNOME Desktop Fails ” (page 931) and Section 51.4.4, “Login Successful but KDE Desktop Fails” (page 932).

4 If the user's home directory has been used with another Linux distribution, remove the `Xauthority` file in the user's home. Use a console login via Ctrl + Alt + F1 and run `rm .Xauthority` as this user. This should eliminate X authentication problems for this user. Try a graphical login again.

5 If graphical login still fails, do a console login with Ctrl + Alt + F1. Try to start an X session on another display—the first one (`:0`) is already in use:

```
startx -- :1
```

This should bring up a graphical screen and your desktop. If it does not, check the log files of the X Window System (`/var/log/Xorg.`*`displaynumber`*`.log`) or the log file for your desktop applications (`.xsession-errors` in the user's home directory) for any irregularities.

6 If the desktop could not start because of corrupt configuration files, proceed with Section 51.4.3, “Login Successful but GNOME Desktop Fails ” (page 931) or Section 51.4.4, “Login Successful but KDE Desktop Fails” (page 932).

The following are some common reasons why network authentication for a particular user might fail on a specific machine:

- The user might have entered the wrong password.
- The username exists in the machine's local authentication files and is also provided by a network authentication system, causing conflicts.
- The home directory exists but is corrupt or unavailable. Perhaps it is write protected or is on a server that is inaccessible at the moment.

- The user does not have permission to log in to that particular host in the authentication system.
- The machine has changed hostnames, for whatever reason, and the user does not have permission to log in to that host.
- The machine cannot reach the authentication server or directory server that contains that user's information.
- There might be problems with the X Window System authenticating this particular user, especially if the user's home has been used with another Linux distribution prior to installing the current one.

To locate the cause of the login failures with network authentication, proceed as follows:

1. Check whether the user remembered his password correctly before you start debugging the whole authentication mechanism.
2. Determine the directory server the machine relies on for authentication and make sure that it is up and running and properly communicating with the other machines.
3. Determine that the user's username and password work on other machines to make sure that his authentication data exists and is properly distributed.
4. See if another user can log in to the misbehaving machine. If another user can log in without difficulty or if `root` can log in, log in and examine the `/var/log/messages` file. Locate the time stamps that correspond to the login attempts and determine if PAM has produced any error messages.
5. Try to log in from a console (using Ctrl + Alt + F1). If this is successful, the blame cannot be put on PAM or the directory server on which the user's home is hosted, because it is possible to authenticate this user on this machine. Try to locate any problems with the X Window System or the desktop (GNOME or KDE). For more information, refer to Section 51.4.3, “Login Successful but GNOME Desktop Fails ” (page 931) and Section 51.4.4, “Login Successful but KDE Desktop Fails” (page 932).
6. If the user's home directory has been used with another Linux distribution, remove the `Xauthority` file in the user's home. Use a console login via Ctrl + Alt + F1 and run `rm .Xauthority` as this user. This should eliminate X authentication problems for this user. Try a graphical login again.

7 If graphical login still fails, do a console login with Ctrl + Alt + F1. Try to start an X session on another display—the first one (`:0`) is already in use:

```
startx -- :1
```

This should bring up a graphical screen and your desktop. If it does not, check the log files of the X Window System (`/var/log/Xorg.`*`displaynumber`*`.log`) or the log file for your desktop applications (`.xsession-errors` in the user's home directory) for any irregularities.

8 If the desktop could not start because of corrupt configuration files, proceed with Section 51.4.3, “Login Successful but GNOME Desktop Fails ” (page 931) or Section 51.4.4, “Login Successful but KDE Desktop Fails” (page 932).

51.4.3 Login Successful but GNOME Desktop Fails

If this is true for a particular user, it is likely that the user's GNOME configuration files have become corrupted. Some symptoms might include the keyboard failing to work, the screen geometry becoming distorted, or even the screen coming up as a bare gray field. The important distinction is that if another user logs in, the machine works normally. If this is the case, it is likely that the problem can be fixed relatively quickly by simply moving the user's GNOME configuration directory to a new location, which causes GNOME to initialize a new one. Although the user is forced to reconfigure GNOME, no data is lost.

1 Switch to a text console by pressing Ctrl + Alt + F1.

2 Log in with your user name.

3 Move the user's GNOME configuration directories to a temporary location:

```
mv .gconf  .gconf-ORIG-RECOVER
mv .gnome2 .gnome2-ORIG-RECOVER
```

4 Log out.

5 Log in again, but do not run any applications.

6 Recover your individual application configuration data (including the Evolution e-mail client data) by copying the `~/.gconf-ORIG-RECOVER/apps/` directory back into the new `~/.gconf` directory as follows:

```
cp -a .gconf-ORIG-RECOVER/apps .gconf/
```

If this causes the login problems, attempt to recover only the critical application data and reconfigure the remainder of the applications.

51.4.4 Login Successful but KDE Desktop Fails

There are several reasons why a KDE desktop would not allow users to login. Corrupted cache data can cause login problems as well as corrupt KDE desktop configuration files.

Cache data is used at desktop start-up to increase performance. If this data is corrupted, start-up is slowed down or fails entirely. Removing them forces the desktop start-up routines to start from scratch. This takes more time than a normal start-up, but data is intact after this and the user can login.

To remove the cache files of the KDE desktop, issue the following command as `root`:

```
rm -rf /tmp/kde-user /tmp/socket-user
```

Replace *`user`* with the actual username. Removing these two directories just removes the corrupted cache files. No real data is harmed using this procedure.

Corrupted desktop configuration files can always be replaced with the initial configuration files. If you want to recover the user's adjustments, carefully copy them back from their temporary location after the configuration has been restored using the default configuration values.

To replace a corrupted desktop configuration with the initial configuration values, proceed as follows:

1 Switch to a text console by pressing Ctrl + Alt + F1.

2 Log in with your user name.

3 Move the KDE configuration directory and the `.skel` files to a temporary location:

```
mv .kde  .kde-ORIG-RECOVER
mv .skel .skel-ORIG-RECOVER
```

4 Log out.

5 Log in again.

6 After the desktop has started successfully, copy the user's own configurations back into place:

```
cp -a .kde-ORIG-RECOVER/share .kde/share
```

IMPORTANT

If the user's own adjustments caused the login to fail and continue to do so, repeat the procedure as described above, but do not copy the `.kde/share` directory.

51.5 Network Problems

Many problems of your system may be network-related, even though they do not seem to be at first. For example, the reason for a system not allowing users to log in might be a network problem of some kind. This section introduces a simple check list you can apply to identify the cause of any network problem encountered.

When checking the network connection of your machine, proceed as follows:

1 If using an ethernet connection, check the hardware first. Make sure that your network cable is properly plugged into your computer. The control lights next to your ethernet connector, if available, should both be active.

If the connection fails, check whether your network cable works with another machine. If it does, your network card causes the failure. If hubs or switches are included in your network setup, suspect them to be the culprits as well.

2 If using a wireless connection, check whether the wireless link can be established by other machines. If this is not the case, contact the wireless network's administrator.

3 Once you have checked your basic network connectivity, try to find out which service is not responding. Gather the address information of all network servers needed in your setup. Either look them up in the appropriate YaST module or ask your system administrator. The following list gives some of the typical network servers involved in a setup together with the symptoms of an outage.

DNS (Name Service)
: A broken or malfunctioning name service affects the network's functioning in many ways. If the local machine relies on any network servers for authentication and these servers cannot be found due to name resolution issues, users would not even be able to log in. Machines in the network managed by a broken name server would not be able to "see" each other and communicate.

NTP (Time Service)
: A malfunctioning or completely broken NTP service could affect Kerberos authentication and X server functionality.

NFS (File Service)
: If any application needed data stored in an NFS mounted directory, it would not be able to start or function properly if this service was down or misconfigured. In a worst case scenario, a user's personal desktop configuration would not come up if his home directory containing the `.gconf` or `.kde` subdirectories could not be found due to an outage of the NFS server.

Samba (File Service)
: If any application needed data stored in a directory on a Samba server, it would not be able to start or function properly if this service was down.

NIS (User Management)
: If your SUSE Linux Enterprise system relied on a NIS server to provide the user data, users would not be able to log in to this machine if the NIS service was down.

LDAP (User Management)
: If your SUSE Linux Enterprise system relied on an LDAP server to provide the user data, users would not be able to log in to this machine if the LDAP service was down.

Kerberos (Authentication)
: Authentication would not work and login to any machine would fail.

CUPS (Network Printing)
: Users would not be able to print.

4 Check whether the network servers are running and whether your network setup allows you to establish a connection:

IMPORTANT

The debugging procedure described below only applies to a simple network server/client setup that does not involve any internal routing. It assumes both server and client are members of the same subnet without the need for additional routing.

4a Use `ping` *`hostname`* (replace *`hostname`* with the hostname of the server) to check whether each one of them is up and responding to the network. If this command is successful, it tells you that the host you were looking for is up and running and that the name service for your network is configured correctly.

If ping fails with `destination host unreachable`, either your system or the desired server is not properly configured or down. Check whether your system is reachable by running `ping` *`your_hostname`* from another machine. If you can reach your machine from another machine, it is the server that is not running at all or not configured correctly.

If ping fails with `unknown host`, the name service is not configured correctly or the hostname used was incorrect. Use `ping -n` *`ipaddress`* to try to connect to this host without name service. If this is successful, check the spelling of the hostname and for a misconfigured name service in your network. For further checks on this matter, refer to Step 4b (page 936). If ping still fails, either your network card is not configured correctly or your

network hardware is faulty. Refer to Step 4c (page 937) for information about this.

4b Use `host` *`hostname`* to check whether the hostname of the server you are trying to connect to is properly translated into an IP address and vice versa. If this command returns the IP address of this host, the name service is up and running. If the `host` command fails, check all network configuration files relating to name and address resolution on your host:

`/etc/resolv.conf`
: This file is used to keep track of the name server and domain you are currently using. It can be modified manually or automatically adjusted by YaST or DHCP. Automatic adjustment is preferable. However, make sure that this file has the following structure and all network addresses and domain names are correct:

```
search fully_qualified_domain_name
nameserver ipaddress_of_nameserver
```

This file can contain more than one name server address, but at least one of them must be correct to provide name resolution to your host. If needed, adjust this file using the YaST DNS and Hostname module.

If your network connection is handled via DHCP, enable DHCP to change hostname and name service information by selecting *Change Hostname via DHCP* and *Update Name Servers and Search List via DHCP* in the YaST DNS and Hostname module.

`/etc/nsswitch.conf`
: This file tells Linux where to look for name service information. It should look like this:

```
 ...
hosts: files dns
networks: files dns
...
```

The `dns` entry is vital. It tells Linux to use an external name server. Normally, these entries are automatically made by YaST, but it never hurts to check.

If all the relevant entries on the host are correct, let your system administrator check the DNS server configuration for the correct zone infor-

mation. For detailed information about DNS, refer to Chapter 33, *The Domain Name System* (page 615). If you have made sure that the DNS configuration of your host and the DNS server are correct, proceed with checking the configuration of your network and network device.

4c If your system cannot establish a connection to a network server and you have excluded name service problems from the list of possible culprits, check the configuration of your network card.

Use the command `ifconfig` *`network_device`* (executed as `root`) to check whether this device was properly configured. Make sure that both `inet address` and `Mask` are configured correctly. An error in the IP address or a missing bit in your network mask would render your network configuration unusable. If necessary, perform this check on the server as well.

4d If the name service and network hardware are properly configured and running, but some external network connections still get long time-outs or fail entirely, use `traceroute` *`fully_qualified_domain_name`* (executed as `root`) to track the network route these requests are taking. This command lists any gateway (hop) a request from your machine passes on its way to its destination. It lists the response time of each hop and whether this hop is reachable at all. Use a combination of traceroute and ping to track down the culprit and let the administrators know.

Once you have identified the cause of your network trouble, you can resolve it yourself (if the problem is located on your machine) or let the system administrators of your network know about your findings so they can reconfigure the services or repair the necessary systems.

51.5.1 NetworkManager Problems

If you have a problem with network connectivity, narrow it down as described in Procedure 51.2, “” (page 933). If NetworkManager seems to be the culprit, proceed as follows to get logs providing hints on why NetworkManager fails:

1 Open a shell and log in as `root`.

2 Restart the NetworkManager:

```
rcnetwork restart -o nm
```

3 Open a web page, for example, http://www.opensuse.org as normal user to see, if you can connect.

4 Collect any information about the state of NetworkManager in /var/log/NetworkManager.

For more information about NetworkManager, refer to Section 30.6, “Managing Network Connections with NetworkManager” (page 584).

51.6 Data Problems

Data problems are when the machine might or might not boot properly but, in either case, it is clear that there is data corruption on the system and that the system needs to be recovered. These situations call for a backup of your critical data, enabling you to recover a system state from before your system failed. SUSE Linux Enterprise offers dedicated YaST modules for system backup and restoration as well as a rescue system that can be used to recover a corrupted system from the outside.

51.6.1 Backing Up Critical Data

System backups can be easily managed using the YaST System Backup module:

1 As root, start YaST and select *System* > *System Backup*.

2 Create a backup profile holding all details needed for the backup, filename of the archive file, scope, and type of the backup:

2a Select *Profile Management* > *Add*.

2b Enter a name for the archive.

2c Enter the path to the location of the backup if you want to keep a local backup. For your backup to be archived on a network server (via NFS), enter

the IP address or name of the server and the directory that should hold your archive.

2d Determine the archive type and click *Next*.

2e Determine the backup options to use, such as whether files not belonging to any package should be backed up and whether a list of files should be displayed prior to creating the archive. Also determine whether changed files should be identified using the time-consuming MD5 mechanism.

Use *Expert* to enter a dialog for the backup of entire hard disk areas. Currently, this option only applies to the Ext2 file system.

2f Finally, set the search constraints to exclude certain system areas from the backup area that do not need to be backed up, such as lock files or cache files. Add, edit, or delete items until your needs are met and leave with *OK*.

3 Once you have finished the profile settings, you can start the backup right away with *Create Backup* or configure automatic backup. It is also possible to create other profiles tailored for various other purposes.

To configure automatic backup for a given profile, proceed as follows:

1 Select *Automatic Backup* from the *Profile Management* menu.

2 Select *Start Backup Automatically*.

3 Determine the backup frequency. Choose *daily*, *weekly*, or *monthly*.

4 Determine the backup start time. These settings depend on the backup frequency selected.

5 Decide whether to keep old backups and how many should be kept. To receive an automatically generated status message of the backup process, check *Send Summary Mail to User root*.

6 Click *OK* to apply your settings and have the first backup start at the time specified.

51.6.2 Restoring a System Backup

Use the YaST System Restoration module to restore the system configuration from a backup. Restore the entire backup or select specific components that were corrupted and need to be reset to their old state.

1. Start *YaST > System > System Restoration*.

2. Enter the location of the backup file. This could be a local file, a network mounted file, or a file on a removable device, such as a floppy or a CD. Then click *Next*.

   The following dialog displays a summary of the archive properties, such as the filename, date of creation, type of backup, and optional comments.

3. Review the archived content by clicking *Archive Content*. Clicking *OK* returns you to the *Archive Properties* dialog.

4. *Expert Options* opens a dialog in which to fine-tune the restore process. Return to the *Archive Properties* dialog by clicking *OK*.

5. Click *Next* to open the view of packages to restore. Press *Accept* to restore all files in the archive or use the various *Select All*, *Deselect All*, and *Select Files* buttons to fine-tune your selection. Only use the *Restore RPM Database* option if the RPM database is corrupted or deleted and this file is included in the backup.

6. After you click *Accept*, the backup is restored. Click *Finish* to leave the module after the restore process is completed.

51.6.3 Recovering a Corrupted System

There are several reasons why a system could fail to come up and run properly. A corrupted file system after a system crash, corrupted configuration files, or a corrupted boot loader configuration are the most common ones.

SUSE Linux Enterprise offers two different methods to cope with this kind of situation. You can either use the YaST System Repair functionality or boot the rescue system. The following sections cover both flavors of system repair.

Using YaST System Repair

Before launching the YaST System Repair module, determine in which mode to run it to best fit your needs. Depending on the severeness and cause of your system failure and your expertise, there are three different modes to choose from:

Automatic Repair
: If your system failed due to an unknown cause and you basically do not know which part of the system is to blame for the failure, use *Automatic Repair*. An extensive automated check will be performed on all components of your installed system. For a detailed description of this procedure, refer to Section "Automatic Repair" (page 941).

Customized Repair
: If your system failed and you already know which component is to blame, you can cut the lengthy system check with *Automatic Repair* short by limiting the scope of the system analysis to those components. For example, if the system messages prior to the failure seem to indicate an error with the package database, you can limit the analysis and repair procedure to checking and restoring this aspect of your system. For a detailed description of this procedure, refer to Section "Customized Repair" (page 943).

Expert Tools
: If you already have a clear idea of what component failed and how this should be fixed, you can skip the analysis runs and directly apply the tools necessary for the repair of the respective component. For details, refer to Section "Expert Tools" (page 944).

Choose one of the repair modes as described above and proceed with the system repair as outlined in the following sections.

Automatic Repair

To start the automatic repair mode of YaST System Repair, proceed as follows:

1. Insert the first installation medium of SUSE Linux Enterprise into your CD or DVD drive.

2. Reboot the system.

3 At the boot screen, select *Installation*.

4 Select the language and click *Next*.

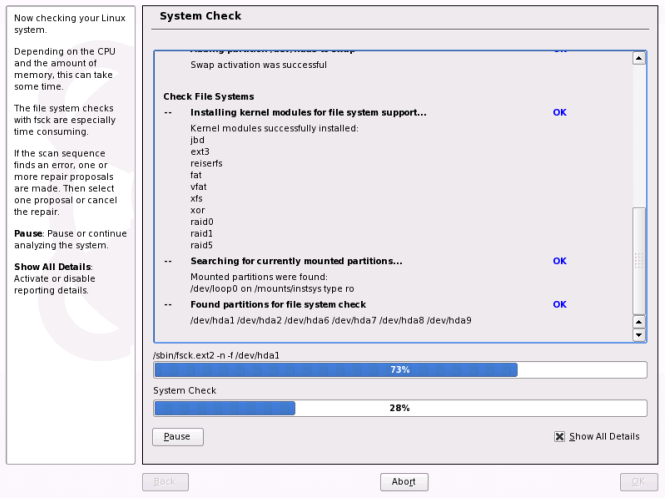
5 Confirm the license agreement and click *Next*.

6 In *System Analysis*, select *Other* > *Repair Installed System*.

7 Select *Automatic Repair*.

YaST now launches an extensive analysis of the installed system. The progress of the procedure is displayed at the bottom of the screen with two progress bars. The upper bar shows the progress of the currently running test. The lower bar shows the overall progress of the analysis. The log window in the top section tracks the currently running test and its result. See Figure 51.2, “Automatic Repair Mode” (page 942). The following main test runs are performed with every run. They contain, in turn, a number of individual subtests.

Figure 51.2 *Automatic Repair Mode*



Partition Tables of All Hard Disks
: Checks the validity and coherence of the partition tables of all detected hard disks.

Swap Partitions
: The swap partitions of the installed system are detected, tested, and offered for activation where applicable. The offer should be accepted for the sake of a higher system repair speed.

File Systems
: All detected file systems are subjected to a file system–specific check.

Entries in the File `/etc/fstab`
: The entries in the file are checked for completeness and consistency. All valid partitions are mounted.

Boot Loader Configuration
: The boot loader configuration of the installed system (GRUB or LILO) is checked for completeness and coherence. Boot and root devices are examined and the availability of the initrd modules is checked.

Package Database
: This checks whether all packages necessary for the operation of a minimal installation are present. While it is optionally possible also to analyze the base packages, this takes a long time because of their vast number.

8 Whenever an error is encountered, the procedure stops and a dialog opens outlining the details and possible solutions.

Read the screen messages carefully before accepting the proposed fix. If you decide to decline a proposed solution, your system remains unchanged.

9 After the repair process has been terminated successfully, click *OK* and *Finish* and remove the installation media. The system automatically reboots.

Customized Repair

To launch the *Customized Repair* mode and selectively check certain components of your installed system, proceed as follows:

1 Insert the first installation medium of SUSE Linux Enterprise into your CD or DVD drive.

2 Reboot the system.

3 At the boot screen, select *Installation*.

4 Select the language and click *Next*.

5 Confirm the license agreement and click *Next*.

6 In *System Analysis*, select *Other* > *Repair Installed System*.

7 Select *Customized Repair*.

Choosing *Customized Repair* shows a list of test runs that are all marked for execution at first. The total range of tests matches that of automatic repair. If you already know where no damage is present, unmark the corresponding tests. Clicking *Next* starts a narrower test procedure that probably has a significantly shorter running time.

Not all test groups can be applied individually. The analysis of the fstab entries is always bound to an examination of the file systems, including existing swap partitions. YaST automatically resolves such dependencies by selecting the smallest number of necessary test runs.

8 Whenever an error is encountered, the procedure stops and a dialog opens outlining the details and possible solutions.

Read the screen messages carefully before accepting the proposed fix. If you decide to decline a proposed solution, your system remains unchanged.

9 After the repair process has been terminated successfully, click *OK* and *Finish* and remove the installation media. The system automatically reboots.

Expert Tools

If you are knowledgeable with SUSE Linux Enterprise and already have a very clear idea of what needs to be repaired in your system, directly apply the tools skipping the system analysis.

To make use of the *Expert Tools* feature of the YaST System Repair module, proceed as follows:

1 Boot the system with the original installation medium used for the initial installation (as outlined in Chapter 3, *Installation with YaST* (page 17)).

2 In *System Analysis*, select *Other* > *Repair Installed System*.

3 Select *Expert Tools* and choose one or more repair options.

4 After the repair process has been terminated successfully, click *OK* and *Finish* and remove the installation media. The system automatically reboots.

Expert tools provides the following options to repair your faulty system:

Install New Boot Loader
: This starts the YaST boot loader configuration module. Find details in Section 21.3, "Configuring the Boot Loader with YaST" (page 416).

Start Partitioning Tool
: This starts the expert partitioning tool in YaST.

Repair File System
: This checks the file systems of your installed system. You are first offered a selection of all detected partitions and can then choose the ones to check.

Recover Lost Partitions
: It is possible to attempt to reconstruct damaged partition tables. A list of detected hard disks is presented first for selection. Clicking *OK* starts the examination. This can take a while depending on the processing power and size of the hard disk.

IMPORTANT: Reconstructing a Partition Table

The reconstruction of a partition table is tricky. YaST attempts to recognize lost partitions by analyzing the data sectors of the hard disk. The lost partitions are added to the rebuilt partition table when recognized. This is, however, not successful in all imaginable cases.

Save System Settings to Floppy
: This option saves important system files to a floppy disk. If one of these files become damaged, it can be restored from disk.

Verify Installed Software
: This checks the consistency of the package database and the availability of the most important packages. Any damaged installed packages can be reinstalled with this tool.

Using the Rescue System

SUSE Linux Enterprise contains a rescue system. The rescue system is a small Linux system that can be loaded into a RAM disk and mounted as root file system, allowing you to access your Linux partitions from the outside. Using the rescue system, you can recover or modify any important aspect of your system:

- Manipulate any type of configuration file.
- Check the file system for defects and start automatic repair processes.
- Access the installed system in a "change root" environment
- Check, modify, and reinstall the boot loader configuration
- Resize partitions using the parted command. Find more information about this tool at the Web site of GNU Parted (`http://www.gnu.org/software/parted/parted.html`).

The rescue system can be loaded from various sources and locations. The simplest option is to boot the rescue system from the original installation CD or DVD:

1. Insert the installation medium into your CD or DVD drive.
2. Reboot the system.
3. At the boot screen, choose the *Rescue System* option.
4. Enter `root` at the `Rescue:` prompt. A password is not required.

If your hardware setup does not include a CD or DVD drive, you can boot the rescue system from a network source. The following example applies to a remote boot scenario—if using another boot medium, such as a floppy disk, modify the `info` file accordingly and boot as you would for a normal installation.

1. Enter the configuration of your PXE boot setup and replace `install=`*`protocol`*`://`*`instsource`* with `rescue=`*`protocol`*`://`*`instsource`*. As with a normal installation, *`protocol`* stands for any of the supported network protocols (NFS, HTTP, FTP, etc.) and *`instsource`* for the path to your network installation source.

2 Boot the system using “Wake on LAN”, as described in Section 4.3.7, “Wake on LAN” (page 75).

3 Enter `root` at the `Rescue:` prompt. A password is not required.

Once you have entered the rescue system, you can make use of the virtual consoles that can be reached with Alt + F1 to Alt + F6.

A shell and many other useful utilities, such as the mount program, are available in the `/bin` directory. The `sbin` directory contains important file and network utilities for reviewing and repairing the file system. This directory also contains the most important binaries for system maintenance, such as fdisk, mkfs, mkswap, mount, mount, init, and shutdown, and ifconfig, ip, route, and netstat for maintaining the network. The directory `/usr/bin` contains the vi editor, find, less, and ssh.

To see the system messages, either use the command `dmesg` or view the file `/var/log/messages`.

Checking and Manipulating Configuration Files

As an example for a configuration that might be fixed using the rescue system, imagine you have a broken configuration file that prevents the system from booting properly. You can fix this using the rescue system.

To manipulate a configuration file, proceed as follows:

1 Start the rescue system using one of the methods described above.

2 To mount a root file system located under `/dev/sda6` to the rescue system, use the following command:

```
mount /dev/sda6 /mnt
```

All directories of the system are now located under `/mnt`

3 Change the directory to the mounted root file system:

```
cd /mnt
```

4 Open the problematic configuration file in the vi editor. Adjust and save the configuration.

5 Unmount the root file system from the rescue system:

```
umount /mnt
```

6 Reboot the machine.

Repairing and Checking File Systems

Generally, file systems cannot be repaired on a running system. If you encounter serious problems, you may not even be able to mount your root file system and the system boot may end with a `kernel panic`. In this case, the only way is to repair the system from the outside. It is strongly recommended to use the YaST System Repair for this task (see Section "Using YaST System Repair" (page 941) for details). However, if you need to do a manual file system check or repair, boot the rescue system. It contains the utilities to check and repair the `ext2`, `ext3`, `reiserfs`, `xfs`, `dosfs`, and `vfat` file systems.

Accessing the Installed System

If you need to access the installed system from the rescue system to, for example, modify the boot loader configuration, or to execute a hardware configuration utility, you need to do this in a "change root" environment.

To set up a "change root" environment based on the installed system, proceed as follows:

1 First mount the root partition from the installed system and the device file system:

```
mount /dev/sda6 /mnt
mount --bind /dev /mnt/dev
```

2 Now you can "change root" into the new environment:

```
chroot /mnt
```

3 Then mount `/proc` and `/sys`:

```
mount /proc
mount /sys
```

4 Finally, mount the remaining partitions from the installed system:

```
mount -a
```

5 Now you have access to the installed system. Before rebooting the system, unmount the partitions with `umount -a` and leave the "change root" environment with `exit`.

WARNING: Limitations

Although you have full access to the files and applications of the installed system, there are some limitations. The kernel that is running is the one that was booted with the rescue system. It only supports essential hardware and it is not possible to add kernel modules from the installed system unless the kernel versions are exactly the same (which is unlikely). So you cannot access a sound card, for example. It is also not possible to start a graphical user interface.

Also note that you leave the "change root" environment when you switch the console with Alt + F1 to Alt + F6.

Modifying and Reinstalling the Boot Loader

Sometimes a system cannot boot because the boot loader configuration is corrupted. The start-up routines cannot, for example, translate physical drives to the actual locations in the Linux file system without a working boot loader.

To check the boot loader configuration and reinstall the boot loader, proceed as follows:

1 Perform the necessary steps to access the installed system as described in Section "Accessing the Installed System" (page 948).

2 Check whether the following files are correctly configured according to the GRUB configuration principles outlined in Chapter 21, *The Boot Loader* (page 405).

- `/etc/grub.conf`
- `/boot/grub/device.map`
- `/boot/grub/menu.lst`

Apply fixes to the device mapping (`device.map`) or the location of the root partition and configuration files, if necessary.

3 Reinstall the boot loader using the following command sequence:

```
grub --batch < /etc/grub.conf
```

4 Unmount the partitions, log out from the "change root" environment, and reboot the system:

```
umount -a
exit
reboot
```

51.7 IBM System z: Using initrd as a Rescue System

If the kernel of the SUSE® Linux Enterprise Server for IBM System z is upgraded or modified, it is possible to reboot the system accidentally in an inconsistent state, so standard procedures of IPLing the installed system fail. This most commonly occurs if a new or updated SUSE Linux Enterprise Server kernel has been installed and the zipl program has not been run to update the IPL record. In this case, use the standard installation package as a rescue system from which the zipl program can be executed to update the IPL record.

51.7.1 IPLing the Rescue System

IMPORTANT: Making the Installation Data Available

For this method to work, the SUSE Linux Enterprise Server for IBM System z installation data must be available. For details, refer to Section "Making the Installation Data Available" (Chapter 2, *Preparing for Installation*, ↑*Architecture-Specific Information*) from *Architecture-Specific Information*. Additionally, you need the channel number of the device and the partition number within the device that contains the root file system of the SUSE Linux Enterprise Server installation.

First, IPL the SUSE Linux Enterprise Server for IBM System z installation system as described in the *Architecture-Specific Information* manual. A list of choices for the network adapter to use is then presented.

Select *Start Installation or System* then *Start Rescue System* to start the rescue system. Depending on the installation environment, you now must specify the parameters for the network adapter and the installation source. The rescue system is loaded and the following login prompt is shown at the end:

```
Skipped services in runlevel 3:  nfs nfsboot

Rescue login:
```

You can now login as `root` without a password.

51.7.2 Configuring Disks

In this state, no disks are configured. You need to configure them before you can proceed.

Procedure 51.3 *Configuring DASDs*

1. Configure DASDs with the following command:

   ```
   dasd_configure 0.0.0150 1 0
   ```

   0.0.0150 is the channel to which the DASD is connected. The `1` means activate the disk (a `0` at this place would deactivate the disk). The `0` stands for "no DIAG mode" for the disk (a `1` here would enable DAIG access to the disk).

2. Now the DASD is online (check with `cat /proc/partitions`) and can used for subsequent commands.

Procedure 51.4 *Configuring a zFCP Disk*

1. To configure a zFCP disk, it is necessary to first configure the zFCP adapter. Do this with the following command:

   ```
   zfcp_host_configure 0.0.4000 1
   ```

`0.0.4000` is the channel to which the adapter is attached and `1` stands for activate (a `0` here would deactivate the adapter).

2 After the adapter is activated, a disk can be configured. Do this with the following command:

```
zfcp_disk_configure 0.0.4000 1234567887654321 8765432100000000  1
```

`0.0.4000` is the previously-used channel ID, `1234567887654321` is the WWPN (World wide Port Number), and `8765432100000000` is the LUN (logical unit number). The `1` stands for activating the disk (a `0` here would deactivate the disk).

3 Now the zFCP disk is online (check with `cat /proc/partitions`) and can used for subsequent commands.

51.7.3 Mounting the Root Device

If all needed disks are online, you should now be able to mount the root device. Assuming that the root device is on the second partition of the DASD device (`/dev/dasda2`), the corresponding command is `mount /dev/dasda2 /mnt`.

IMPORTANT: File System Consistency

If the installed system has not been shut down properly, it may be advisable to check the file system consistency prior to mounting. This prevents any accidental loss of data. Using this example, issue the command `fsck /dev/dasda2` to ensure that the file system is in a consistent state.

By just issuing the command `mount`, it is possible to check whether the file system could be mounted correctly.

Example 51.1 *Output of the Mount Command*

```
SuSE Instsys suse:/ # mount
shmfs on /newroot type shm (rw,nr_inodes=10240)
devpts on /dev/pts type devpts (rw)
virtual-proc-filesystem on /proc type proc (rw)
/dev/dasda2 on /mnt type reiserfs (rw)
```

51.7.4 Changing to the Mounted File System

For the `zipl` command to read the configuration file from the root device of the installed system and not from the rescue system, change the root device to the installed system with the `chroot` command:

Example 51.2 *chroot to the Mounted File System*

```
SuSE Instsys suse:/ # cd /mnt
SuSE Instsys suse:/mnt # chroot /mnt
```

51.7.5 Executing zipl

Now execute `zipl` to rewrite the IPL record with the correct values:

Example 51.3 *Installing the IPL Record with zipl*

```
sh-2.05b# zipl
building bootmap : /boot/zipl/bootmap
adding Kernel Image : /boot/kernel/image located at 0x00010000
adding Ramdisk : /boot/initrd located at 0x00800000
adding Parmline : /boot/zipl/parmfile located at 0x00001000
Bootloader for ECKD type devices with z/OS compatible layout installed.
Syncing disks....
...done
```

51.7.6 Exiting the Rescue System

To exit the rescue system, first leave the shell opened by the `chroot` command with `exit`. To prevent any loss of data, flush all unwritten buffers to disk with the `sync` command. Now change to the root directory of the rescue system and unmount the root device of SUSE Linux Enterprise Server for IBM System z installation.

Example 51.4 *Unmounting the File System*

```
SuSE Instsys suse:/mnt # cd /
SuSE Instsys suse:/ # umount /mnt
```

Finally, halt the rescue system with the `halt` command. The SUSE Linux Enterprise Server system can now be IPLed as described in Section 3.13.1, “IBM System z: IPLing the Installed System” (page 34).

Index

Symbols

64-bit Linux, 383
  kernel specifications, 388
  runtime support, 384
  software development, 385

A

access permissions (see permissions)
ACLs, 303-314
  access, 305, 308
  check algorithm, 313
  default, 306, 311
  definitions, 305
  effects, 311
  handling, 306
  masks, 310
  permission bits, 307
  structure, 306
  support, 314
ACPI
  disabling, 21
add-on products, 139
Apache, 168, 745-784
  CGI scripts, 771
  configuring, 747
    files, 747
    manually, 747-754
    virtual host, 750
    YaST, 755-761
  installing, 746
  modules, 763-771
    available, 765
    building, 770
    external, 769
    installing, 764
    multiprocessing, 768
  quick start, 745
  security, 779
  Squid, 800
  SSL, 773-779
    configure Apache with SSL, 778
    creating an SSL certificate, 774
  starting, 761
  stopping, 761
  troubleshooting, 781
authentication
  Kerberos, 227
  PAM, 497-505
AutoYaST, 182
  cloning system, 45

B

backups, 145
  creating with YaST, 153
  restoring, 153
Bash, 350-361
  .bashrc, 426
  .profile, 426
  commands, 350
  features, 357
  pipes, 359
  profile, 425
  wild cards, 357
BIND, 626-637
BIOS
  boot sequence, 918
booting, 389
  boot sectors, 405, 406
  CD, from, 918
  configuring
    YaST, 416
  floppy disks, from, 917
  graphic, 422
  GRUB, 405, 406-424
  initramfs, 391

initrd, 391
log, 183
bzip2, 361

C

cards
graphics, 489
network, 565
sound, 151
cat, 370
cd, 367
CDs
booting from, 918
checking, 146, 916
chgrp, 364, 367
chmod, 363, 367
chown, 364, 367
CJK, 433
clear, 376
commands, 365-376
bzip2, 361
cat, 370
cd, 367
chgrp, 364, 367
chmod, 363, 367
chown, 364, 367
clear, 376
cp, 366
date, 373
df, 372
diff, 371
du, 373
file, 370
find, 370
fonts-config, 491
free, 373, 430
getfacl, 309
grep, 371
grub, 406
gzip, 361, 368
halt, 376
help, 352
ifconfig, 599
ip, 597
kadmin, 853
kill, 374
killall, 374
kinit, 860
ktadd, 862
ldapadd, 680
ldapdelete, 683
ldapmodify, 682
ldapsearch, 682, 866
less, 370
ln, 367
locate, 369
lp, 450
ls, 366
man, 365
mkdir, 367
mount, 371
mv, 366
nslookup, 375
passwd, 375
ping, 374, 598
ps, 374
reboot, 376
rm, 366
rmdir, 367
route, 600
rpm, 315
rpmbuild, 315
scp, 836
setfacl, 309
sftp, 837
slptool, 606
smbpasswd, 713
ssh, 836
ssh-agent, 839

ssh-keygen, 839
su, 375
tar, 360, 369
telnet, 375
top, 373
umount, 372
updatedb, 370
configuration files, 589
.bashrc, 426, 429
.emacs, 431
.profile, 426
.xsession, 839
acpi, 512
asound.conf, 153
crontab, 426
csh.cshrc, 435
dhclient.conf, 653
dhcp, 590
dhcpd.conf, 654
fstab, 159, 371
group, 212
grub.conf, 414
host.conf, 593
HOSTNAME, 596
hosts, 168, 563, 592
ifcfg-*, 590
inittab, 393, 395, 396, 432
inputrc, 433
kernel, 391
krb5.conf, 854, 855, 857, 863
krb5.keytab, 862
language, 433, 435
logrotate.conf, 427
menu.lst, 408
modprobe.d/sound, 153
named.conf, 627, 628-637, 790
network, 590
networks, 593
nscd.conf, 596
nsswitch.conf, 594, 689
openldap, 865
pam_unix2.conf, 689, 863
passwd, 212
permissions, 896
powersave, 511
powersave.conf, 235
profile, 425, 429, 435
resolv.conf, 430, 591, 627, 789
routes, 590
samba, 707
services, 707, 798
slapd.conf, 673, 867
smb.conf, 707, 717
smppd.conf, 603
smpppd-c.conf, 604
squid.conf, 789, 791, 795, 797, 800, 802
squidguard.conf, 802
sshd_config, 840, 864
ssh_config, 864
suseconfig, 404
sysconfig, 163, 402-404
termcap, 433
wireless, 590
XF86Config, 230
xorg.conf, 230, 483
Device, 488
Monitor, 489
Screen, 486
configuring, 402
cable modem, 578
DASD, 148
DNS, 167, 615
DSL, 164, 578
e-mail, 165
firewalls, 181
graphics cards, 191
groups, 178
GRUB, 406, 414
hard disk controllers, 147

hard disks
DMA, 148
hardware, 146-153
IPv6, 562
ISDN, 164, 575
languages, 163
mail servers, 166
modems, 164, 572
monitor, 191
network cards, 164
networks, 164-172, 565
manually, 586-602
NFS, 168, 169
NTP, 169
PAM, 233
power management, 162
powertweak, 162
printing, 443-447
local printers, 443
network printers, 447
routing, 170, 590
Samba, 705-711
clients, 171, 711
servers, 171
security, 172-181
software, 132-145
sound cards, 151
Squid, 791
SSH, 835
system, 129-184
system services, 169
T-DSL, 581
time zone, 163
users, 172
wireless cards, 164
ZFCP, 149
consoles
assigning, 432
graphical, 422
switching, 432
core files, 429
cp, 366
cpuspeed, 519
cron, 426
CVS, 734, 738-741

D

date, 373
deltarpm, 319
df, 372
DHCP, 167, 641-657
configuring with YaST, 642
dhcpd, 654-655
packages, 653
server, 654-655
static address assignment, 655
diff, 371
directories
changing, 367
creating, 367
deleting, 367
paths, 355
structure, 353
disks
boot, 420
DNS, 563
BIND, 626-637
configuring, 167, 615
domains, 591
forwarding, 627
logging, 631
mail exchanger, 564
name servers, 591
NIC, 564
options, 629
reverse lookup, 636
security and, 894
Squid and, 790
starting, 627

terminology, 615
top level domain, 563
troubleshooting, 627
zones
files, 633
documentation (see help)
domain name system (see DNS)
DOS
sharing files, 703
drives
mounting, 371
unmounting, 372
du, 373

E

e-mail
configuring, 165
editors
Emacs, 431-432
vi, 376
Emacs, 431-432
.emacs, 431
default.el, 431
encoding
ISO-8859-1, 435
encrypting, 869-874
creating partitions, 871
files, 872-874
files with vi, 874
partitions, 870-872
removable media, 873
YaST, with, 870
error messages
bad interpreter, 160
permission denied, 160

F

file, 370
file servers, 169
file systems, 471-481
ACLs, 303-314
changing, 158
cryptofs, 869
encrypting, 869
Ext2, 473-474
Ext3, 474-475
LFS, 479
limitations, 479
OCFS2, 289-301, 477
ReiserFS, 472-473
repairing, 948
selecting, 472
supported, 478-479
terms, 471
XFS, 476
files
archiving, 360, 369
comparing, 371
compressing, 360, 368
copying, 366
deleting, 366
encrypting, 872
finding, 428
moving, 366
paths, 355
searching contents, 371
searching for, 369, 370
synchronizing, 733-743
CVS, 734, 738-741
rsync, 734
uncompressing, 361
viewing, 358, 370
find, 370
Firefox
URL open command, 237
firewalls, 181, 823
packet filters, 823, 828
Squid and, 798
SuSEfirewall2, 823, 828

fonts, 491
  TrueType, 490
  X11 core, 491
  Xft, 492
free, 373

G

GNOME
  shell, 350
graphics
  cards
    drivers, 489
grep, 371
groups
  managing, 178
GRUB, 405-424
  boot menu, 408
  boot password, 414
  boot sectors, 406
  booting, 406
  commands, 406-415
  device names, 409
  device.map, 407, 413
  GRUB Geom Error, 423
  grub.conf, 407, 414
  limitations, 406
  Master Boot Record (MBR), 405
  menu editor, 412
  menu.lst, 407, 408
  partition names, 409
  troubleshooting, 422
  uninstalling, 420
gunzip, 361
gzip, 361, 368

H

halt, 376
hard disks
  DMA, 148
hardware
  DASD, 148
  graphics cards, 191
  hard disk controllers, 147
  information, 148, 916
  ISDN, 575
  monitor, 191
  ZFCP, 149
help, 901-904
  books, 907
  FAQs, 906
  guides, 907
  HOWTOs, 906
  info pages, 431, 906
  Linux documentation (TLDP), 906
  man pages, 365, 431, 905
  manuals, 907
  package documentation, 908
  specifications, 909
  standards, 909
  SUSE books, 907
  SUSE Help Center, 901
  Usenet, 909
  Wikipedia, 907
  X, 490
hostnames, 167

I

I18N, 433
inetd, 169
info pages, 431
init, 393
  adding scripts, 398
  inittab, 393
  scripts, 396-400
installing
  directory, into, 146
  GRUB, 406
  manually, 232

packages, 316
YaST, with, 17-45
internationalization, 433
Internet
cinternet, 604
dial-up, 602-604
DSL, 578
ISDN, 575
KInternet, 604
qinternet, 604
smpppd, 602-604
TDSL, 581
IP addresses, 551
classes, 551
dynamic assignment, 641
IPv6, 554
configuring, 562
masquerading, 826
private, 553
iSCSI, 271

J

joystick
configuring, 150

K

KDE
shell, 350
Kerberos, 841-847
administering, 849-868
authenticators, 842
clients
configuring, 854-856
clock skew, 857
clock synchronization, 851
configuring
clients, 854-856
credentials, 842
installing, 849-868
KDC, 850-854
administering, 859
nsswitch.conf, 850
starting, 854
keytab, 862
LDAP and, 865-868
master key, 852
PAM support, 863
principals, 842
creating, 853
host, 861
realms, 849
creating, 853
session key, 842
SSH configuration, 864
stash file, 852
ticket-granting service, 845
tickets, 842, 845
kernels
caches, 430
limits, 480
keyboard
Asian characters, 433
configuring, 150
layout, 432
mapping, 432
compose, 433
multikey, 433
X Keyboard Extension, 433
XKB, 433
kill, 374
killall, 374

L

L10N, 433
languages, 145, 163
laptops
power management, 507-519
LDAP, 667-701

access control, 676
ACLs, 674
adding data, 679
administering groups, 697
administering users, 697
configuring
YaST, 683
deleting data, 683
directory tree, 669
Kerberos and, 865-868
ldapadd, 679
ldapdelete, 683
ldapmodify, 681
ldapsearch, 682
modifying data, 681
searching data, 682
server configuration
manual, 673
YaST, 683
YaST
client, 688
modules, 690
templates, 690
less, 358, 370
LFS, 479
license agreement, 27
Lightweight Directory Access Protocol (see LDAP)
Linux
networks and, 547
sharing files with another OS, 703
uninstalling, 420
linuxrc
manual installation, 232
ln, 367
local APIC
disabling, 21
localization, 433
locate, 369, 428
log files, 179, 427
boot.msg, 183, 511
messages, 183, 627, 833
Squid, 790, 793, 799
logging
login attempts, 179
Logical Volume Manager (see LVM)
logrotate, 427
LPAR installation
IPL, 34
ls, 351, 366
LSB
installing packages, 315
LVM
YaST, 115

M

mail servers
configuring, 166
man pages, 365, 431
masquerading, 826
configuring with SuSEfirewall2, 828
Master Boot Record (see MBR)
MBR, 405, 406
memory
RAM, 430
mkdir, 367
modems
cable, 578
YaST, 572
more, 358
mount, 371
mouse
configuring, 150
mv, 366

N

name servers (see DNS)
NAT (see masquerading)
NetBIOS, 704

Network File System (see NFS)
Network Information Service (see NIS)
NetworkManager, 584
networks, 547
  authentication
    Kerberos, 841-847
  base network address, 553
  broadcast address, 553
  configuration files, 589-596
  configuring, 164-172, 565-581, 586-602
    IPv6, 562
  DHCP, 167, 641
  DNS, 563
  localhost, 553
  netmasks, 551
  routing, 170, 551
  SLP, 605
  TCP/IP, 547
  virtual LAN, 583
  YaST, 565
    alias, 567
    gateway, 569
    hostname, 568
    IP address, 566
    starting, 570
NFS, 719
  clients, 168, 720
  exporting, 729
  importing, 721
  mounting, 721
  servers, 169, 723
NIS, 659-666
  clients, 169, 665
  masters, 659-665
  servers, 169
  slaves, 659-665
nslookup, 375
NSS, 594
  databases, 595
NTP
  client, 169

O

OpenLDAP (see LDAP)
OpenSSH (see SSH)
OpenWBEM, 241-269
OS/2
  sharing files, 703

P

package management
  zmd, 199
packages
  compiling, 323
  compiling with build, 325
  installing, 316
  LSB, 315
  package manager, 315
  RPMs, 315
  uninstalling, 316
  verifying, 316
packet filters (see firewalls)
PAM, 497-505
  configuring, 233
partitions
  creating, 31, 155, 157
  encrypting, 871
  EVMS, 158
  fstab, 159
  LVM, 158
  parameters, 158
  partition table, 405
  RAID, 158
  reformatting, 158
  types, 156
passwd, 375
passwords
  changing, 375
paths, 355

absolute, 355
relative, 355
PCI device
drivers, 161
permissions, 361
ACLs, 303-314
changing, 363, 367
directories, 363
file permissions, 428
file systems, 362
files, 362
viewing, 363
ping, 374, 598
Pluggable Authentication Modules (see PAM)
ports
53, 629
scanning, 799
PostgreSQL
updating, 212
power management, 507-528
ACPI, 507, 511-518, 523
APM, 509-510, 523
battery monitor, 508
charge level, 524
cpufrequency, 519
cpuspeed, 519
hibernation, 508
powersave, 519
standby, 508
suspend, 508
YaST, 528
powersave, 519
configuring, 520
printing, 439
command line, 450
configuration with YaST, 443-447
local printers, 443
network printers, 447
CUPS, 450
GDI printers, 455
kprinter, 450
network, 457
Samba, 704
troubleshooting
network, 457
xpp, 450
private branch exchange, 576
processes, 373
killing, 374
overview, 374
protocols
CIFS, 703
IPv6, 554
LDAP, 667
SLP, 605
SMB, 703
proxies, 170, 785 (see Squid)
advantages, 785
caches, 785
transparent, 797
ps, 374

R

RAID
YaST, 123
reboot, 376
registering
YaST, 140
release notes, 44, 183
repairing systems, 941
rescue system, 946, 950
starting from CD, 946
starting from network source, 946
RFCs, 547
rm, 366
rmdir, 367
routing, 170, 551, 590-591
masquerading, 826

netmasks, 551
routes, 590
static, 590
RPM, 315-326
database
rebuilding, 317, 323
deltarpm, 319
dependencies, 316
patches, 317
queries, 320
rpmnew, 316
rpmorig, 316
rpmsave, 316
security, 896
SRPMS, 323
tools, 326
uninstalling, 317
updating, 316
verify, 322
verifying, 316
rpmbuild, 315
rsync, 734, 741
rug, 200-204
runlevels, 162, 393-396
changing, 395-396
editing in YaST, 400

S

Samba, 703-718
CIFS, 703
clients, 171, 704, 711-712
configuring, 705-711
installing, 705
login, 712
names, 704
permissions, 711
printers, 704
printing, 712
security, 711
server, 704
servers, 171, 705-711
shares, 704, 709
SMB, 703
starting, 705
stopping, 705
swat, 707
TCP/IP and, 703
SaX2
display device, 192
display settings, 191
dual head, 193
graphics card, 192
graphics tablet, 196
keyboard settings, 195
mouse settings, 194
multihead, 194
resolution and color depth, 193
touchscreen, 196
SCPM, 162
screen
resolution, 487
scripts
init.d, 393, 396-400, 601
boot, 398
boot.local, 398
boot.setup, 398
halt, 398
network, 601
nfsserver, 602
portmap, 602
postfix, 602
rc, 396, 398
squid, 789
xinetd, 601
ypbind, 602
ypserv, 602
mkinitrd, 391
modify_resolvconf, 430, 591
SuSEconfig, 402-404

disabling, 404
security, 885-897
attacks, 893-894
booting, 886, 888
bugs and, 889, 892
configuring, 172-181
DNS, 894
engineering, 886
firewalls, 181, 823
intrusion detection, 233
local, 887-891
network, 891-894
passwords, 887-888
permissions, 888-889
reporting problems, 897
RPM signatures, 896
Samba, 711
serial terminals, 886, 887
Squid, 786
SSH, 835-840
tcpd, 897
telnet, 835
tips and tricks, 895
viruses, 890
worms, 894
X and, 891
Service Location Protocol (see SLP)
shells, 349-380
Bash, 350
commands, 365-376
pipes, 359
wild cards, 357
SLP, 605
browser, 606
Konqueror, 606
providing services, 607
registering services, 607
slptool, 606
SMB (see Samba)
smbd, 703
soft RAID (see RAID)
software
compiling, 323
installing, 132-139
removing, 132-139
sound
configuring in YaST, 151
mixers, 232
source
compiling, 323
spm, 323
Squid, 785
access controls, 800
ACLs, 794
Apache, 800
cachemgr.cgi, 800, 801
caches, 785, 786
damaged, 790
size, 788
Calamaris, 803, 804
configuring, 791
CPU and, 789
directories, 789
DNS, 790
features, 785
firewalls and, 798
log files, 790, 793, 799
object status, 787
permissions, 789, 794
RAM and, 788
reports, 803, 804
security, 786
squidGuard, 802
starting, 789
statistics, 800, 801
stopping, 790
system requirements, 787
transparent proxies, 797, 799
troubleshooting, 790
uninstalling, 790

SSH, 835-840
authentication mechanisms, 839
daemon, 837
key pairs, 837, 839
scp, 836
sftp, 837
ssh, 836
ssh-agent, 839, 840
ssh-keygen, 839
sshd, 837
X and, 840
su, 375
support query, 913
SUSE books, 907
system
configuring, 129-184
languages, 163
limiting resource use, 429
localizing, 433
rebooting, 376
rescuing, 946
security, 179
services, 169
shutdown, 376
updating, 144

T

tar, 360, 369
TCP/IP, 547
ICMP, 548
IGMP, 548
layer model, 548
packets, 549, 550
TCP, 548
UDP, 548
telnet, 375
time zones, 163
TLDP, 906
top, 373
Tripwire
replaced by AIDE, 233

U

ulimit, 429
options, 429
umount, 372
uninstalling
GRUB, 420
Linux, 420
updatedb, 370
updating
online, 140-143
command line, 200
passwd and group, 212
patch CD, 144
problems, 212
sound mixers, 232
YaST, 213
US Keyboard Layout, 920
users
/etc/passwd, 500, 690
managing, 172

V

variables
environment, 434
VNC
administration, 170

W

whois, 564
wild cards, 370
Windows
sharing files, 703

X

X
character sets, 490

configuring, 483-490
display device, 192
display settings, 191
drivers, 489
dual head, 193
font systems, 491
fonts, 490
graphics card, 192
graphics tablet, 196
help, 490
keyboard settings, 195
mouse settings, 194
multihead, 194
resolution and color depth, 193
SaX2, 484
security, 891
SSH and, 840
touchscreen, 196
TrueType fonts, 490
virtual screen, 488
X11 core fonts, 491
xft, 490
Xft, 492
xorg.conf, 484
X Keyboard Extension (see keyboard, XKB)
X Window System (see X)
X.509 certification
certificates, 809
principles, 807
repository, 811
revocation list, 810
YaST, 807
X.Org, 483
Xft, 492
xinetd, 169
XKB (see keyboard, XKB)
xorg.conf
color depth, 487
Depth, 487
Device, 488
Display, 487
Files, 485
InputDevice, 485
Modeline, 487
modelines, 485
Modes, 485, 487
modules, 485
Monitor, 485, 487
ServerFlags, 485

Y

YaST
add-on products, 28, 139
auto login, 175
autoinstallation, 182
profiles, 182
AutoYaST, 182
backups, 145, 153
boot configuration, 416
default system, 419
security, 419
time-out, 419
boot loader
location, 418
password, 419
type, 417
CA management, 180, 812
cable modem, 578
CD creator, 182
clustering, 154
command line, 188
configuring, 129-184
Control Center, 130
DASD, 148
DHCP, 642
DMA, 148
DNS, 167
driver CDs, 184

DSL, 578
e-mail, 165
EVMS, 154
firewall, 181
graphics cards, 191
group management, 178
GRUB, 417
hard disk controllers, 147
hardware, 146-153
  information, 148, 916
Heartbeat, 154
high availability, 154
hostname, 37, 167
installation into directory, 146
installation mode, 28
installation server, 182
installation settings, 29
installation sources, 139
installation summary, 29
installing with, 17-45
ISDN, 575
joystick, 150
Kerberos client, 168
keyboard, 150
languages, 24, 130, 145, 163
LDAP, 168
  clients, 688
  servers, 683
LILO, 417
LVM, 115, 154
mail server, 166
media check, 27, 146, 916
memory test, 21
modems, 572
monitor, 191
ncurses, 185
network card, 565
network configuration, 38, 164-172
NFS clients, 168
NFS server, 169
NIS clients, 665
Novell AppArmor, 172
Novell Customer Center, 140
NTP client, 169
online update, 140-143
partitioning, 31, 155
PCI device drivers, 161
power management, 162, 528
powertweak, 162
printer configuration, 443-447
  local printers, 443
  network printers, 447
profile manager, 162
RAID, 123
registering, 140
release notes, 183
repairing systems, 941
rescue system, 21
root password, 37
routing, 170
runlevels, 400
safe settings, 21
Samba
  clients, 171, 711
  servers, 171
SCPM, 162
security, 172-181
sendmail, 165
server certificate, 180
SLP, 171
SLP browser, 606
software, 132-145
software updates, 41
sound cards, 151
starting, 18, 129
support query, 182, 913
sysconfig editor, 163, 402
system security, 179
system start-up, 18
T-DSL, 581

text mode, 185-187
time zone, 29, 163
updating, 144, 213
user management, 172
virtualization, 181
hypervisor, 181
installing, 181
X.509 certification, 807
certificates, 815
changing default values, 817
creating CRLs, 819
exporting CA objects as a file, 821
exporting CA objects to LDAP, 819
importing general server certificates, 821
root CA, 812
sub-CA, 814
ZFCP, 149
YP (see NIS)

Z

z/VM Installation
IPL, 35
ZENworks
zmd, 199
zmd, 199