SUSE LINUX Enterprise Server

INSTALLATION AND ADMINISTRATION

9. Edition 2004

Copyright ©

This publication is intellectual property of SUSE LINUX AG.

Its contents can be duplicated, either in part or in whole, provided that a copyright label is visibly located on each copy.

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

Many of the software and hardware descriptions cited in this book are registered trademarks. All trade names are subject to copyright restrictions and may be registered trade marks. SUSE LINUX AG essentially adheres to the manufacturer's spelling. Names of products and trademarks appearing in this book (with or without specific notation) are likewise subject to trademark and trade protection laws and may thus fall under copyright restrictions.

Please direct suggestions and comments to `documentation@suse.de`.

Authors: Stefan Behlert, Frank Bodammer, Stefan Dirsch, Olaf Donjak, Roman Drahtmüller, Torsten Duwe, Thorsten Dubiel, Thomas Fehr, Stefan Fent, Werner Fink, Kurt Garloff, Carsten Groß, Joachim Gleißner, Andreas Grünbacher, Franz Hassels, Andreas Jaeger, Klaus Kämpf, Hubert Mantel, Lars Marowsky-Bree, Johannes Meixner, Lars Müller, Matthias Nagorni, Anas Nashif, Siegfried Olschner, Peter Pöml, Thomas Renninger, Heiko Rommel, Marcus Schäfer, Nicolaus Schüler, Klaus Singvogel, Hendrik Vogelsang, Klaus G. Wagner, Rebecca Walter, Christian Zoz
Translators: Olaf Niepolt, Daniel Pisano
Editors: Jörg Arndt, Antje Faber, Berthold Gunreben, Roland Haidl, Jana Jaeger, Edith Parzefall, Ines Pozo, Thomas Rölz, Thomas Schraitle, Rebecca Walter
Layout: Manuela Piotrowski, Thomas Schraitle
Setting: DocBook-XML und LaTeX

This book has been printed on 100 % chlorine-free bleached paper.

Contents

I Installation 5

1 Installation with YaST 7

1.1 s390, zseries: Booting the System into the Installation Program 8
1.2 Booting into the Installation Program 8
1.2.1 Possible Problems when Starting from the CD or DVD 8
1.3 The Boot Screen 10
1.4 Language Selection 12
1.5 s390, zseries: Hard Disk Configuration 12
1.6 Installation Mode 14
1.7 Installation Suggestion 15
1.7.1 Installation Mode 15
1.7.2 Keyboard Layout 15
1.7.3 Mouse 16
1.7.4 Partitioning 17
1.7.5 Expert Partitioning with YaST 23
1.7.6 Software 29
1.7.7 Boot Configuration (Boot Loader Installation) 32
1.7.8 Time Zone 32
1.7.9 Language 33
1.7.10 Launching the Installation 34
1.7.11 s390, zseries: IPLing the Installed System 34

1.7.12 s390, zseries: Connecting to the Installed System 34
1.8 Finishing the Installation 35
1.8.1 root Password 36
1.8.2 Network Configuration 36
1.8.3 Testing the Internet Connection 38
1.8.4 Loading Software Updates 39
1.8.5 Network Services 39
1.8.6 User Authentication 41
1.8.7 Configuring the Host as a NIS Client 42
1.8.8 Creating Local User Accounts 43
1.8.9 Reading the Release Notes 44
1.9 Hardware Configuration 44
1.10 Graphical Login 46

2 YaST — Configuration 47
2.1 Starting YaST 48
2.1.1 Running YaST on a Graphical Desktop 48
2.1.2 Running from a Remote Terminal 48
2.2 The YaST Control Center 48
2.3 Software 49
2.3.1 Change Installation Source 49
2.3.2 YaST Online Update 50
2.3.3 Patch CD Update 53
2.3.4 Installing and Removing Software 53
2.3.5 System Update 61
2.3.6 SUSE Software Development Kit (SDK) 64
2.4 Hardware 65
2.4.1 CD-ROM Drives 65
2.4.2 Printer 66
2.4.3 Hard Disk Controller 69
2.4.4 Graphics Card and Monitor (SaX2) 70
2.4.5 Hardware Information 79

- 2.4.6 IDE DMA Mode 80
- 2.4.7 Mouse 80
- 2.4.8 Scanner 80
- 2.4.9 Sound 82
- 2.5 Network Devices 84
- 2.6 Network Services 84
  - 2.6.1 DHCP Server 84
  - 2.6.2 Host Name and DNS 84
  - 2.6.3 NFS Client and NFS Server 84
  - 2.6.4 Configuration of a Samba Server 84
  - 2.6.5 Configuration of Samba Clients 85
  - 2.6.6 NTP Client 85
  - 2.6.7 Routing 85
  - 2.6.8 Mail Transfer Agent 85
  - 2.6.9 Mail Server 86
  - 2.6.10 Network Services (inetd) 87
- 2.7 Security and Users 88
  - 2.7.1 User Administration 88
  - 2.7.2 Group Administration 89
  - 2.7.3 Security Settings 90
  - 2.7.4 Firewall 92
- 2.8 System 93
  - 2.8.1 Backup Copy of the System Areas 93
  - 2.8.2 Restoring the System 93
  - 2.8.3 Creating a Boot, Rescue, or Module Disk 94
  - 2.8.4 Boot Loader Configuration 96
  - 2.8.5 LVM 97
  - 2.8.6 EVMS 97
  - 2.8.7 Partitioning 97
  - 2.8.8 Profile Manager (SCPM) 97
  - 2.8.9 Runlevel Editor 98
  - 2.8.10 Sysconfig Editor 98

2.8.11 Time Zone Selection 99
2.8.12 Language Selection 99
2.8.13 Keyboard Layout Selection 99
2.9 Miscellaneous 100
2.9.1 Submitting a Support Request 100
2.9.2 Boot Log 100
2.9.3 System Log 100
2.9.4 Loading a Vendor's Driver CD 101
2.10 YaST in Text Mode (ncurses) 101
2.10.1 Navigation in Modules 102
2.10.2 Restriction of Key Combinations 103
2.10.3 Starting the Individual Modules 104

3 Special Installation Procedures 105
3.1 linuxrc 106
3.1.1 Main Menu 107
3.1.2 System Information 107
3.1.3 Loading Modules 109
3.1.4 Entering Parameters 109
3.1.5 Start Installation or System 111
3.1.6 Potential Problems 113
3.1.7 Passing Parameters to linuxrc 114
3.2 Installation with VNC 115
3.2.1 Preparing for the VNC Installation 115
3.2.2 Clients for the VNC Installation 116
3.3 Text-Based Installation with YaST 116
3.4 Starting SUSE LINUX 118
3.4.1 The Graphical SUSE Screen 118
3.5 Special Installation Procedures 119
3.5.1 Automatic Installation with AutoYaST 119
3.5.2 Installation from a Network Source 120
3.6 Tips and Tricks 122

3.6.1 Creating a Boot Disk in DOS 122
3.6.2 Creating a Boot Disk in a UNIX-Type System 124
3.6.3 Booting from a Floppy Disk (SYSLINUX) 125
3.6.4 Using CD 2 for Booting 125
3.6.5 Supported CD-ROM Drives 126
3.7 ATAPI CD-ROM Hangs While Reading 126
3.8 Permanent Device File Names for SCSI Devices 127
3.9 Partitioning for Experts 128
3.9.1 Size of the Swap Partition 128
3.9.2 Partitioning Proposals for Special Purposes 128
3.9.3 Optimization 129
3.10 LVM Configuration 131
3.10.1 Logical Volume Manager (LVM) 132
3.10.2 LVM Configuration with YaST 133
3.10.3 LVM — Partitioning 133
3.10.4 LVM — Configuring Physical Volumes 136
3.10.5 Logical Volumes 137
3.11 Soft RAID 139
3.11.1 Common RAID Levels 140
3.11.2 Soft RAID Configuration with YaST 140
3.11.3 Troubleshooting 141
3.11.4 For More Information 141
3.12 Mass storage via IP networks – iSCSI 142

4 Central software installation and updating 145
4.1 Setting up a central installation server 146
4.1.1 Configuration with YaST 146
4.1.2 Client installation via the installation server 149
4.2 Managing software updates with the YOU server 150
4.2.1 Configuring the local YOU server 150
4.2.2 Configuring the clients 151
4.3 Booting From The Network 152
4.3.1 Configuration Of tftpd 153
4.3.2 Configuration Of dhcpd 154
4.3.3 Launching The Boot Process 155

5 Updating the System and Package Management 157
5.1 Updating SUSE LINUX 158
5.1.1 Preparations 158
5.1.2 Possible Problems 158
5.1.3 Updating with YaST 160
5.1.4 Manual Update 160
5.2 Software Changes from Version to Version 162
5.2.1 From SLES8 to SLES9 163
5.3 RPM — the Package Manager 168
5.3.1 Verifying Package Authenticity 169
5.3.2 Managing Packages: Install, Update, and Uninstall . . 169
5.3.3 RPM and Patches 171
5.3.4 RPM Queries 173
5.3.5 Installing and Compiling Source Packages 175
5.3.6 Compiling RPM Packages with build 177
5.3.7 Tools for RPM Archives and the RPM Database 178

6 System Repair 179
6.1 Starting YaST System Repair 180
6.2 Automatic Repair 180
6.3 User-Defined Repair 181
6.4 Expert Tools 182
6.5 Using initrd as a Rescue System 183
6.5.1 IPLing the Rescue System 185
6.5.2 Loading DASD Modules 186
6.5.3 Mounting the Root Device 187
6.5.4 Changing to the Mounted File System 188
6.5.5 Executing zipl 188
6.5.6 Exiting the Rescue System 188

II System 191

7 32-bit and 64-bit applications in a 64-bit system environment 193
7.1 Introduction 194
7.2 Runtime support 194
7.3 Software development 195
7.4 Software Compilation on Biarch Platforms 196
7.5 Kernel specifications 197

8 Booting and Boot Managers 199
8.1 Booting a PC 200
8.1.1 Master Boot Record 200
8.1.2 Boot Sectors 200
8.1.3 Booting DOS or Windows 201
8.2 Boot Concepts 201
8.3 Map Files, GRUB, and LILO 202
8.4 Booting with GRUB 203
8.4.1 The GRUB Boot Menu 204
8.4.2 The File device.map 208
8.4.3 The File /etc/grub.conf 209
8.4.4 The GRUB Shell 209
8.4.5 Setting a Boot Password 210
8.4.6 Boot Problems with GRUB 211
8.4.7 For More Information 211
8.5 Booting with LILO 211
8.5.1 Basics 212
8.5.2 Configuring LILO 213
8.5.3 Structure of lilo.conf 213
8.5.4 Installing and Uninstalling LILO 217
8.6 Configuring the Boot Loader with YaST 218
8.6.1 The Main Window 218
8.6.2 Boot Loader Configuration Options 220
8.7 Uninstalling the Linux Boot Loader 221

8.7.1 Restoring the MBR (DOS, Win9x, or ME) 222
8.7.2 Restoring the MBR of Windows XP 222
8.7.3 Restoring the MBR of Windows 2000 222
8.8 Creating Boot CDs 223
8.8.1 Procedure 223
8.8.2 Boot CD with ISOLINUX 223
8.9 The Boot Loader ZIPL 224
8.9.1 For use with Kernel Version 2.4.x 225
8.9.2 The ZIPL Configuration File 226

9 The Linux Kernel 227
9.1 Kernel Update 228
9.2 Kernel Sources 229
9.3 Kernel Configuration 229
9.3.1 Configuration on the Command Line 229
9.3.2 Configuration in Text Mode 230
9.3.3 Configuration in the X Window System 230
9.4 Kernel Modules 230
9.4.1 Hardware Detection with the Help of hwinfo 231
9.4.2 Handling Modules 231
9.4.3 /etc/modprobe.conf 232
9.4.4 Kmod — the Kernel Module Loader 232
9.5 Settings in the Kernel Configuration 232
9.6 Compiling the Kernel 233
9.7 Installing the Kernel 234
9.8 Cleaning Your Hard Disk after Compilation 235

10 Special Features of SUSE LINUX 237
10.1 Linux Standards 238
10.1.1 Linux Standard Base (LSB) 238
10.1.2 File System Hierarchy Standard (FHS) 238
10.1.3 teTeX — TeX in SUSE LINUX 238
10.1.4 Example Environment for FTP Server 238

10.1.5 Example Environment for HTTP Server 239
10.2 Hints on Special Software Packages 239
10.2.1 Package bash and /etc/profile 239
10.2.2 cron Package 240
10.2.3 Log Files: the Package logrotate 240
10.2.4 Man Pages 242
10.2.5 The Command ulimit 242
10.2.6 The free Command 243
10.2.7 The File /etc/resolv.conf 244
10.2.8 Settings for GNU Emacs 244
10.3 Booting with the Initial Ramdisk 245
10.3.1 Concept of the Initial Ramdisk 246
10.3.2 The Order of the Booting Process with initrd 246
10.3.3 Boot Loaders 247
10.3.4 Using initrd in SUSE 247
10.3.5 Possible Difficulties — Custom Kernels 249
10.3.6 Prospects 249
10.4 The SUSE Rescue System 249
10.4.1 Starting the Rescue System 250
10.4.2 Working with the Rescue System 251
10.5 Virtual Consoles 254
10.6 Keyboard Mapping 254
10.7 Local Adjustments — I18N and L10N 255
10.7.1 Some Examples 256
10.7.2 Settings for Language Support 257

11 The SUSE LINUX Boot Concept 259
11.1 The init Program 260
11.2 Runlevels 260
11.3 Changing Runlevels 262
11.4 Init Scripts 263
11.4.1 Adding init Scripts 265
11.5 The YaST Runlevel Editor 266
11.6 SuSEconfig and /etc/sysconfig 268
11.7 The YaST sysconfig Editor 270

12 The X Window System 273

12.1 Optimizing the Installation of the X Window System 274
  12.1.1 Screen Section 276
  12.1.2 Device Section 277
  12.1.3 Monitor and Modes Section 278
12.2 Installing and Configuring Fonts 279
  12.2.1 Font Systems 280
12.3 OpenGL — 3D Configuration 285
  12.3.1 Hardware Support 285
  12.3.2 OpenGL Drivers 286
  12.3.3 The Diagnosis Tool 3Ddiag 286
  12.3.4 OpenGL Test Utilities 286
  12.3.5 Troubleshooting 286
  12.3.6 Installation Support 287
  12.3.7 Additional Online Documentation 287

13 Printer Operation 289

13.1 Updating, upgrading and migrating the print system 290
  13.1.1 Updating CUPS 290
  13.1.2 Migration from LPRng/lpdfilter to CUPS 291
13.2 Preparation and Other Considerations 293
13.3 Methods and Protocols for Connecting Printers 295
13.4 Installing the Software 295
13.5 Configuring the Printer 296
  13.5.1 Local Printers 296
  13.5.2 Network Printers 297
  13.5.3 Configuration Tasks 298
13.6 Special Features in SUSE LINUX 299
  13.6.1 Administration with the Web Front-end (CUPS) 299
  13.6.2 Changes in the CUPS Print Service (cupsd) 300
  13.6.3 PPD Files in SUSE Packages 301
13.7 Printer Hardware 304

13.7.1 Printers that Do Not Support a Standard Printer Language 304
13.7.2 No Suitable PPD File Available for a PostScript Printer 305
13.7.3 Parallel Ports 305
13.7.4 Troubleshooting Network Printers 306
13.7.5 Defective Printouts without Error Message 309
13.7.6 Disabled Queues 309
13.7.7 CUPS Browsing: Deleting Print Jobs 309
13.7.8 Defective Print Jobs and Data Transfer Errors 310
13.7.9 Troubleshooting the CUPS Print System 311

14 The Hotplug System 313

14.1 Devices and Interfaces 314
14.2 Hotplug Events 314
14.3 Hotplug Agents 315
14.4 Automatic Module Loading 316
14.5 Network Devices and Interface Designations 317
14.6 Hotplug with PCI 317
14.7 Coldplug 317
14.8 Error Analysis 318
14.8.1 Log Files 318
14.8.2 Boot Problems 318
14.8.3 The Event Recorder 318

15 Dynamic Device Nodes With udev 319

15.1 Background Information 320
15.2 Basics On How To Create Rules 320
15.3 Automatisms With NAME And SYMLINK 321
15.4 Regular Expressions In Keys 321
15.5 Tips On The Selection Of Suitable Keys 322
15.6 Consistent Names For Mass Storage Devices 323

16 Linux on Mobile Devices 325

16.1 PCMCIA . . . 326

16.1.1 The Hardware . . . 326

16.1.2 The Software . . . 326

16.1.3 Configuration . . . 328

16.1.4 Troubleshooting . . . 330

16.1.5 Installation with PCMCIA . . . 334

16.1.6 Other Utilities . . . 335

16.1.7 Updating the Kernel or PCMCIA Package . . . 335

16.1.8 For More Information . . . 336

16.2 SCPM — System Configuration Profile Management . . . 336

16.2.1 Basic Terminology and Concepts . . . 337

16.2.2 The YaST Profile Manager . . . 338

16.2.3 Configuring SCPM . . . 338

16.2.4 Creating and Managing Profiles . . . 338

16.2.5 Switching Configuration Profiles . . . 339

16.2.6 Advanced Profile Settings . . . 340

16.2.7 Profile Selection at Boot . . . 341

16.2.8 Troubleshooting . . . 342

16.3 IrDA — Infrared Data Association . . . 343

16.3.1 Software . . . 343

16.3.2 Configuration . . . 344

16.3.3 Usage . . . 344

16.3.4 Troubleshooting . . . 345

16.4 Bluetooth — Wireless Connections . . . 346

16.4.1 Profiles . . . 346

16.4.2 Software . . . 346

16.4.3 Configuration . . . 346

16.4.4 System Components and Useful Tools . . . 347

16.4.5 Examples . . . 349

16.4.6 Troubleshooting . . . 351

16.4.7 For More Information . . . 352

17 Power Management **353**
17.1 Power Saving Functions 354
17.2 APM 356
  17.2.1 The APM Daemon (apmd) 357
  17.2.2 Further Commands 358
17.3 ACPI 358
  17.3.1 ACPI in Action 359
  17.3.2 The ACPI Daemon (acpid) 361
  17.3.3 ACPI Tools 362
  17.3.4 Troubleshooting 362
17.4 Rest for the Hard Disk 364
17.5 powersave 365
  17.5.1 Configuration of powersave 366
  17.5.2 Configuration of APM and ACPI 366
  17.5.3 Additional ACPI Features 368
  17.5.4 Troubleshooting 369
17.6 The YaST Power Management Module 371
17.7 WOL – Wake On LAN 373
  17.7.1 BIOS Configuration 375
  17.7.2 Configuration With YaST 376
  17.7.3 Waking Up Computers 376
  17.7.4 Further Information 376

18 File Systems in Linux **377**
18.1 Glossary 378
18.2 Major File Systems in Linux 378
  18.2.1 Ext2 379
  18.2.2 Ext3 380
  18.2.3 ReiserFS 381
  18.2.4 JFS 382
  18.2.5 XFS 383
18.3 Some Other Supported File Systems 384
18.4 Large File Support in Linux 385
18.5 For More Information 386

19 High availability under Linux 389

19.1 What does high availability mean? 390

19.2 Important terms 390

19.3 A sample minimum scenario 391

19.4 Components of a high availability solution 392

19.5 The software side of high availability 394

19.5.1 heartbeat 394

19.5.2 RAID 394

19.5.3 rsync 394

19.5.4 DRBD 395

19.6 Clustering 395

19.6.1 Cluster alias 395

19.6.2 Linux Virtual Server 395

19.6.3 High availability clusters 396

19.7 Further information 396

19.7.1 High availability (HA) in general and heartbeat 396

19.7.2 DRBD 396

19.7.3 RAID 397

19.7.4 Clustering 397

20 PAM — Pluggable Authentication Modules 399

20.1 Structure of a PAM Configuration File 400

20.2 The PAM Configuration of sshd 402

20.3 Configuration of PAM Modules 403

20.3.1 pam_unix2.conf 403

20.3.2 pam_env.conf 404

20.3.3 pam_pwcheck.conf 405

20.3.4 limits.conf 405

20.4 For More Information 406

III Services 407

21 Linux in the Network 409
21.1 TCP/IP — The Protocol Used by Linux 410
21.1.1 Layer Model 411
21.1.2 IP Addresses and Routing 414
21.1.3 Domain Name System 417
21.2 IPv6 — The Next Generation Internet 418
21.2.1 Advantages of IPv6 419
21.2.2 The IPv6 Address System 420
21.2.3 Coexistance of IPv4 and IPv6 426
21.2.4 For More Information 428
21.3 Manual Network Configuration 428
21.3.1 Configuration Files 430
21.3.2 Start-Up Scripts 436
21.4 Network Integration 437
21.4.1 Requirements 437
21.4.2 Configuring the Network Card with YaST 438
21.4.3 Modem 441
21.4.4 DSL 443
21.4.5 ISDN 445
21.4.6 Hotplug and PCMCIA 449
21.4.7 Configuring IPv6 450
21.5 Routing in SUSE LINUX 450
21.6 SLP services in the network 452
21.6.1 SLP support in SUSE LINUX 452
21.6.2 Further information 455
21.7 DNS — Domain Name System 455
21.7.1 Starting the Name Server BIND 455
21.7.2 The Configuration File /etc/named.conf 457
21.7.3 Important Configuration Options 458
21.7.4 The Configuration Section Logging 459

21.7.5 Zone Entry Structure 460
21.7.6 Structure of Zone Files 461
21.7.7 Secure Transactions 464
21.7.8 Dynamic Update of Zone Data 466
21.7.9 DNSSEC 466
21.7.10 Configuration with YaST 467
21.7.11 For More Information 482
21.8 LDAP — A Directory Service 483
21.8.1 LDAP versus NIS 485
21.8.2 Structure of an LDAP Directory Tree 486
21.8.3 Server Configuration with slapd.conf 489
21.8.4 Data Handling in the LDAP Directory 493
21.8.5 LDAP Server Configuration with YaST 498
21.8.6 The YaST LDAP Client 502
21.8.7 For More Information 510
21.9 NIS — Network Information Service 513
21.9.1 NIS Master and Slave Servers 513
21.9.2 The NIS Client Module of YaST 516
21.10 NFS — Shared File Systems 518
21.10.1 Importing File Systems with YaST 518
21.10.2 Importing File Systems Manually 519
21.10.3 Exporting File Systems with YaST 519
21.10.4 Exporting File Systems Manually 519
21.11 DHCP 522
21.11.1 The DHCP Protocol 522
21.11.2 DHCP Software Packages 523
21.11.3 The DHCP Server dhcpd 524
21.11.4 Hosts with Fixed IP Addresses 526
21.11.5 The SUSE LINUX Version 527
21.11.6 DHCP Configuration with YaST 528
21.11.7 For More Information 534
21.12 Time Synchronization with xntp 535
21.12.1 Configuration in the Network 536
21.12.2 Setting up a Local Reference Clock 537

22 The Apache Web Server 541

22.1 Basics 542
    22.1.1 Web Server 542
    22.1.2 HTTP 542
    22.1.3 URLs 542
    22.1.4 Automatic Display of a Default Page 543
22.2 Setting up the HTTP Server with YaST 543
22.3 Apache Modules 544
22.4 New Features of Apache 2 545
22.5 Threads 546
22.6 Installation 546
    22.6.1 Package Selection in YaST 546
    22.6.2 Activating Apache 547
    22.6.3 Modules for Active Contents 547
    22.6.4 Other Recommended Packages 547
    22.6.5 Installation of Modules with apxs 547
22.7 Configuration 548
    22.7.1 Configuration with SuSEconfig 548
    22.7.2 Manual Configuration 549
22.8 Using Apache 553
22.9 Active Contents 553
    22.9.1 Server Side Includes: SSI 555
    22.9.2 Common Gateway Interface: CGI 555
    22.9.3 GET and POST 555
    22.9.4 Languages for CGI 556
    22.9.5 Generating Active Contents with Modules 556
    22.9.6 mod_perl 556
    22.9.7 mod_php4 559
    22.9.8 mod_python 559
    22.9.9 mod_ruby 559
22.10 Virtual Hosts 560
    22.10.1 Name-Based Virtual Hosts 560

22.10.2 IP-Based Virtual Hosts 561
22.10.3 Multiple Instances of Apache 562
22.11 Security 563
22.11.1 Minimizing the Risk 563
22.11.2 Access Permissions 563
22.11.3 Staying Updated 563
22.12 Troubleshooting 564
22.13 For More Information 564
22.13.1 Apache 564
22.13.2 CGI 564
22.13.3 Security 565
22.13.4 Additional Sources 565

23 File Synchronization **567**
23.1 Available Data Synchronization Software 568
23.1.1 Unison 568
23.1.2 CVS 569
23.1.3 subversion 569
23.1.4 mailsync 569
23.1.5 rsync 570
23.2 Determining Factors for Selecting a Program 570
23.2.1 Client-Server versus Peer-to-Peer 570
23.2.2 Portability 570
23.2.3 Interactive versus Automatic 570
23.2.4 Conflicts: Incidence and Solution 571
23.2.5 Selecting and Adding Files 571
23.2.6 History 571
23.2.7 Data Volume and Hard Disk Requirements 572
23.2.8 GUI 572
23.2.9 User Friendliness 572
23.2.10 Security against Attacks 572
23.2.11 Protection against Data Loss 573

23.3 Introduction to Unison 574
  23.3.1 Requirements 574
  23.3.2 Using Unison 574
  23.3.3 For More Information 575
23.4 Introduction to CVS 575
  23.4.1 Configuring a CVS Server 576
  23.4.2 Using CVS 577
  23.4.3 For More Information 578
23.5 Introduction to Subversion 578
  23.5.1 Installing a Subversion Server 578
  23.5.2 Usage and Operation 579
  23.5.3 For More Information 581
23.6 An Introduction to rsync 581
  23.6.1 Configuration and Operation 581
  23.6.2 For More Information 583
23.7 Introduction to mailsync 583
  23.7.1 Configuration and Use 583
  23.7.2 Possible Problems 585
  23.7.3 For More Information 586

24 Heterogenous Networks 587
24.1 Samba 588
  24.1.1 Introduction to Samba 588
  24.1.2 Installing and Configuring the Server 590
  24.1.3 Samba as Login Server 594
  24.1.4 Installation and Configuration with YaST 596
  24.1.5 Installing Clients 596
  24.1.6 Optimization 598
24.2 Netatalk 599
  24.2.1 Configuring the File Server 601
  24.2.2 Configuring the Print Server 606
  24.2.3 Starting the Server 606
  24.2.4 Additional Information 607

25 Internet 609

25.1 The smpppd as Dial-up Assistant 610

25.1.1 Program Components for the Internet Dial-Up 610

25.1.2 Configuring smpppd 610

25.1.3 Configuring kinternet and cinternet for Remote Use . . 611

25.2 Configuring an ADSL or T-DSL Connection 612

25.2.1 Default Configuration 612

25.2.2 DSL Connection by Dial-on-Demand 613

25.3 Proxy Server: Squid 614

25.3.1 Squid as Proxy Cache 614

25.3.2 Some Facts about Proxy Caches 614

25.3.3 System Requirements 616

25.3.4 Starting Squid 618

25.3.5 The Configuration File /etc/squid/squid.conf 619

25.3.6 Configuring a Transparent Proxy 624

25.3.7 Squid and Other Programs 627

26 Security in the Network 633

26.1 X.509 certification with YaST 634

26.1.1 The principles of digital certification 634

26.1.2 YaST modules for CA management 638

26.2 VPN with SUSE LINUX 647

26.2.1 Setting up Road Warrior servers 648

26.2.2 Setting up a VPN Linux client with FreeS/WAN 651

26.2.3 IPsec clients on Windows XP and Windows 2000 653

26.3 Masquerading and Firewalls 657

26.3.1 Packet Filtering with iptables 657

26.3.2 Masquerading Basics 659

26.3.3 Firewalling Basics 660

26.3.4 SuSEfirewall2 661

26.3.5 For More Information 664

26.4 SSH — Secure Shell, the Safe Alternative 665

26.4.1 The OpenSSH Package 665
26.4.2 The ssh Program 665
26.4.3 scp — Secure Copy 666
26.4.4 sftp — Secure File Transfer 667
26.4.5 The SSH Daemon (sshd) — Server-Side 667
26.4.6 SSH Authentication Mechanisms 669
26.4.7 X, Authentication, and Other Forwarding Mechanisms 670
26.5 Network Authentication — Kerberos 671
26.5.1 Kerberos Terminology 672
26.5.2 How Kerberos Works 673
26.5.3 Users’ View of Kerberos 676
26.5.4 For More Information 677
26.6 Installing and Administering Kerberos 678
26.6.1 Choosing the Kerberos Realms 678
26.6.2 Setting up the KDC Hardware 679
26.6.3 Clock Synchronization 680
26.6.4 Log Configuration 680
26.6.5 Installing the KDC 681
26.6.6 Configuring Kerberos Clients 683
26.6.7 Remote Kerberos Administration 687
26.6.8 Creating Kerberos Host Principals 689
26.6.9 Enabling PAM Support for Kerberos 690
26.6.10 Configuring SSH for Kerberos Authentication 691
26.6.11 Using LDAP and Kerberos 692
26.7 Security and Confidentiality 695
26.7.1 Basic Considerations 695
26.7.2 Local Security and Network Security 695
26.7.3 Some General Security Tips and Tricks 704
26.7.4 Using the Central Security Reporting Address 706

IV Administration 709

27 Access Control Lists in Linux 711

27.1 Advantages of ACLs 712

27.2 Definitions 713

27.3 Handling ACLs 714

27.3.1 Structure of ACL Entries 714

27.3.2 ACL Entries and File Mode Permission Bits 715

27.3.3 A Directory with Access ACL 716

27.3.4 A Directory with a Default ACL 719

27.3.5 The ACL Check Algorithm 722

27.4 Support by Applications 723

28 System Monitoring Utilities 725

28.1 List of Open Files: lsof 726

28.2 User Accessing Files: fuser 727

28.3 File Properties: stat 728

28.4 Processes: top 728

28.5 Process List: ps 729

28.6 Process Tree: pstree 730

28.7 Who Is Doing What: w 731

28.8 Memory Usage: free 731

28.9 Kernel Ring Buffer: dmesg 732

28.10 File Systems and Their Usage: mount, df, and du 733

28.11 The /proc File System 734

28.12 procinfo 735

28.13 PCI Resources: lspci 736

28.14 System Calls of a Program Run: strace 737

28.15 Library Calls of a Program Run: ltrace 738

28.16 Specifying the Required Library: ldd 739

28.17 Interprocess Communication: ipcs 739

V Appendix 741

A Information sources and documentation 743

B Manual Page of e2fsck 747

C Manual Page of reiserfsck 753

D The GNU General Public License 757

Bibliography 765

Introduction

This book guides you from the initial installation of your SUSE LINUX Enterprise Server through to full configuration of your system and complex administration tasks. It contains, in compressed form, descriptions of the installation and administration tasks for all hardware platforms supported by SUSE LINUX Enterprise Server.

Note

Preparing for Installation

Detailed information about how to prepare your hardware platform can be found in the Architecture-specific Information manual, which is located in the `preparation.pdf` file contained in the `/docu/` directory of the first CD that comes with SUSE LINUX Enterprise Server.

Note

Organization of the Manual

This manual is divided into four main subject areas:

Installation Follow this part from the first installation dialog through to the ready configured system. It also tells you how to use the various SUSE LINUX Enterprise Server installation types, how to configure installation sources, and how to update individual software packages or the overall operating system.

System This part provides detailed information about the organization of your Linux system and how it works, how to distinguish between the 32-(31-)bit and 64-bit worlds, and how to configure a graphical interface and a print infrastructure.

Services This part covers the main server services on your system and their configuration with YaST. This is followed by information about security, firewalls, CA administration, VPN configuration, and authentication procedures.

Administration Finally, this part contains information about the use of file system ACLs and a chapter about the various system monitoring programs under Linux.

The appendix provides a summary of the most important sources of information on Linux.

Conventions and Abbreviations

The following typographic conventions are used in this manual:

Font Style	Meaning
YaST	This font is used to indicate a program name.
`/etc/passwd`	This font is used to specify a file or directory.
⟨*placeholder*⟩	The character string `placeholder` (including angle brackets) should be replaced by the actual value.
`PATH`	This indicates an environment variable called `PATH`.
`192.168.1.2`	The value of a variable.
`ls`	Specifies a command to enter.
`user`	This style refers to a user.
Alt	This symbol indicates a key to press.
Ctrl-Alt-Del	This indicates two or more keys to press simultaneously.
"Permission denied"	This is a system message.
‘Update system’	Menu option or button.

The manual uses the following naming conventions and abbreviations for the hardware architectures supported by SUSE LINUX Enterprise Server:

Abbreviation	Full Name or Processor Name
x86	x86
amd64	AMD64
em64t	Intel EM64T
ipf	Itanium Processor Family
ppc	POWER
ppc64	POWER
s390	IBM S/390
zseries/s390x	IBM zSeries

To indicate differences between the individual platforms in continuous text, the authors use the conventions listed below. The type of identifier depends on the amount of divergent information. Here are a few brief examples:

If only a few words or characters within a sentence or section are different, the difference is worked into the continuous text without any further identifier.

Note

zseries: A Brief Tip

Brief tips, notes, or warnings that apply exclusively to one or more architectures are identified by the platform abbreviation followed by the actual title of the note.

Note

► **em64t**
Entire paragraphs that are different on different platforms are separated using two graphical symbols plus the platform abbreviation. ◄

amd64: A Section That Applies to AMD64 Only

If a complete section of a chapter is concerned with a single platform, this is introduced, as in the case of tips, notes, and warnings, by the platform abbreviation in the title, so the table of contents for the manual reflects this difference.

Labeling of screenshots and examples of code are provided in similar fashion — the relevant caption or example title is preceded by the platform abbreviation. If no abbreviation is used, the information may be assumed to be valid for all platforms.

Target Audience

The authors have assumed that readers of this manual possess the following basic knowledge:

- You are familiar with the terminology used for your hardware platform or have access to the appropriate documentation of the hardware vendor.
- You are familiar with the characteristics and special features of your hardware platform.
- You have at least a basic knowledge of how to administer a Linux system.

Part I

Installation

1

Installation with YaST

After your hardware has been prepared for the installation of SUSE LINUX Enterprise Server as described in the Architecture-specific Information manual and after the connection with the installation system has been established, you will be presented with the (graphical) interface of SUSE's system assistant YaST. YaST takes care of all the following steps to set up the system, guiding you through the entire installation and configuration procedure.

1.1 s390, zseries: Booting the System into the Installation Program 8
1.2 Booting into the Installation Program 8
1.3 The Boot Screen 10
1.4 Language Selection 12
1.5 s390, zseries: Hard Disk Configuration 12
1.6 Installation Mode 14
1.7 Installation Suggestion 15
1.8 Finishing the Installation 35
1.9 Hardware Configuration 44
1.10 Graphical Login 46

1.1 s390, zseries: Booting the System into the Installation Program

For IBM's S/390 and zSeries platforms, the system is initialized (IPL) as described in the Architecture-specific Information manual. SUSE LINUX Enterprise Server does not show a splash screen on these systems, because including for the installation, kernel and initrd need to be loaded manually.

YaST starts with its installation screen as soon as a connection has been established to the installation system via VNC, X, or SSH. As there is no splash screen, kernel or boot parameters cannot be entered on screen, but must be specified in a parmfile (see the parmfile chapter in the Architecture-specific Information manual for a description).

Note

s390, zseries: The Next Steps

To install, follow the description of the installation procedure with YaST starting from Section 1.4 on page 12.

Note

1.2 Booting into the Installation Program

Insert the first SUSE LINUX CD or the DVD into the drive. Then reboot the computer to start the installation program from the medium in the drive.

1.2.1 Possible Problems when Starting from the CD or DVD

If you experience problems booting from the CD or DVD, there may be a number of reasons for this. If your CD-ROM drive is an older model, it is possible that it is not supported.

SUSE LINUX Professional on the Intel platform only: It is possible that the CD-ROM drive is not able to read the boot image on the first CD. In this case, use CD 2 to boot the system. CD 2 contains a conventional 2.88 MB boot image, which can be read even by older drives.

The boot sequence in the BIOS (basic input output system) may be incorrect. Information about changing the BIOS settings is provided in the documentation of your motherboard and also in the following paragraphs.

The BIOS is a piece of 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, as 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.

Note

Keyboard Layout in the BIOS

The BIOS is often limited to the US keyboard layout.

Note

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).

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 (Page up) or (Page down) until the sequence is `A,CDROM,C`.

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).

If you have a SCSI CD-ROM drive, change the setup of the SCSI BIOS. In the case of an Adaptec host adapter, for instance, open the setup by pressing (Ctrl)-(A). After that, select 'Disk Utilities', which displays the connected hardware components. Make a note of the SCSI ID for your CD-ROM drive. Exit the menu with (Esc) then open 'Configure Adapter Settings'. Under 'Additional Options', select 'Boot Device Options' and press (Enter). Enter the ID of the CD-ROM drive and press (Enter) again. Then press (Esc) twice to return to the start screen of the SCSI BIOS. Exit this screen and confirm with 'Yes' to boot the computer.

1.3 The Boot Screen

The boot screen has a number of menu items from which to select. 'Boot from Hard Disk' top boots the system already installed on the host (if any). This item is selected by default, because the CD is often left in the drive. To install the system, select 'Installation' with the arrow keys. This loads YaST and starts the installation.

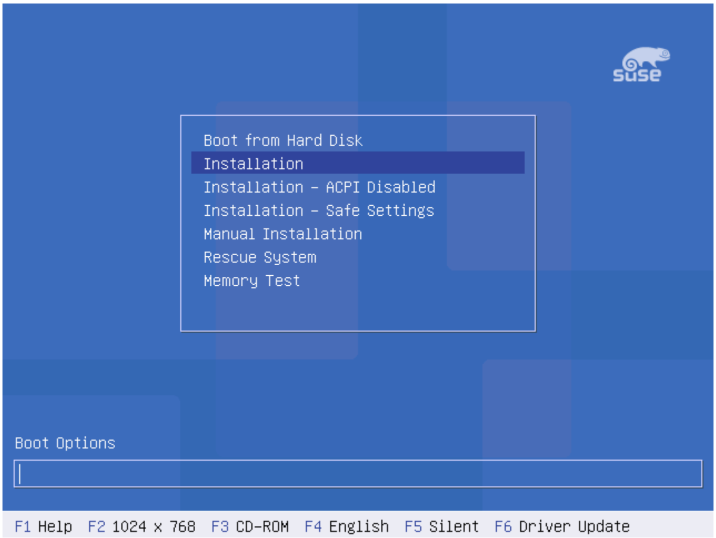

Boot from Hard Disk
Installation
Installation - ACPI Disabled
Installation - Safe Settings
Manual Installation
Rescue System
Memory Test
Boot Options
F1 Help F2 1024 x 768 F3 CD-ROM F4 English F5 Silent F6 Driver Update

Figure 1.1: The Boot Screen

The menu items in the boot screen provide a number of options for starting from the CD-ROM. They trigger the following actions:

Boot from Hard Disk Boots the system on the hard disk (the system normally booted when the machine is started). This option is preselected.

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

Installation — ACPI Disabled If the normal installation fails, this may 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 — Safe Settings Boots the system with the DMA mode (for CD-ROM drives) and any interfering power management functions disabled. Experts can also use the command line to enter or change kernel parameters.

Manual Installation By default, drivers are loaded automatically during the installation. If this appears to cause problems, use this option to load drivers *manually*. However, this does not work if you use a USB keyboard on your machine.

Rescue System If you are unable to boot into your installed Linux system for some reason, you can boot the computer from the DVD or CD1 and select this item. This starts a *rescue system* — a minimal Linux system without a graphical user interface, which allows experts to access disk partitions for troubleshooting and repairing an installed system. Less experienced users can alternatively use the system repair tool supplied with YaST. Refer to Chapter 6 on page 179 for details.

Memory Test This tests your system RAM by means of repeated read and write cycles. This is done in an endless loop, because memory corruption often shows up very sporadically and many read and write cycles might be necessary to detect it. If you suspect that your RAM might be defective, start this test and let it run for several hours. If no errors are detected after a longer time period, you can assume that the memory is intact. Terminate the test by rebooting.

Use the function keys, as indicated in the bar at the bottom of the screen, to change a number of installation settings, if needed.

F1 Access context-sensitive help — help for the currently active screen element of the boot screen.

F2 Select different graphical display modes for the installation. Also included is an entry to select the text mode, which is useful if the installation in graphical mode causes problems for some reason.

F3 Choose among different installation media. Normally, install from the inserted installation disk, but in some cases you may want to select another source, such as FTP or NFS. The SLP (service location protocol) entry allows you to access an SLP server in the network, which in turn gives access to a selection of installation media as made available by that server. Details of the SLP protocol are discussed in Section 21.6 on page 452.

F4 Select the display language for the installation.

F5 By default, diagnostic messages of the Linux kernel are not displayed during system start-up. You only see a progress bar. To display these messages, select 'Native'. For a maximum of information, select 'Verbose'.

F6 Allows you to tell the system that you have an optional disk with a driver update for SUSE LINUX. You will be asked to insert the update disk at the appropriate point in the installation process.

A few seconds after starting the installation, SUSE LINUX loads a minimal Linux system to run the installation procedure. A number of messages and copyright notices scrolls by and, at the end of the loading process, the YaST installation program is started. After a few more seconds, the screen should display the graphical interface that will guide you through the installation.

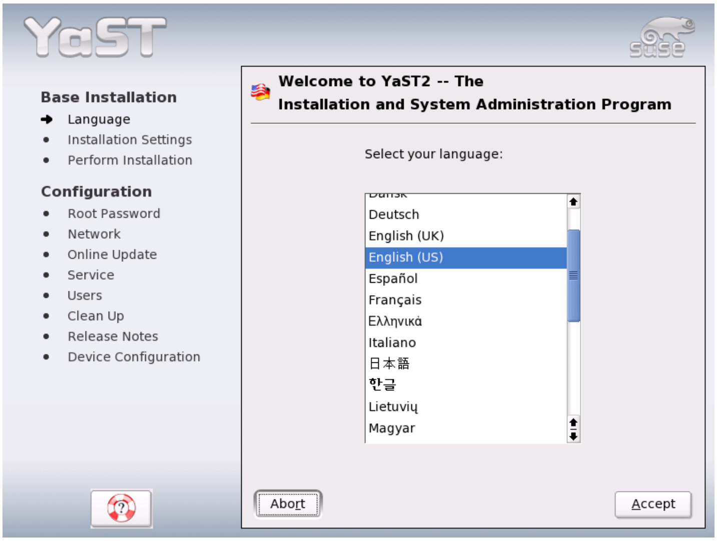
This is where the actual installation procedure begins, which is controlled by the YaST installation program. All YaST screens have a common layout. All buttons, entry fields, and lists can be accessed with the mouse. If your mouse pointer cannot move, the mouse has not been autodetected. For the time being, you can use the keyboard for navigation.

1.4 Language Selection

YaST and SUSE LINUX in general can be configured to use different languages according to your needs. The language selected at this point is the default used for the keyboard layout. In addition, YaST uses the language setting to guess a time zone for the system clock. If your mouse does not work, navigate with the arrow keys until the desired language is selected. After this, press Tab until 'Next' is highlighted. Then press Enter to confirm your language selection.

1.5 s390, zseries: Hard Disk Configuration

When installing on IBM S/390 and zSeries platforms, the language selection dialog is followed by a dialog to configure the attached hard disks (see



YaST
SuSE
Base Installation
Language
Installation Settings
Perform Installation
Configuration
Root Password
Network
Online Update
Service
Users
Clean Up
Release Notes
Device Configuration
Abort
Accept

Welcome to YaST2 -- The Installation and System Administration Program

Select your language:

- Dansk
- Deutsch
- English (UK)
- English (US)
- Español
- Français
- Ελληνικά
- Italiano
- 日本語
- 한글
- Lietuvių
- Magyar

Figure 1.2: *Selecting the Language*

Figure 1.3 on the next page). Select DASDs and Fibre Channel Attached SCSI Disks (ZFCP)

for the installation of SUSE LINUX Enterprise Server.

After selecting 'Configure DASD Disks', you are presented with an overview listing 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 1.4 on page 15).

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 1.5 on page 16). To format the DASDs, select 'Perform Action' → 'Format' right away or use the YaST partitioner later (see Section 1.7.4 on page 21).

To use ZFCP disks for the SUSE LINUX Enterprise Server installation, select 'Configure ZFCP Disks' in the selection dialog (see Figure 1.3 on the next page). 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 1.6 on page 17).

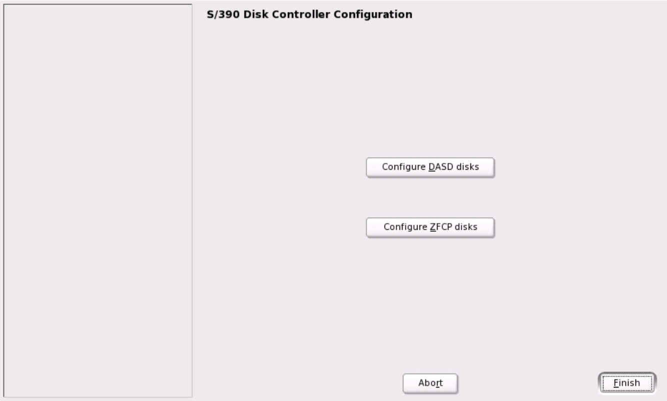


Figure 1.3: *s390, zseries: Configuring the Hard Disk Controllers*

To make a ZFCP disk available for the SUSE LINUX Enterprise Server installation, use the entry fields ‘Channel Number’, ‘WWPN’ (World Wide Port), and ‘FCP-LUN’ to specify the parameters identifying the corresponding disk (see Figure 1.7 on page 18).

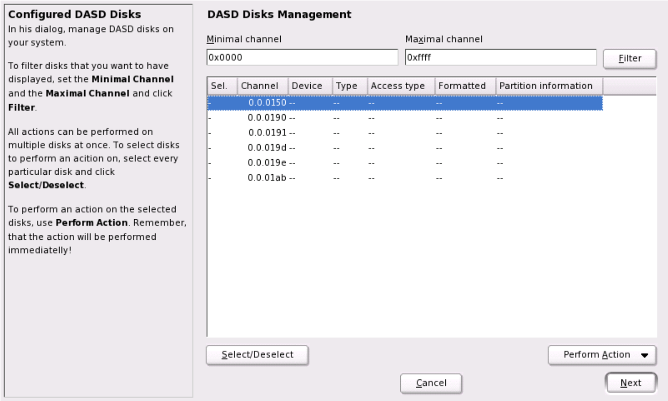
When you are done, exit the ZFCP dialog with ‘Next’ and the general hard disk configuration dialog (Figure 1.3) with ‘Finish’ to continue with the rest of the configuration.

1.6 Installation Mode

In this dialog, select between a ‘New Installation’ and an ‘Update of an Installed System’. The latter is, of course, only possible if a previous version of SUSE LINUX is already present. In this case, you can also boot into this system with ‘Boot Installed System’. If your installed system fails to boot, perhaps because some important system configuration has been corrupted, you can try to make the system bootable again with ‘System Repair’. Without a previously-installed version of SUSE LINUX, it is only possible to perform a new installation.

Click ‘OK’ to continue. See Figure 1.8 on page 19.

The following sections describe the procedure of installing a new system. Detailed instructions for a system update can be found in Section 2.3.5 on



Configured DASD Disks
In his dialog, manage DASD disks on your system.

To filter disks that you want to have displayed, set the **Minimal Channel** and the **Maximal Channel** and click **Filter**.

All actions can be performed on multiple disks at once. To select disks to perform an acition on, select every particular disk and click **Select/Deselect**.

To perform an action on the selected disks, use **Perform Action**. Remember, that the action will be performed immediatelly!

DASD Disks Management

Minimal channel: 0x0000

Maximal channel: 0xffff

Sel.	Channel	Device	Type	Access type	Formatted	Partition information
-	0.0.0150	--	--	--	--	--
-	0.0.0190	--	--	--	--	--
-	0.0.0191	--	--	--	--	--
-	0.0.019d	--	--	--	--	--
-	0.0.019e	--	--	--	--	--
-	0.0.01ab	--	--	--	--	--

Figure 1.4: *s390, zseries: Selecting a DASD*

page 61. A description of the system repair options can be found in Chapter 6 on page 179.

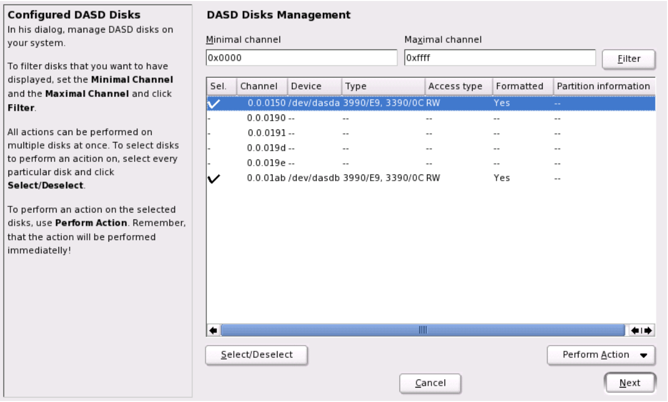
1.7 Installation Suggestion

After hardware detection, the suggestion window (shown in Figure 1.9 on page 20) displays some information about the hardware recognized and proposes a number of installation and partitioning options. After selecting any of these items and configuring them in the corresponding dialogs, you are always returned to the suggestion window, which is updated accordingly. The individual settings are discussed in the following sections.

1.7.1 Installation Mode

Use this to change the previously selected installation mode. The options are the same as already described in Section 1.6 on the preceding page.

1.7.2 Keyboard Layout



Configured DASD Disks

In his dialog, manage DASD disks on your system.

To filter disks that you want to have displayed, set the **Minimal Channel** and the **Maximal Channel** and click **Filter**.

All actions can be performed on multiple disks at once. To select disks to perform an acition on, select every particular disk and click **Select/Deselect**.

To perform an action on the selected disks, use **Perform Action**. Remember, that the action will be performed immediatelly!

DASD Disks Management

Minimal channel: 0x0000

Maximal channel: 0xffff

Sel.	Channel	Device	Type	Access type	Formatted	Partition information
✓	0.0.0150	/dev/dasda	3990/E9, 3390/0C	RW	Yes	--
-	0.0.0190	--	--	--	--	--
-	0.0.0191	--	--	--	--	--
-	0.0.019d	--	--	--	--	--
-	0.0.019e	--	--	--	--	--
✓	0.0.01ab	/dev/dasdb	3990/E9, 3390/0C	RW	Yes	--

Figure 1.5: *s390, zseries: Activating a DASD*

Note

s390, zseries: Keyboard and Mouse Configuration

On IBM S/390 and zSeries platforms, the installation is performed from a remote terminal. The host as such has no keyboard or mouse locally connected to it.

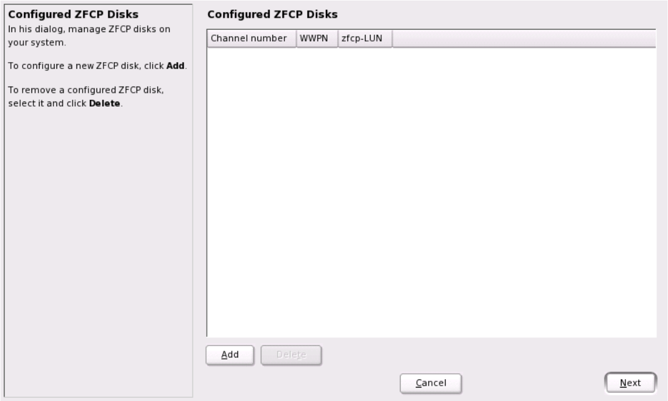
Note

Select the keyboard layout. By default, the layout corresponds to the selected language. After changing the layout, test Y, Z, and special characters to make sure the selection is correct. When finished, select 'Next' to return to the suggestion window.

1.7.3 Mouse

If YaST failed to detect your mouse automatically, press Tab in the suggestion window several times until 'Mouse' is selected. Then use Space to open the dialog in which to set the mouse type. This dialog is shown in Figure 1.10 on page 21.

To select the mouse type, use ↑ and ↓. Consult your mouse documentation for information about the mouse type. After selecting a mouse type, use Alt-T to test whether the device works correctly without making the selection permanent. If the mouse does not behave as expected, use the



Configured ZFCP Disks

In his dialog, manage ZFCP disks on your system.

To configure a new ZFCP disk, click **Add**.

To remove a configured ZFCP disk, select it and click **Delete**.

Figure 1.6: *s390, zseries: Overview of Available ZFCP Disks*

keyboard to select another type and test again. Use Tab and Enter to make the current selection permanent.

1.7.4 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. This section describes the necessary steps.

Partition Types

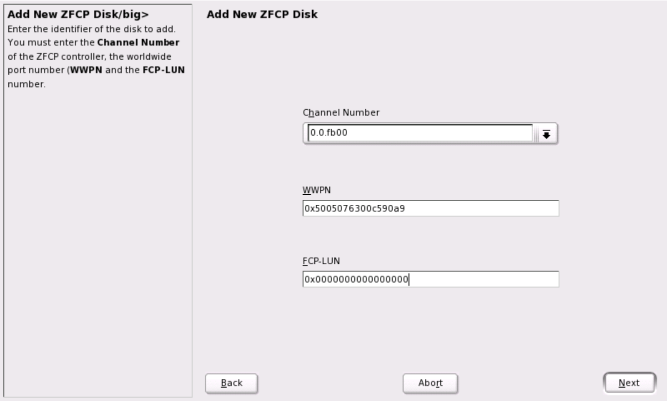
Note

s390, zseries: "Hard Disks" on IBM's S/390 and zSeries platforms

On IBM S/390 and zSeries 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.

Note

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.



Add New ZFCP Disk/big>
Enter the identifier of the disk to add. You must enter the **Channel Number** of the ZFCP controller, the worldwide port number (**WWPN** and the **FCP-LUN** number.

Add New ZFCP Disk

Channel Number: 0.0.fb00

WWPN: 0x5005076300c590a9

FCP-LUN: 0x0000000000000000

Figure 1.7: *s390, zseries: Adding a ZFCP Disk*

A primary partition simply consists of a continuous range of cylinders (physical disk areas), assigned to a particular operating system. With primary partitions only, you would be limited to four partitions per hard disk, because more do not fit in the partition table.

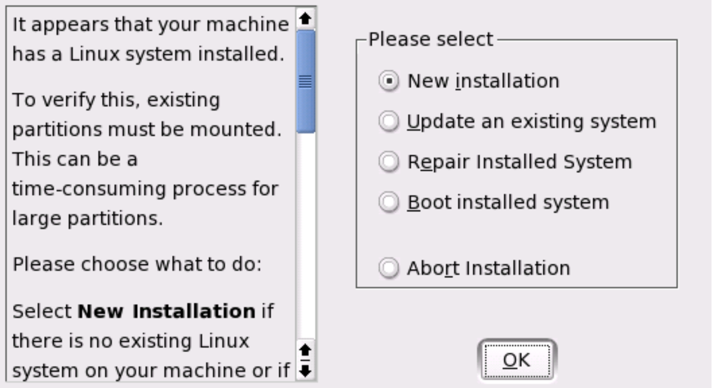
This is why extended partitions are used. Extended partitions are also continuous ranges of disk cylinders, but an extended partition may itself be subdivided into *logical partitions*. Logical partitions do not require entries in the partition table. In other words, an extended partition is a container for logical partitions.

If you need more than four partitions, create an extended partition as soon 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 fifteen on SCSI disks and 63 on (E)IDE disks.

It does not matter which type of partitions are used for Linux. Primary and logical partitions both work fine.

Required Disk Space

YaST normally proposes a reasonable partitioning scheme with sufficient disk space. If you want to implement your own partitioning scheme, consider the following recommendations concerning the requirements for different system types.



It appears that your machine has a Linux system installed.

To verify this, existing partitions must be mounted. This can be a time-consuming process for large partitions.

Please choose what to do:

Select **New Installation** if there is no existing Linux system on your machine or if

Figure 1.8: Selecting the Installation Mode

Minimal System: 500 MB No graphical interface (X Window System) is installed, which means that only console applications can be used. Also, only a very basic selection of software is installed.

Minimal System with Graphical Interface: 700 MB This includes the X Window System and some applications.

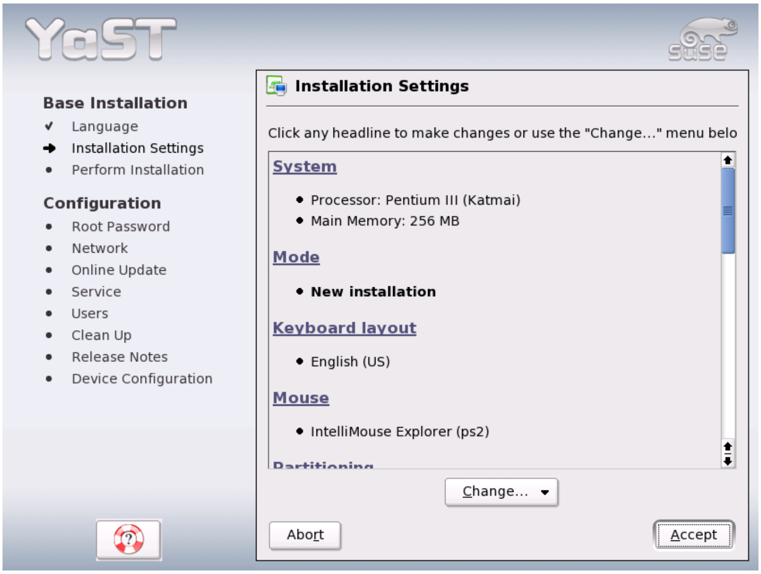
Default System: 1.5 GB This includes a modern desktop environment, like KDE or GNOME, and also provides enough space for large application suites like Netscape or Mozilla.

Full Installation: 2.5 GB All the packages included with SUSE LINUX can be installed.

Depending on the amount of space and how the computer will be used, adjust the distribution of the available disk space. These are some basic guidelines for partitioning:

Up to 4 GB: One partition for the swap space and one root partition (`/`). In this case, the root partition must allow for those directories that often reside on their own partitions if more space is available.

4 GB or More: A swap partition, a root partition (1 GB), and one partition each for the following directories as needed: `/usr` (4 GB or more), `/opt` (4 GB or more), and `/var` (1 GB). The rest of the available space can be used for `/home`.


YaST
Base Installation
Language
Installation Settings
Perform Installation
Configuration
Root Password
Network
Online Update
Service
Users
Clean Up
Release Notes
Device Configuration
Change...
Abort
Accept

Installation Settings

Click any headline to make changes or use the "Change..." menu belo

System

- Processor: Pentium III (Katmai)
- Main Memory: 256 MB

Mode

- **New installation**

Keyboard layout

- English (US)

Mouse

- IntelliMouse Explorer (ps2)

Figure 1.9: Suggestion Window

Depending on the hardware, it may also be useful to create a boot partition (`/boot`) to hold the boot mechanism and the Linux kernel. This partition should be located at the start of the disk and should be at least 8 MB or 1 cylinder. As a rule of thumb, always create such a partition if it was included in YaST's original proposal. If you are unsure about this, create a boot partition to be on the safe side.

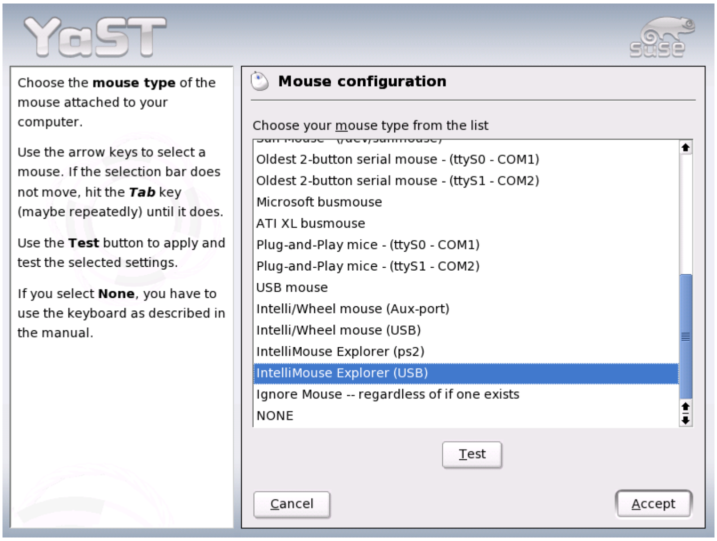
You should also be aware that some (mostly commercial) programs install their data in `/opt`. Therefore, you may either want to create a separate partition for `/opt` or make the root partition large enough. KDE and GNOME are also installed in `/opt`.

Note

Partitioning Tips

Things should be fine if your partitioning setup is similar to that proposed by YaST. This is usually a small partition for `/boot` at the beginning of the hard disk (about 10 MB, or 1 cylinder on a large hard disk), a swap partition (between 256 and 500 MB), and the rest for `/`.

Note



Choose the **mouse type** of the mouse attached to your computer.

Use the arrow keys to select a mouse. If the selection bar does not move, hit the ***Tab*** key (maybe repeatedly) until it does.

Use the **Test** button to apply and test the selected settings.

If you select **None**, you have to use the keyboard as described in the manual.

Mouse configuration

Choose your mouse type from the list

Oldest 2-button serial mouse - (ttyS0 - COM1)
Oldest 2-button serial mouse - (ttyS1 - COM2)
Microsoft busmouse
ATI XL busmouse
Plug-and-Play mice - (ttyS0 - COM1)
Plug-and-Play mice - (ttyS1 - COM2)
USB mouse
Intelli/Wheel mouse (Aux-port)
Intelli/Wheel mouse (USB)
IntelliMouse Explorer (ps2)
IntelliMouse Explorer (USB)
Ignore Mouse -- regardless of if one exists
NONE

Figure 1.10: Selecting the Mouse Type

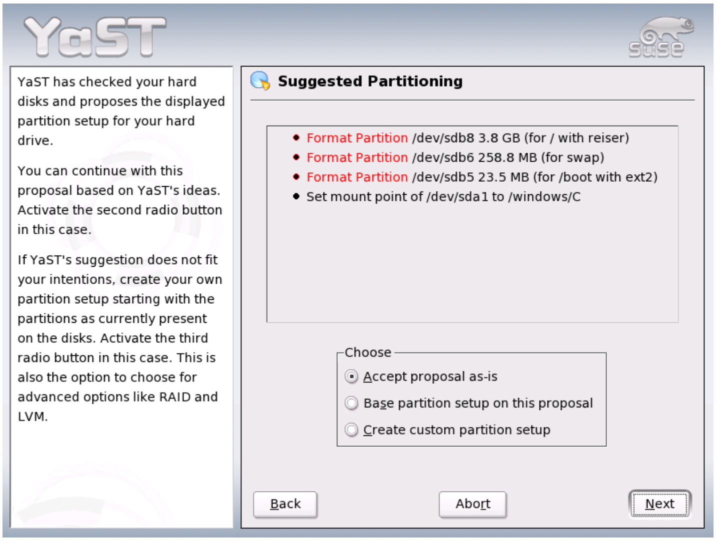
Partitioning with YaST

When you select the partitioning item in the suggestion window for the first time, YaST displays a dialog listing the partition settings as currently proposed. Accept these current settings without change or change them before continuing. Alternatively, discard all the settings and start over from scratch.

Nothing in the partitioning setup is changed if you select 'Accept Suggested Partitioning Setup'. If you select 'Change Suggested Partitioning Setup', the 'Expert Partitioner' opens. It allows tweaking the partition setup in every detail. This dialog is explained in Section 1.7.5 on page 23. The original setup as proposed by YaST is offered there as a starting point.

Selecting 'Create Custom Partitioning Setup' opens the dialog as shown in Figure 1.12 on page 23. Use the list to choose among the existing hard disks on your system. SUSE LINUX will be installed on the disk selected in this dialog.

The next step is to determine whether the entire disk should be used ('Use Entire Hard Disk') or whether to use any existing partitions (if available) for the installation. If a Windows operating system using a FAT



YaST has checked your hard disks and proposes the displayed partition setup for your hard drive.

You can continue with this proposal based on YaST's ideas. Activate the second radio button in this case.

If YaST's suggestion does not fit your intentions, create your own partition setup starting with the partitions as currently present on the disks. Activate the third radio button in this case. This is also the option to choose for advanced options like RAID and LVM.

- Format Partition /dev/sdb8 3.8 GB (for / with reiser)
- Format Partition /dev/sdb6 258.8 MB (for swap)
- Format Partition /dev/sdb5 23.5 MB (for /boot with ext2)
- Set mount point of /dev/sda1 to /windows/C

Choose
● Accept proposal as-is
○ Base partition setup on this proposal
○ Create custom partition setup

Figure 1.11: Editing the Partitioning Setup

file system was found on the disk, you may be asked whether to delete or resize the partition. Before doing so, read Section 1.7.5 on page 25. If desired, go to the 'Expert Partitioner' dialog to create a custom partition setup at this point (see Section 1.7.5 on the facing page).

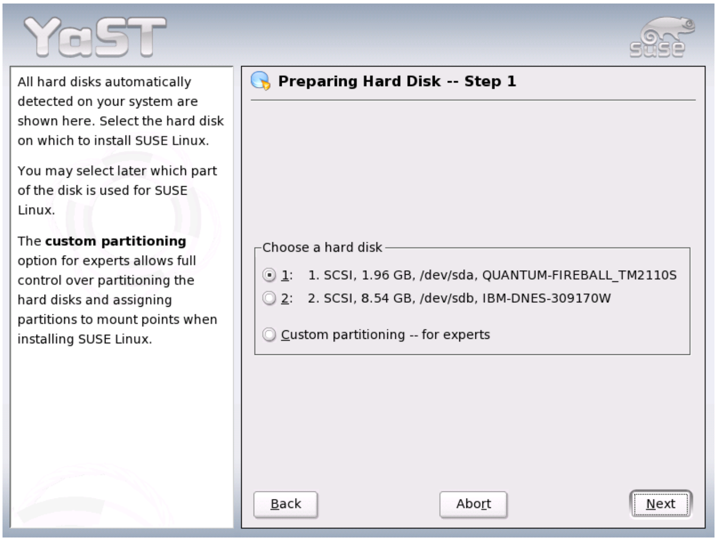
Caution

Using the Entire Hard Disk for Installation

If you choose 'Use Entire Hard Disk', all existing data on that disk is completely erased later in the installation process and is then lost.

Caution

YaST checks during the installation whether the disk space is sufficient for the software selection made. If not, YaST automatically removes parts from the software selection as needed. The suggestion window then includes a notice to inform you about this. As long as there is sufficient disk space available, YaST simply accepts your settings and partitions the hard disk accordingly.



All hard disks automatically detected on your system are shown here. Select the hard disk on which to install SUSE Linux.

You may select later which part of the disk is used for SUSE Linux.

The **custom partitioning** option for experts allows full control over partitioning the hard disks and assigning partitions to mount points when installing SUSE Linux.

Preparing Hard Disk -- Step 1

Choose a hard disk

● 1: 1. SCSI, 1.96 GB, /dev/sda, QUANTUM-FIREBALL_TM2110S
○ 2: 2. SCSI, 8.54 GB, /dev/sdb, IBM-DNES-309170W

○ Custom partitioning -- for experts

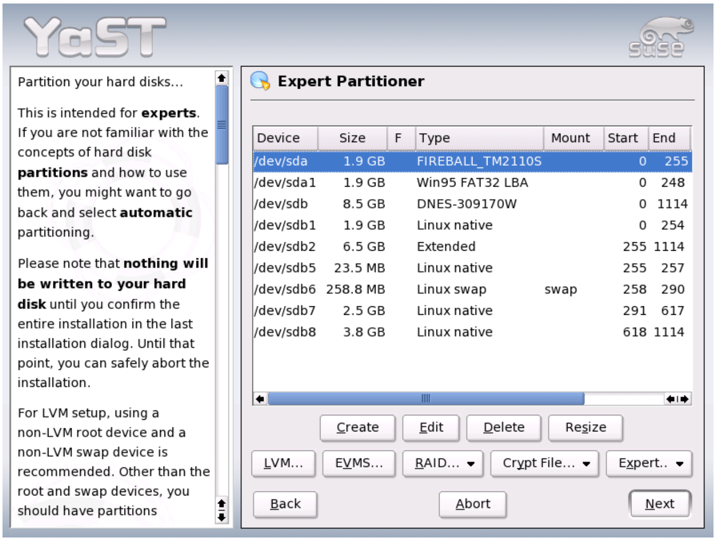
Figure 1.12: *Selecting the Hard Disk*

1.7.5 Expert Partitioning with YaST

With the expert dialog, shown in Figure 1.13 on the next page, manually modify the partitioning of your hard disk. Partitions can be added, deleted, or edited.

All existing or suggested partitions on all connected hard disks are displayed in the list of the expert dialog. Entire hard disks are listed as devices without numbers, such as `/dev/hda` or `/dev/sda`. Partitions are listed as parts of these devices, such as `/dev/hda1` or `/dev/sda1`. The size, type, file system, and mount point of the hard disks and their partitions are also displayed. The mount point describes where the partition is mounted in the Linux file system tree.

Any free hard disk space is also listed and automatically selected. To provide more disk space to Linux, 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 Linux and retain the third and first for other operating systems.



Partition your hard disks...

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

Please note that **nothing will be written to your hard disk** until you confirm the entire installation in the last installation dialog. Until that point, you can safely abort the installation.

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

Device	Size	F	Type	Mount	Start	End
/dev/sda	1.9 GB		FIREBALL_TM2110S		0	255
/dev/sda1	1.9 GB		Win95 FAT32 LBA		0	248
/dev/sdb	8.5 GB		DNES-309170W		0	1114
/dev/sdb1	1.9 GB		Linux native		0	254
/dev/sdb2	6.5 GB		Extended		255	1114
/dev/sdb5	23.5 MB		Linux native		255	257
/dev/sdb6	258.8 MB		Linux swap	swap	258	290
/dev/sdb7	2.5 GB		Linux native		291	617
/dev/sdb8	3.8 GB		Linux native		618	1114

Figure 1.13: The YaST Partitioner in Expert Mode

Creating a Partition

Select 'New'. If several hard disks are connected, a selection dialog appears in which to select a hard disk for the new partition. Then, 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 1.7.4 on page 17).

Select the file system to use to format the partition and a mount point, if necessary. YaST suggests a mount point for each partition created. Details of the parameters are provided in the next section.

Select 'OK' to apply your changes. The new partition is then listed in the partition table. If you click 'Next', the current values are adopted and you are returned to the suggestion screen.

Partitioning Parameters

If you create a new partition or modify an existing partition, various parameters can be set in the partitioning tool. For new partitions, suitable parameters are set by YaST and usually do not require any modification. To perform manual settings, proceed as follows:

1. Select the partition.
2. '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', or 'Linux RAID'.

   For details on LVM and RAID, refer to Section 3.10 on page 131 and Section 3.11 on page 139.

   File System To format the partition immediately within the scope of the installation, specify one of the following file systems for the partition: 'Swap', 'Ext2', 'Ext3', 'ReiserFS', or 'JFS'.

   File System Options Set various parameters for the selected file system here.

   Encrypt File System If you activate the encryption, all data is written to the hard disk in encrypted form.

   fstab Options Here, specify various parameters for the administration file of the file systems (`/etc/fstab`).

   Mount Point This specifies the directory at which the partition should be mounted in the file system tree. Various YaST suggestions can be expanded at the respective entry field. If you accept these suggestions, the default file system structure is implemented. However, you can also specify any other names.

3. Select 'Next' to activate the partition.

Note

s390, zseries: Further Reading

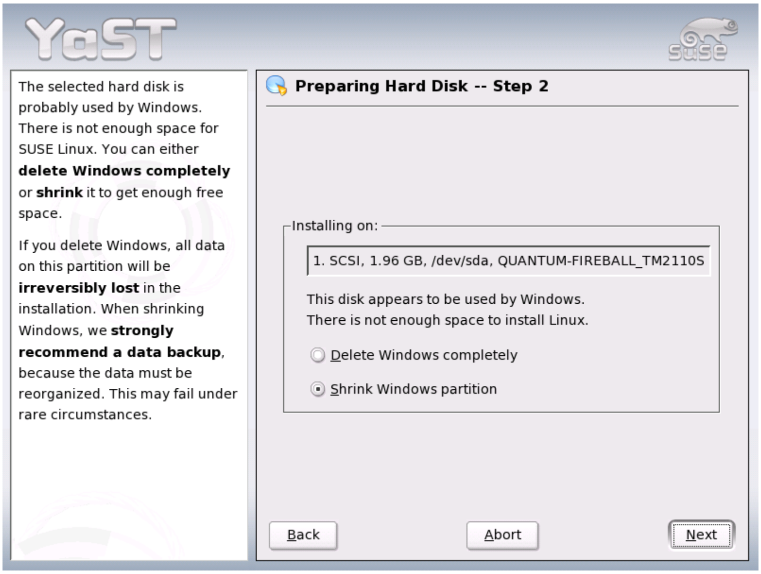
Additional relevant information for IBM S/390 and zSeries can be found in Section 1.7.5 on page 29.

Note

Resizing a Windows Partition

If a hard disk containing a Windows FAT or NTFS partition was selected as the installation target, YaST offers to delete or shrink this partition. In this way, you can install SUSE LINUX even if there is currently not enough space on the hard disk. This functionality is especially useful if the selected hard disk contains only one Windows partition that covers the entire hard disk. This is sometimes the case on computers where Windows comes preinstalled.

If YaST sees that there is not enough space on the selected hard disk, but that space could be made available by deleting or shrinking a Windows partion, it presents a dialog in which to choose one of these two options.



The selected hard disk is probably used by Windows. There is not enough space for SUSE Linux. You can either **delete Windows completely** or **shrink** it to get enough free space.

If you delete Windows, all data on this partition will be **irreversibly lost** in the installation. When shrinking Windows, we **strongly recommend a data backup**, because the data must be reorganized. This may fail under rare circumstances.

Preparing Hard Disk -- Step 2

Installing on:

1. SCSI, 1.96 GB, /dev/sda, QUANTUM-FIREBALL_TM2110S

This disk appears to be used by Windows.
There is not enough space to install Linux.

○ Delete Windows completely

● Shrink Windows partition

Figure 1.14: *Possible Options for Windows Partitions*

If you select 'Delete Windows Completely', the Windows partition is marked for deletion and the space is used for the installation of SUSE LINUX.

Caution

Deleting Windows

If you delete Windows, all data will be lost beyond recovery as soon as the formatting starts.

Caution

To shrink the Windows partition, interrupt the installation and boot Windows to prepare the partition from there. Although this step is not strictly required for FAT partitions, it speeds up the resizing process and also makes it safer. These steps are vital for NTFS partitions.

FAT File System In Windows, first run `scandisk` to make sure the FAT partition is free of lost file fragments and crosslinks. After that, run `defrag` to move files to the beginning of the partition. This accelerates the resizing procedure in Linux.

If you have optimized virtual memory settings for Windows in such a way that a contiguous swap file is used with the same initial (minimum) and maximum size limit, consider another step. With these Windows settings, the resizing might split the swap file into many small parts scattered all over the FAT partition. Also, the entire swap file would need to be moved during the resizing, which makes the process rather slow. It is therefore useful to unset these Windows optimizations for the time being and reenable them after the resizing has been completed.

NTFS File System In Windows, run `scandisk` and `defrag` to move the files to the beginning of the hard disk. In contrast to the FAT file system, you *must* perform these steps. Otherwise the NTFS partition cannot be resized.

Note

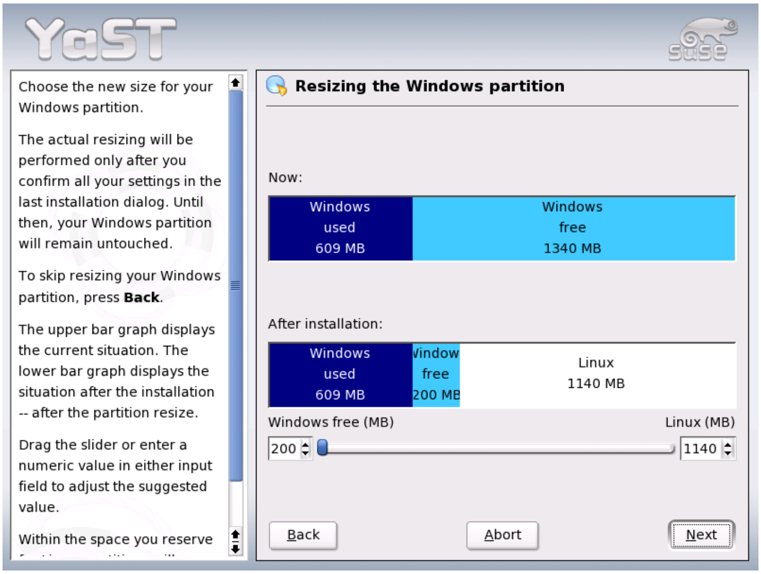
Disabling the Windows Swap File

If you operate your system with a permanent swap file on an NTFS file system, this file may be located at the end of the hard disk and remain there despite `defrag`. Therefore, it may be impossible to shrink the partition sufficiently. In this case, temporarily deactivate the swap file (the virtual memory in Windows). After the partition has been resized, reconfigure the virtual memory.

Note

After these preparations, return to the Linux partitioning setup and select 'Shrink Windows Partition'. After a quick check of the partition, YaST opens a dialog with a suggestion for resizing the Windows partition.

The first bar graph shows how much disk space is currently occupied by Windows and how much space is still available. The second bar graph shows how the space would be distributed after the resizing, according to YaST's current proposal (Figure 1.15 on the following page). Accept the proposed settings or use the slider to change the partition sizing (within certain limits).



Choose the new size for your Windows partition.

The actual resizing will be performed only after you confirm all your settings in the last installation dialog. Until then, your Windows partition will remain untouched.

To skip resizing your Windows partition, press **Back**.

The upper bar graph displays the current situation. The lower bar graph displays the situation after the installation -- after the partition resize.

Drag the slider or enter a numeric value in either input field to adjust the suggested value.

Within the space you reserve

Resizing the Windows partition

Windows free (MB): 200

Linux (MB): 1140

Figure 1.15: Resizing the Windows Partition

If you leave this dialog by selecting 'Next', the settings are stored and you are returned to the previous dialog. The actual resizing takes place later, before the hard disk is formatted.

Note

Windows Systems Installed on NTFS Partitions

By default, the Windows versions NT, 2000, and XP use the NTFS file system. Unlike FAT file systems, NTFS file systems can (currently) only be read from Linux. Therefore, you can read your Windows files from Linux, but you cannot edit them. If you want write access to your Windows data and do not need the NTFS file system, reinstall Windows on a FAT32 file system. In this case, you will have full access to your Windows data from SUSE LINUX.

Note

More Partitioning Tips

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

Example 1.1: */etc/fstab: Partition Data*

```
/dev/sda1    /data1  auto    noauto,user 0 0
/dev/sda8    /data2  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 respective 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`.

Detailed background information and tips for partitioning are provided in Section 3.9 on page 128.

1.7.6 Software

SUSE LINUX contains a number of software packages for various application purposes. As it would be burdensome to select the needed packages one by one, SUSE LINUX offers three system types with various installation scopes. Depending on the available disk space, YaST selects one of these predefined systems and displays it in the suggestion window.

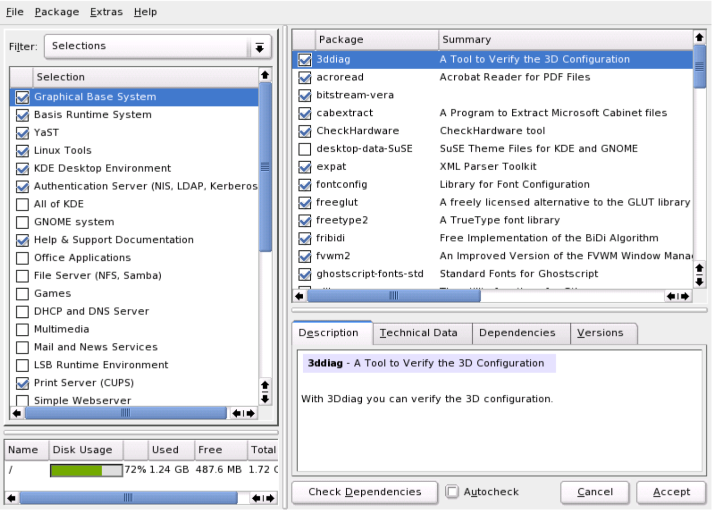
Minimal System (only recommended for special purposes)
This basically includes the core operating system with various services, but without any graphical user interface. The machine can only be operated using ASCII consoles. This system type is especially suitable for server scenarios that require little direct user interaction.

Minimal Graphical System (without KDE)
If you do not want the KDE desktop or if there is insufficient disk space, install this system type. The installed system includes the X Window System and a basic window manager. You can use all programs that have their own graphical user interface.

Default System (with KDE) This system type includes the KDE desktop together with most of the KDE programs and the CUPS print server. This is the most suitable system type for normal stand-alone machines. If possible, YaST selects this system type.

Full Installation This system type is the largest one and includes all packages coming with SUSE LINUX, except those that would result in dependency conflicts.

Click 'Software Selection' in the suggestion window to open a dialog in which to select one of the predefined systems. To start the software installation module (package manager) and modify the installation scope, click 'Detailed Selection'.See Figure 1.16.



Filter: Selections

Selection
- ☑ Graphical Base System
- ☑ Basis Runtime System
- ☑ YaST
- ☑ Linux Tools
- ☑ KDE Desktop Environment
- ☑ Authentication Server (NIS, LDAP, Kerberos
- ☐ All of KDE
- ☐ GNOME system
- ☑ Help & Support Documentation
- ☐ Office Applications
- ☐ File Server (NFS, Samba)
- ☐ Games
- ☐ DHCP and DNS Server
- ☐ Multimedia
- ☐ Mail and News Services
- ☐ LSB Runtime Environment
- ☑ Print Server (CUPS)
- ☐ Simple Webserver

Name	Disk Usage	Used	Free	Total
/	72%	1.24 GB	487.6 MB	1.72

	Package	Summary
☑	3ddiag	A Tool to Verify the 3D Configuration
☑	acroread	Acrobat Reader for PDF Files
☑	bitstream-vera	
☑	cabextract	A Program to Extract Microsoft Cabinet files
☑	CheckHardware	CheckHardware tool
☐	desktop-data-SuSE	SuSE Theme Files for KDE and GNOME
☑	expat	XML Parser Toolkit
☑	fontconfig	Library for Font Configuration
☑	freeglut	A freely licensed alternative to the GLUT library
☑	freetype2	A TrueType font library
☑	fribidi	Free Implementation of the BiDi Algorithm
☑	fvwm2	An Improved Version of the FVWM Window Mana
☑	ghostscript-fonts-std	Standard Fonts for Ghostscript

3ddiag - A Tool to Verify the 3D Configuration

With 3Ddiag you can verify the 3D configuration.

Figure 1.16: Installing and Removing Software with the YaST Package Manager

Changing the Installation Scope

If you install the default system, there is usually no need to add or remove individual packages. It consists of a software selection that meets most requirements without any changes. If you have specific needs, modify this selection with the package manager, which greatly eases this task. It offers various filter criteria to simplify selection from the numerous packages in SUSE LINUX.

The filter selection box is located at the top left under the menu bar. After starting, the active filter is ‘Selections’. This filter sorts program packages by application purpose, such as multimedia or office applications. These groups are listed under the filter selection box. The packages included in the current system type are preselected. Click the respective check boxes to select or deselect entire selections or groups for installation.

The right part of the window displays a table listing the individual packages included in the current selection. The leftmost table column shows the current status of each package. Two status flags are especially relevant for the installation: ‘Install’ (the box in front of the package name is checked) and ‘Do Not Install’ (the box is empty). To select or deselect individual software packages, click the status box until the desired status is displayed.

Alternatively, right-click the package line to access a pop-up menu listing all the possible status settings. However, most of them are not really relevant for the installation. To learn more about them, read the detailed description of this module in Section 2.3.4 on page 53.

Other Filters

Click the filter selection box to view the other possible filters. The selection according to ‘Package Groups’ can also be used for the installation. This filter sorts the program packages by subjects in a tree structure to the left. The more you expand the branches, the more specific the selection of packages is and the fewer packages are displayed in the list of associated packages to the right.

Use ‘Search’ to search for a specific package. This is explained in detail in Section 2.3.4 on page 53.

Package Dependencies and Conflicts

As with all operating systems, SUSE LINUX has certain restrictions as to which software combinations are possible and which are not. The different software packages must be compatible. Otherwise they might interfere with each other and cause conflicts that affect the system as a whole. Therefore, you may see alerts about unresolved package dependencies or conflicts after selecting or deselecting software packages in this dialog. If you install SUSE LINUX for the first time or if you do not understand the alerts, read Section 2.3.4 on page 53, which provides detailed information about the operation of the package manager and a brief summary of the software organization in Linux.

Exiting the Software Selection

When satisfied with your software selection and all package dependencies or conflicts are resolved, click 'Accept' to apply your changes and exit the module. If this module is started in the installed system, the changes are applied immediately. During the installation, however, the changes are recorded internally and applied later when the actual installation starts.

1.7.7 Boot Configuration (Boot Loader Installation)

Note

s390, zseries: Boot Loader Configuration

The module described below cannot be used to configure the boot loader (zipl) on IBM S/390 and zSeries platforms.

Note

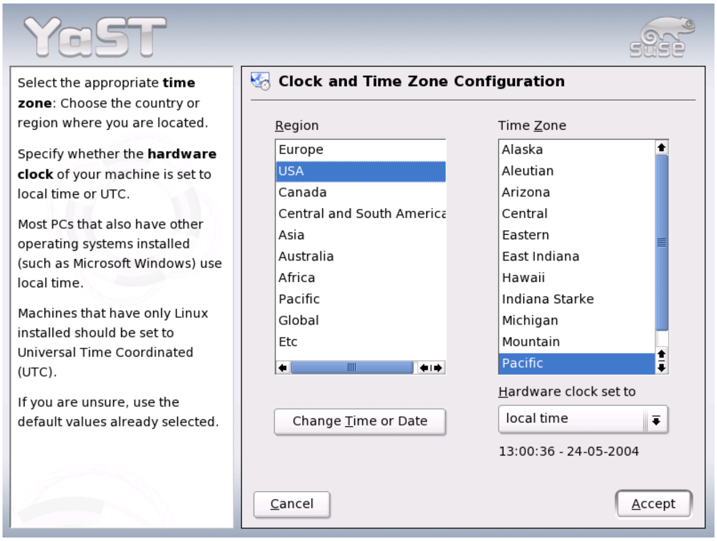
During the installation, YaST proposes a boot configuration for your system. Normally, leave these settings unchanged. However, if you need a custom setup, modify the proposal for your system.

One possibility is to configure the boot mechanism to rely on a special boot floppy. Although this has the disadvantage that it requires the floppy to be in the drive when booting, it leaves an existing boot mechanism untouched. Normally this should not be necessary, however, because YaST can configure the boot loader to boot existing operating systems as well. Another possibility with the configuration is to change the location of the boot mechanism on the hard disk.

To change the boot configuration proposed by YaST, select 'Booting' to open a dialog in which to change many details of the boot mechanism. For information, read Section 8.6 on page 218.

1.7.8 Time Zone

In this dialog, shown in Figure 1.17 on the next page, choose between `Local Time` and `UTC` under 'Hardware clock set to'. The selection depends on how the hardware (BIOS) clock is set on your machine. If it is set to `GMT`, which corresponds to UTC, your system can rely on SUSE LINUX to switch from standard time to daylight savings time and back automatically.



YaST
SuSE
Clock and Time Zone Configuration
Region
Europe
USA
Canada
Central and South America
Asia
Australia
Africa
Pacific
Global
Etc
Time Zone
Alaska
Aleutian
Arizona
Central
Eastern
East Indiana
Hawaii
Indiana Starke
Michigan
Mountain
Pacific
Change Time or Date
Cancel
Accept

Select the appropriate **time zone**: Choose the country or region where you are located.

Specify whether the **hardware clock** of your machine is set to local time or UTC.

Most PCs that also have other operating systems installed (such as Microsoft Windows) use local time.

Machines that have only Linux installed should be set to Universal Time Coordinated (UTC).

If you are unsure, use the default values already selected.

Region: USA

Time Zone: Pacific

Hardware clock set to: local time

13:00:36 - 24-05-2004

Figure 1.17: Selecting the Time Zone

1.7.9 Language

The language was already selected at the beginning of the installation (see Section 1.4 on page 12). However, you can change this setting here.

Optionally use 'Details' to set the language for the user `root`. There are three options:

ctype The value of the variable `LC_CTYPE` in the file `/etc/sysconfig/language` is adopted for the user `root`. This sets the localization for language-specific function calls.

yes The user `root` has the same language settings as the local user.

no The language settings for the user `root` are not affected by the language selection.

Click 'OK' to complete the configuration or 'Discard' to undo your changes.

1.7.10 Launching the Installation

When satisfied with the installation settings, click 'Next' in the suggestion window to begin the installation. Confirm with 'Yes' in the green dialog that opens. The installation usually takes between fifteen and thirty minutes, depending on the system performance and the software selected. 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.

1.7.11 s390, zseries: IPLing the Installed System

On IBM S/390 and zSeries platforms, another IPL must be performed after installing the selected software packages. However, the procedure varies according to the type of installation:

ESA Native and LPAR Installation
In the S/390 or zSeries terminal, select 'LOAD', select 'Clear', then enter the loading address (the device address of the root device). Now start the loading process.

z/VM Installation Shut down the installed system with the `halt` command. Log in at the VM guest, under the account name `LINUX1` and proceed to IPL the installed system:

```
IPL 151 CLEAR
```

1.7.12 s390, zseries: Connecting to the Installed System

After IPLing the installed system, establish a connection with it to complete the installation. The steps involved in this varies 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 may quit before you become aware of the message. If nothing happens during 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 the IP address of the installed system together 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 the X server from which you connect is running on the system used for the first installation phase. A YaST window opens, after which the installation is finished.

Using SSH to Connect

Note

s390, zseries: 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.

Note

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 may quit before you become aware of the message.

Now perform the following steps to complete the installation:

- Use SSH to log into the Linux system as `root` If the connection is denied or times out, wait a few minutes then try again.
- Execute the following command:

  ```
  /usr/lib/YaST2/bin/YaST2.sshinstall
  ```

  `yast` does not suffice in this case.

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

1.8 Finishing the Installation

After completing the basic system setup and the installation of all selected software packages, provide a password for the account of the system administrator (the `root` user). You can then 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. If desired, also configure a name server for centralized user administration in a local network. Finally, you can round off the installation with the configuration of the hardware devices connected to the machine.

1.8.1 root Password

root is the name of the superuser, the administrator of the system. Unlike regular users, which 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 many system files.

For verification purposes, the password for `root` must be entered twice (Figure 1.18 on the next page). You should never forget the `root` password. Once entered this password cannot be retrieved.

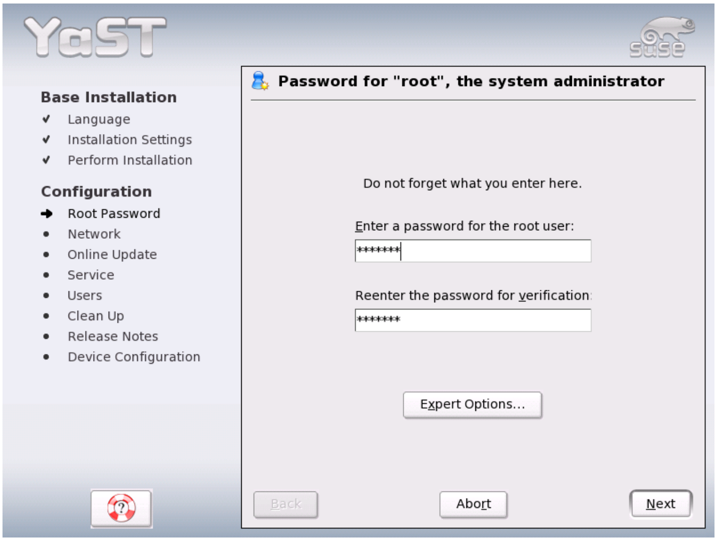
Caution

The root User

The user `root` has all the permissions needed to make changes to the system. To carry out such tasks, the `root` password is required. You cannot carry out any administrative tasks without this password.

Caution

1.8.2 Network Configuration


YaST
SUSE
Base Installation
Language
Installation Settings
Perform Installation
Configuration
Root Password
Network
Online Update
Service
Users
Clean Up
Release Notes
Device Configuration
Expert Options...
Back
Abort
Next

Password for "root", the system administrator

Do not forget what you enter here.

Enter a password for the root user: *******

Reenter the password for verification: *******

Figure 1.18: Setting the root Password

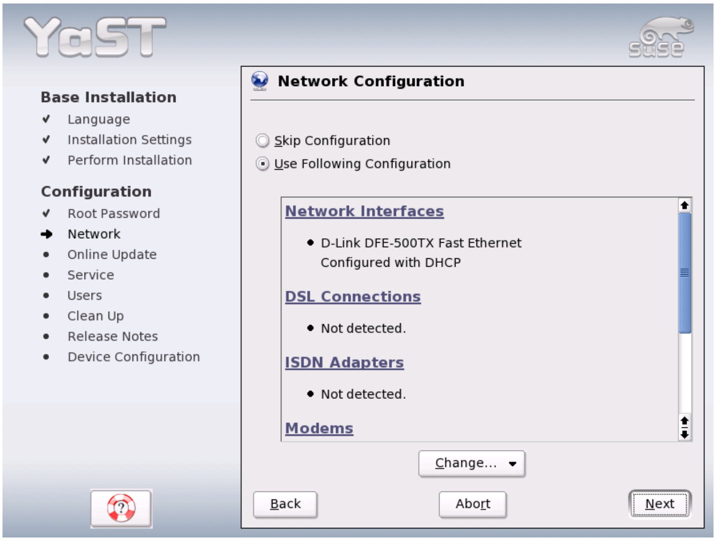
Note

s390, zseries: Network Configuration

For IBM S/390 and zSeries 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. The S/390 and zSeries platforms only support the types of network interfaces mentioned in that chapter (OSA Token Ring, OSA Ethernet, OSA Gigabit Ethernet, OSA Express Fast Ethernet, Escon, IUCV, OSA Express High-Speed Token Ring). The YaST dialog simply displays the interface with its settings as previously configured. Just confirm this dialog to continue.

Note

You can now configure the network connections of your system. If you have such devices, it is a good idea to configure them now, because an Internet connection allows YaST to retrieve any available SUSE LINUX updates and to include them in the installation.


YaST
Base Installation
✓ Language
✓ Installation Settings
✓ Perform Installation
Configuration
✓ Root Password
➔ Network
● Online Update
● Service
● Users
● Clean Up
● Release Notes
● Device Configuration
Change...
Back
Abort
Next

Network Configuration

○ Skip Configuration

● Use Following Configuration

Network Interfaces

- D-Link DFE-500TX Fast Ethernet
  Configured with DHCP

DSL Connections

- Not detected.

ISDN Adapters

- Not detected.

Modems

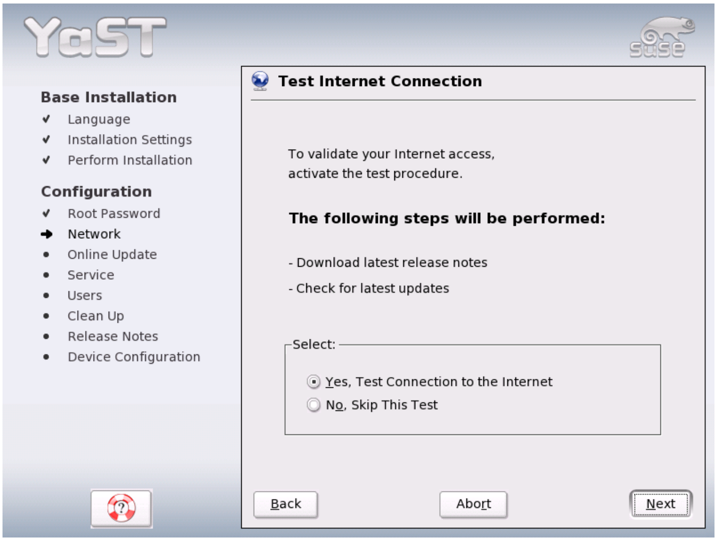
Figure 1.19: *Configuring the Network Devices*

To configure your network hardware now, refer to the relevant parts of Section 2.5 on page 84. Otherwise, select 'Skip Network Setup' and confirm with 'Continue'. The network hardware can also be configured after the system installation has been completed.

1.8.3 Testing the Internet Connection

If you have configured an Internet connection, you can test it now. For this purpose, YaST establishes a connection to the SUSE server and checks if any product updates are available for your version of SUSE LINUX. If there are such updates, they can be included in the installation. Also, the latest release notes are downloaded. You can read them at the end of the installation.

If you do not want to test the connection at this point, select 'Skip Test' then 'Next'. This also skips downloading product updates and release notes.


YaST
Base Installation
✓ Language
✓ Installation Settings
✓ Perform Installation
Configuration
✓ Root Password
➔ Network
• Online Update
• Service
• Users
• Clean Up
• Release Notes
• Device Configuration
Back
Abort
Next

Test Internet Connection

To validate your Internet access,
activate the test procedure.

The following steps will be performed:

- Download latest release notes
- Check for latest updates

Select:

● Yes, Test Connection to the Internet
○ No, Skip This Test

Figure 1.20: *Testing the Internet Connection*

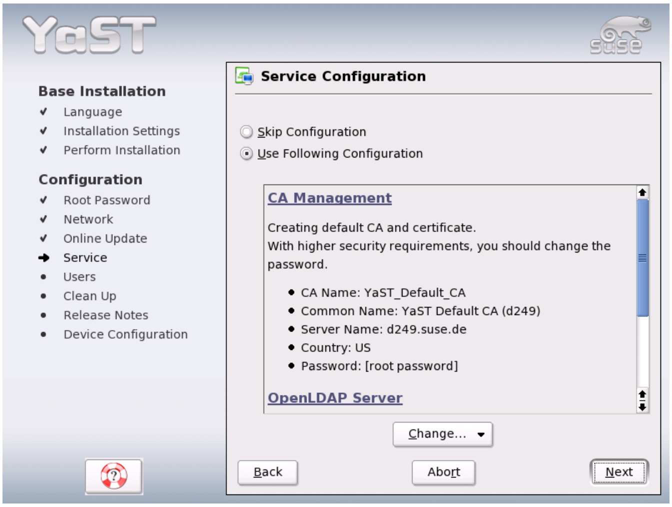
1.8.4 Loading Software Updates

If YaST was able to connect to the SUSE servers, 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.

To perform a software update immediately, select 'Perform Update Now' and click 'OK'. This opens YaST's online update dialog with a list of the available patches (if any), which can be selected and loaded. To learn about the process, read Section 2.3.2 on page 50. This kind of update can be performed at any time after the installation. If you prefer not to update now, select 'Skip Update' then click 'OK'.

1.8.5 Network Services

After testing the Internet connection and downloading the first updates, a dialog opens in which to enable and to configure two important network services (see Figure 1.21 on the following page:


YaST
Base Installation
Language
Installation Settings
Perform Installation
Configuration
Root Password
Network
Online Update
Service
Users
Clean Up
Release Notes
Device Configuration
Service Configuration

○ Skip Configuration

● Use Following Configuration

CA Management

Creating default CA and certificate.
With higher security requirements, you should change the password.

- CA Name: YaST_Default_CA
- Common Name: YaST Default CA (d249)
- Server Name: d249.suse.de
- Country: US
- Password: [root password]

OpenLDAP Server

Change...
Back
Abort
Next

Figure 1.21: 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. If you decide that you do not want to establish a CA, secure server communications on the basis of SSL and TLS, but separately for each individual service. By default, a CA is created and enabled during the installation. Details about the creation of a CA with YaST are found in Section 26.1 on page 634, together with some background information on the topic.

LDAP Server You can run an LDAP service on your host to have a central facility managing a range of configuration files. Typically, an LDAP server handles user account data, but with SUSE LINUX Enterprise Server it can also be used for mail, DHCP, and DNS related data. By default, an LDAP server is set up during the installation. If you decide against the use of an LDAP server, the YaST mail server module will not work because it depends on LDAP functionality. Nevertheless, you can still set up a mail server on your system with the help of the 'Mail Transfer Agent' module. Details about LDAP and its configuration with YaST are found in Section 21.8 on page 483.

Like the general network configuration, you may skip this configuration proposal for now. After the installation is finished, you can still configure and start the same services with the help of YaST.

1.8.6 User Authentication

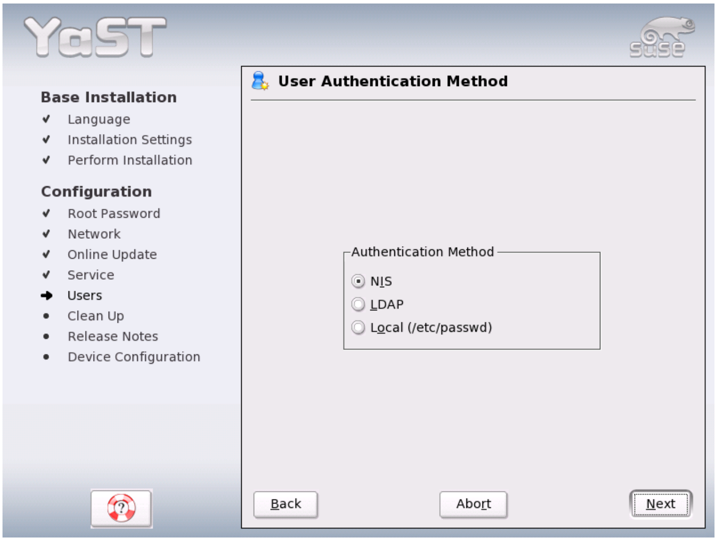
If the network access was configured successfully during the previous steps of the installation, you now have different possibilities for managing user accounts on your system.

NIS User account data is managed centrally by a NIS server.

LDAP User account data is managed centrally by an LDAP server.

Locally (/etc/passwd) This setup is used for systems where no network connection is available or where users are not supposed to log in from a remote location at all. User accounts are managed using the local `/etc/passwd` file.

If all requirements are met, YaST opens a dialog in which to select the user administration method. It is shown in Figure 1.22. If you do not have the necessary network connection, create local user accounts.


YaST
Base Installation
✓ Language
✓ Installation Settings
✓ Perform Installation
Configuration
✓ Root Password
✓ Network
✓ Online Update
✓ Service
➔ Users
• Clean Up
• Release Notes
• Device Configuration
Back
Abort
Next

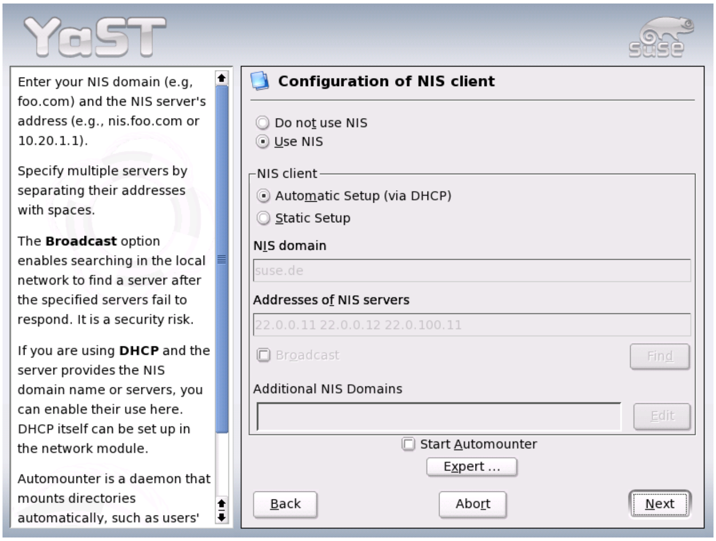
User Authentication Method

Authentication Method
● NIS
○ LDAP
○ Local (/etc/passwd)

Figure 1.22: User Authentication

1.8.7 Configuring the Host as a NIS Client

To manage user accounts through NIS, configure the host as a NIS client. To learn how to configure a NIS server with YaST, read Section 21.9 on page 513.



Enter your NIS domain (e.g, foo.com) and the NIS server's address (e.g., nis.foo.com or 10.20.1.1).

Specify multiple servers by separating their addresses with spaces.

The **Broadcast** option enables searching in the local network to find a server after the specified servers fail to respond. It is a security risk.

If you are using **DHCP** and the server provides the NIS domain name or servers, you can enable their use here. DHCP itself can be set up in the network module.

Automounter is a daemon that mounts directories automatically, such as users'

Configuration of NIS client

○ Do not use NIS
● Use NIS

NIS client
● Automatic Setup (via DHCP)
○ Static Setup

NIS domain
suse.de

Addresses of NIS servers
22.0.0.11 22.0.0.12 22.0.100.11

☐ Broadcast

Additional NIS Domains

☐ Start Automounter

Figure 1.23: NIS Client Configuration

In the following dialog, shown in Figure 1.23, first select whether the host has a fixed IP address or gets one via DHCP. If you select DHCP, you cannot specify a NIS domain or NIS server address, because these are provided by the DHCP server. For information about DHCP, read Section 21.11 on page 522. If a static IP address is used, specify the NIS domain and the NIS server manually.

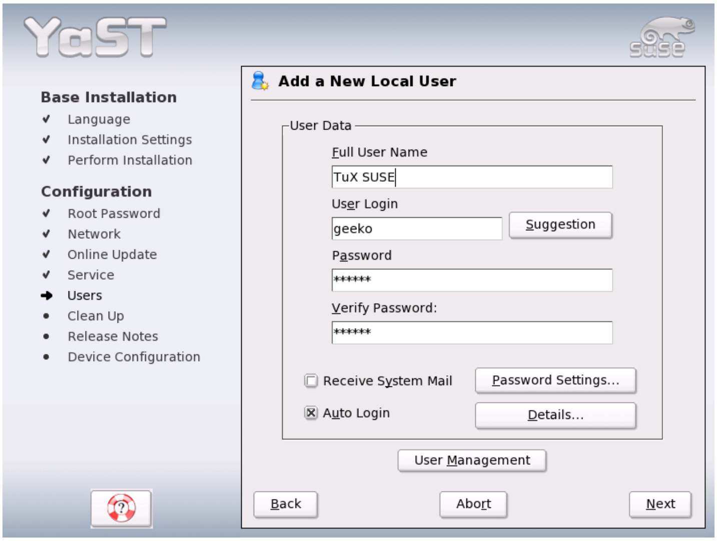
To search for NIS servers broadcasting in the network, check the relevant option. You can also specify several NIS domains and set a default domain. For each domain, select 'Edit' to specify several server addresses or enable the broadcast function on a per-domain basis.

In the expert settings, use 'Answer to the Local Host Only' to prevent other network hosts from being able to query which server your client is using. If you activate 'Broken Server', responses from servers on unprivileged ports are also accepted. For more information, refer to the man page of `ypbind`.

1.8.8 Creating Local User Accounts

If you decide against a name server for user authentication, create local users. Any data related to user accounts (name, login, password, etc.) are stored and managed on the installed system.

Linux is an operating system that allows several users to work on the same system at the same time. Each user needs a user account to log in to the system. By having user accounts, the system gains a lot in terms of security. For instance, regular users cannot change or delete files needed for the system to work properly. At the same time, the personal data of a given user cannot be modified, viewed, or tampered with by other users. Each user can set up his own working environment and always find it unchanged when logging back in.


YaST
SuSE
Base Installation
Language
Installation Settings
Perform Installation
Configuration
Root Password
Network
Online Update
Service
Users
Clean Up
Release Notes
Device Configuration
Suggestion
Password Settings...
Details...
User Management
Back
Abort
Next

Add a New Local User

User Data

Full User Name: TuX SUSE

User Login: geeko

Password: ******

Verify Password: ******

☐ Receive System Mail

☒ Auto Login

Figure 1.24: Entering the User Name and Password

A user account can be created using the dialog shown in Figure 1.24. After entering the first name and last name, specify the user name (login). Click 'Suggestion' for the system to generate a user name automatically.

Finally, enter a password for the user. Reenter it for confirmation (to ensure that you did not type something else by mistake).

To provide effective security, a password should be between five and eight characters long. The maximum length for a password is 128 characters. However, if no special security modules are loaded, only the first eight characters are used to discern the password. Passwords are case-sensitive. Special characters like umlauts are not allowed. Other special characters (7-bit ASCII) and the digits 0 to 9 are allowed.

Two additional options are available for local users:

'Receive System Messages via E-Mail'
Checking this box sends the user messages created by the system services. These are usually only sent to `root`, the system administrator.

'Automatic Login' This option is only available if KDE is used as the default desktop. It automatically logs the current user into the system when it starts. This is mainly useful if the computer is operated by only one user. For the automatic login to work, the option must be explicitly enabled.

Caution

Automatic Login

With the automatic login enabled, the system boots straight into your desktop with no authentication whatsoever. Therefore, if you store sensitive data on your system, you should *not* enable this option if the computer can also be accessed by others.

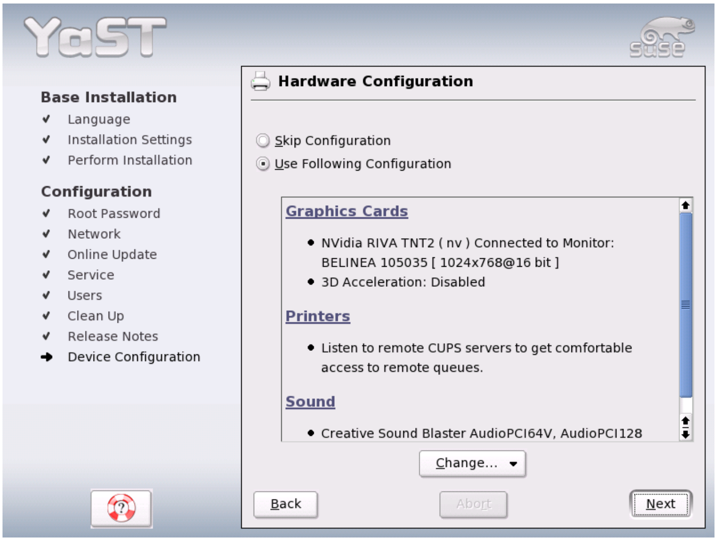
Caution

1.8.9 Reading the Release Notes

After completing the user authentication setup, YaST displays the release notes. Reading them is advised because they contain important up-to-date information that was not available when the manuals were printed. If you have installed update packages, you will be reading the most recent version of the release notes, as fetched from SUSE's servers.

1.9 Hardware Configuration

At the end of the installation, YaST opens a dialog in which to configure the graphics card and other hardware devices. Just click a component to start


YaST
SUSE
Base Installation
✓ Language
✓ Installation Settings
✓ Perform Installation
Configuration
✓ Root Password
✓ Network
✓ Online Update
✓ Service
✓ Users
✓ Clean Up
✓ Release Notes
➜ Device Configuration
Change...
Back
Abort
Next

Hardware Configuration

○ Skip Configuration
● Use Following Configuration

Graphics Cards

- NVidia RIVA TNT2 (nv) Connected to Monitor: BELINEA 105035 [1024x768@16 bit]
- 3D Acceleration: Disabled

Printers

- Listen to remote CUPS servers to get comfortable access to remote queues.

Sound

- Creative Sound Blaster AudioPCI64V, AudioPCI128

Figure 1.25: *Configuring the System Components*

its configuration. For the most part, YaST detects and configure the devices automatically.

Note

s390, zseries: Hardware Configuration

On IBM S/390 and zSeries, there is no display that would be supported by XFree. Accordingly, you will not find a 'Graphics Cards' entry on these systems.

Note

You may skip any peripheral devices and configure them later. However, you should configure the graphics card right away. Although the display settings as autoconfigured 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 'Graphics Cards'. The configuration is explained in Section 2.4.4 on page 70.

After YaST has written the configuration data, finish the installation of SUSE LINUX with 'Finish' in the final dialog.

1.10 Graphical Login

Note

s390, zseries: No Graphical Login

The graphical login is not available on IBM S/390 and zSeries platforms.

Note

SUSE LINUX is now installed. Start without logging in if you enabled the automatic login in the local user administration module. If not, you should see the graphical login on your screen, as shown in Figure 1.26. Enter a previously-defined user name and the corresponding password to log in to the system.

Figure 1.26: *The Login Screen*

2

YaST — Configuration

In SUSE LINUX Enterprise Server, YaST handles both the installation and the configuration of your system. This chapter describes the configuration of system components (hardware), network access, and security settings and administration of users. A short introduction to the text-based YaST can be found at the end of the chapter.

2.1 Starting YaST 48
2.2 The YaST Control Center 48
2.3 Software 49
2.4 Hardware 65
2.5 Network Devices 84
2.6 Network Services 84
2.7 Security and Users 88
2.8 System 93
2.9 Miscellaneous 100
2.10 YaST in Text Mode (ncurses) 101

2.1 Starting YaST

Use various dedicated YaST modules customized for specific purposes to configure a system. Depending on the underlying hardware platform, there are different ways to access YaST in the installed system.

2.1.1 Running YaST on a Graphical Desktop

If you are running KDE or GNOME, start the YaST Control Center from the SUSE menu ('System' → 'YaST'). KDE additionally integrates the individual YaST configuration modules in the KDE Control Center. A requester dialog asks for the root password, because YaST requires system administrator rights to change system files.

From a terminal program, first change to the user `root` with `sux`. Then start the graphical version of YaST with `yast2` or the text version with `yast`. Also use `yast` to start the program from the console.

2.1.2 Running from a Remote Terminal

This method is suited for hardware platforms that do not support a display device of their own, such as IBM S/390 and zSeries. It can also be employed for remote maintenance purposes.

Open a console or terminal. Enter `ssh -X root@<host to configure>` to log in as `root` on the remote system and export the output of the X server to your terminal. Once the connection is established and you have logged in with the correct password, enter `yast2` to open the graphical version of YaST on your local system. To use the text version, skip the `-X` when opening the connection and use `yast` to open the text-based YaST.

2.2 The YaST Control Center

When you start YaST in the graphical mode, the YaST Control Center, as shown in Figure 2.1 on the facing page, opens. The left frame features the categories 'Software', 'Hardware', 'Network Devices', 'Network Services', 'Security & Users', 'System', and 'Miscellaneous'. If you click the icon for one of these, its contents are listed on the right-hand side. Then select the desired element. For example, if you select 'Hardware' and click 'Sound' to the right, a configuration dialog opens for the sound card. The configuration of the individual items usually comprises several steps. Press ‘Next’ to proceed to the following step.

The left frame displays a help text for the topic, explaining the required entries. After making the needed settings, complete the procedure by pressing ‘Finish’ in the last configuration dialog. The configuration is then saved.

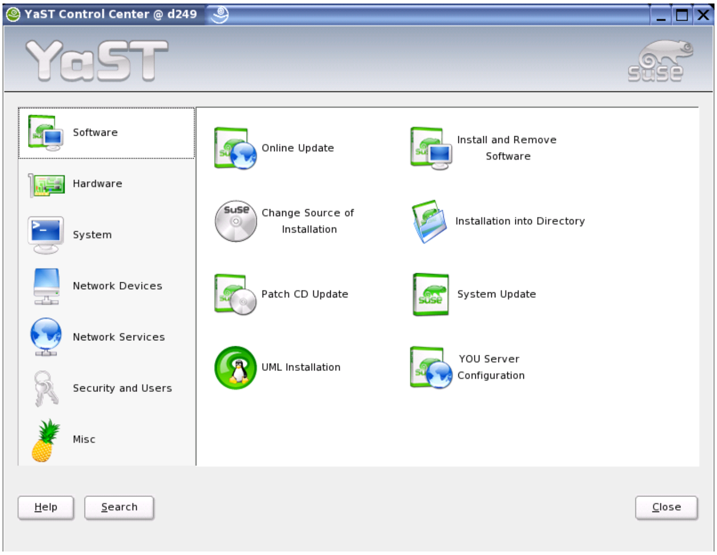

YaST Control Center @ d249
YaST
SuSE
Software
Hardware
System
Network Devices
Network Services
Security and Users
Misc
Online Update
Install and Remove Software
Change Source of Installation
Installation into Directory
Patch CD Update
System Update
UML Installation
YOU Server Configuration
Help
Search
Close

Figure 2.1: The YaST Control Center

2.3 Software

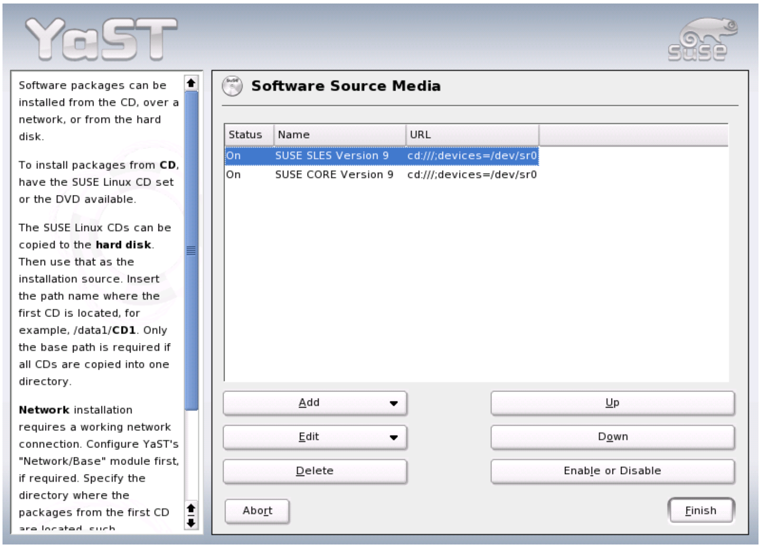
2.3.1 Change Installation Source

The installation source is the medium containing the software to install. YaST can administer a number of different installation sources. It enables their selection for installation or update purposes. For example, add the SUSE Software Development Kit CDs as installation source.

When this module is started, a list displaying all previously registered sources is displayed. Following a normal installation from CD, only the installation CD is listed. Click ‘Add’ to include additional sources in this list.

You can add removable media, such as CDs, and network servers, such as NFS and FTP. Even directories on the local hard disk can be selected as the installation medium. See the detailed YaST help text.

During the installation or update, YaST can take multiple installation sources into consideration. All registered sources have an activation status in the first column of the list. Click 'Activate or Deactivate' to activate or deactivate individual installation sources. During the installation of software packages or updates, YaST selects the suitable installation source from the range of activated installation sources. When you exit the module with 'Close', the current settings are saved and applied to the configuration modules 'Install and Remove Software' and 'System Update'.



Software packages can be installed from the CD, over a network, or from the hard disk.

To install packages from **CD**, have the SUSE Linux CD set or the DVD available.

The SUSE Linux CDs can be copied to the **hard disk**. Then use that as the installation source. Insert the path name where the first CD is located, for example, /data1/**CD1**. Only the base path is required if all CDs are copied into one directory.

Network installation requires a working network connection. Configure YaST's "Network/Base" module first, if required. Specify the directory where the packages from the first CD

Status	Name	URL
On	SUSE SLES Version 9	cd:///;devices=/dev/sr0
On	SUSE CORE Version 9	cd:///;devices=/dev/sr0

Figure 2.2: *Change Installation Source*

2.3.2 YaST Online Update

The YaST Online Update (YOU) enables the installation of important updates and improvements. The current patches for your SUSE product are available from the SUSE Maintenance Web service.

With 'Installation Source' select one of the various servers. When you select a server, its URL is copied to the input field, where it can be edited. Specify local URLs in the form `file:/my/path` or `/my/path`. Expand the existing list with additional servers using 'New Server'. Click 'Edit Server' to modify the settings of the currently selected server.

Note

Availability of a Local Update Server

If a dedicated YOU Server was installed in the local intranet using the 'YOU Server' module (see Section 4.2 on page 150), the YOU clients can be configured to poll this server instead of an external one. The configuration of the clients is described in Section 4.2.2 on page 151.

Note

When the module starts, 'Manual Selection of Patches' is active, enabling determination of whether individual patches should be fetched. To apply all available update packages, deactivate this option. However, depending on the bandwidth of the connection and the amount of data to transmit, this can result in long download times.

If you activate 'Download All Patches Again', all available patches, installable packages, and descriptions are downloaded from the server. If this is not activated (default), only retrieve patches not yet installed on your system.

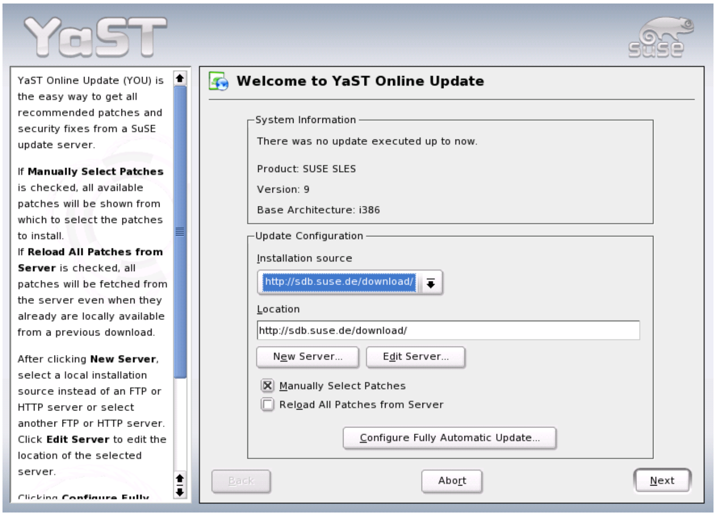
Additionally, there is a possibility to update your system automatically. Click 'Configure Fully Automatic Update' to configure a process that automatically looks for updates and applies them on a daily

basis. This procedure is fully automated and does not require any interaction. This only works if a connection to the update server, such as an Internet connection, exists at the time of the update.

To perform the update, click 'Next'. For a manual update, this loads a list of all available patches and starts the package manager, described in Section 2.3.4 on page 53. In the package manager, the filter for YOU patches is activated, enabling selection of updates to install. Patches recommended for installation are preselected. Normally, accept this suggestion.

After making your selection, click 'Accept' in the package manager. All selected updates are then downloaded from the server and installed on your machine. Depending on the connection speed and hardware performance, this may take some time. Any errors are displayed in a window. If necessary, skip the respective package. Prior to installation, some patches may open a window displaying details, allowing you to confirm the installation or skip the package.

While the updates are downloaded and installed, track actions in the log window. Following the successful installation of all patches, exit YOU with 'Finish'. If you do not need the update files after the installation, delete them with 'Remove Source Packages after Update'. Finally, SuSEconfig is executed to adjust the system configuration as needed.



YaST Online Update (YOU) is the easy way to get all recommended patches and security fixes from a SuSE update server.

If **Manually Select Patches** is checked, all available patches will be shown from which to select the patches to install.
If **Reload All Patches from Server** is checked, all patches will be fetched from the server even when they already are locally available from a previous download.

After clicking **New Server**, select a local installation source instead of an FTP or HTTP server or select another FTP or HTTP server. Click **Edit Server** to edit the location of the selected server.

Clicking **Configure Fully**

Welcome to YaST Online Update

System Information

There was no update executed up to now.

Product: SUSE SLES

Version: 9

Base Architecture: i386

Update Configuration

Installation source: http://sdb.suse.de/download/

Location: http://sdb.suse.de/download/

☒ Manually Select Patches

☐ Reload All Patches from Server

Figure 2.3: YaST Online Update

In addition to operation from the YaST interface, the YaST Online Update can also be run from the command line. The desired actions are, in this case, passed as command line parameters: `online_update [parameters]`. The available parameters are displayed in the following list along with their purpose.

`-u URL` Base URL of the directory tree from which the patches should be fetched.

`-g` Download the patches without installing them.

`-i` Install already fetched patches without downloading anything.

`-k` Check for existing new patches.

`-c` Show current configuration without further action.

`-p product` Product for which patches should be fetched.

`-v version` Product version for which patches should be fetched.

`-a architecture` Base architecture for which patches should be fetched.

`-d` "Dry run" cycle. Fetch patches and simulate installation for test purposes. The system remains unchanged.

`-n` No signature checking of the fetched files.

`-s` Display list of available patches.

`-V` Verbose mode. Print progress messages.

`-D` Debug mode for experts and for troubleshooting.

2.3.3 Patch CD Update

Patches are installed from CD instead of from an FTP server.This CD is supplied to SUSE LINUX Enterprise Server customers.

The advantage lies in a much faster update with CD. Once the Patch CD is inserted, all patches featured on the CD are scanned and displayed in the dialog. The desired packages can then be selected 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.

2.3.4 Installing and Removing Software

This module enables installation, uninstallation, and update of software on your machine. In Linux, software is available in the form of packages. Normally, a package contains everything needed for a program (such as an editor or a compiler). Usually, this includes the actual program, associated configuration files, and documentation. A package containing the source files for the respective program is normally available as well. The sources are not needed for running the program. However, you may want to install the sources to compile a custom version of the program. In Linux, this is usually possible and permitted.

Some packages depend on other packages. This means that the software of the package only works properly if another package is also installed (package dependency). Furthermore, the installation (not only the operation) of some packages is only possible if certain other packages are installed, perhaps because the installation routine needs specific tools. Accordingly, such packages must be installed in the correct sequence. There are some packages with identical or similar functionalities. If these packages use the same system resource, they should not be installed concurrently (package conflict). Dependencies and conflicts can occur between two or more packages and are sometimes very complex. The fact that a specific package version may be required for smooth interaction can make things even more complicated.

All these factors must be taken into consideration when installing, uninstalling, and updating software. YaST features a dedicated software installation tool called the package manager, which assists with this. When the package manager is started, it examines the system and displays installed packages. If you select additional packages for installation, the package manager automatically checks the dependencies and selects any other needed packages (resolution of dependencies). If you unknowingly select conflicting packages, the package manager indicates this and submits suggestions for solving the problem (resolution of conflicts). If a package needed by other installed packages is accidentally marked for deletion, the package manager issues an alert with detailed information and alternative solutions.

Apart from these purely technical aspects, the package manager provides a well-structured overview of the range of packages in SUSE LINUX. The packages are arranged by subjects and the display of these groups is restricted by means of suitable filters.

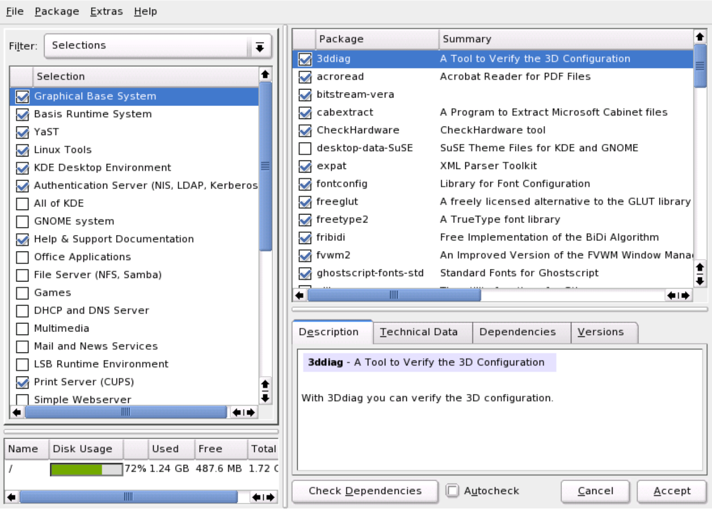
The Package Manager

To change the software selection on your system with the package manager, select 'Install or Remove Software' in the YaST Control Center. The dialog window of the package manager is shown in Figure 2.4 on the facing page.

The window comprises various frames. Modify the frame sizes by clicking and moving the lines separating the areas. The contents of the frames and their uses are described below.

The Filter Window

Because selecting the desired package from a list of all available packages is time-consuming and often difficult, the package manager offers various filter methods for arranging the packages in categories and limiting the number of displayed packages. The filter window is located to the left under the



	Selection
☑	Graphical Base System
☑	Basis Runtime System
☑	YaST
☑	Linux Tools
☑	KDE Desktop Environment
☑	Authentication Server (NIS, LDAP, Kerberos
☐	All of KDE
☐	GNOME system
☑	Help & Support Documentation
☐	Office Applications
☐	File Server (NFS, Samba)
☐	Games
☐	DHCP and DNS Server
☐	Multimedia
☐	Mail and News Services
☐	LSB Runtime Environment
☑	Print Server (CUPS)
☐	Simple Webserver

Name	Disk Usage	Used	Free	Total
/	72%	1.24 GB	487.6 MB	1.72

	Package	Summary
☑	3ddiag	A Tool to Verify the 3D Configuration
☑	acroread	Acrobat Reader for PDF Files
☑	bitstream-vera	
☑	cabextract	A Program to Extract Microsoft Cabinet files
☑	CheckHardware	CheckHardware tool
☐	desktop-data-SuSE	SuSE Theme Files for KDE and GNOME
☑	expat	XML Parser Toolkit
☑	fontconfig	Library for Font Configuration
☑	freeglut	A freely licensed alternative to the GLUT library
☑	freetype2	A TrueType font library
☑	fribidi	Free Implementation of the BiDi Algorithm
☑	fvwm2	An Improved Version of the FVWM Window Mana
☑	ghostscript-fonts-std	Standard Fonts for Ghostscript

3ddiag - A Tool to Verify the 3D Configuration

With 3Ddiag you can verify the 3D configuration.

Figure 2.4: YaST Package Manager

menu bar. It controls and displays various filter methods. The filter selection box at the top determines what will be displayed in the lower part of the filter window. Click the filter selection box to select a filter from the list of available filters.

The Selections Filter At start-up, the 'Selections' filter is active. This filter groups the program packages according to their application purpose, such as multimedia or office applications. The various groups of the 'Selections' filter are listed under the filter selection box. The packages already installed on the system are preselected. Click the status box at the beginning of a line to toggle the status flags of a selection. Select a status directly by right-clicking the selection and using the context menu. The individual package window to the right displays the list of packages included in the current selection, enabling selection and deselection of individual packages.

The Package Groups Filter The 'Package Groups' filter provides a more technical overview of the range of packages and is suitable for users familiar with the package structure of SUSE LINUX. 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 and the fewer packages are displayed in the individual package window to the right.

The filter additionally provides the possibility to display all packages in alphabetic order. To do this, select 'zzz All' in the top level. As SUSE LINUX contains a large number of packages, it may take some time to display this long list.

The Search Function The 'Search' function is the easiest way to find a specific package. By specifying various search criteria, restrict the filter so much that often only one package is displayed in the individual package window. Enter a search string and use the check boxes to determine where to search for this string (in the name, in the description, or in the package dependencies). Advanced users can even define special search patterns using wild cards and regular expressions and search the package dependencies in the 'Provides' and 'Requires' fields. For example, software developers who download source packages from the Internet can use this function to determine which package contains a specific library needed for compiling and linking this package.

Note

In addition to the 'Search' filter, all lists of the package manager feature a quick search for the current list content. 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).

Note

Installation Summary After selecting the packages for installation, update, or deletion, use the filter selection to view the installation summary. It shows what will happen with packages 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, start the package manager and 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'.

The Individual Package Window

As mentioned above, 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 'Selection' 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 means of a symbol in a status box at the beginning of the line. Toggle the status by clicking or select it from the menu that opens when the item is right-clicked. Depending on the current situation, 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 package manager offers the following package status flags:

Do Not Install This package is not installed and will not be installed.

Install This package is not yet installed but will be installed.

Keep This package is already installed and will not be changed.

Update This package is already installed and will be replaced by the version on the installation medium.

Delete This package is already installed and will be deleted.

Taboo — Never Install This package is not installed and will never be installed. It will be treated as if it does not exist on any of the installation media. If a package would automatically be selected to resolve dependencies, this can be prevented by setting the package to "Taboo." However, this may result in inconsistencies that must be resolved manually (dependency check). Thus, "Taboo" is mainly intended for expert users.

Protected This package is installed and should not be modified. Third-party packages (packages without SUSE signature) are automatically assigned this status to prevent them from being overwritten by later versions existing on the installation media. This may cause package conflicts that must be resolved manually (for experts).

Automatic Installation This package has been automatically selected for installation as it is required by another package (resolution of package dependencies).

Note

To deselect such a package, you may need to use the status "Taboo".

Note

Automatic Update This package is already installed. However, as another package requires a newer version of this package, the installed version will automatically be updated.

Delete Automatically This package is already installed, but existing package conflicts require this package be deleted. For example, this may be the case if the current package has been replaced by a different package. However, this does not happen very often.

Automatic Installation (after selection)
This packages has been automatically selected for installation because it is part of a predefined selection, such as "Multimedia" or "Development."

Automatic Update (after selection)
This package is already installed, but a newer version exists on the installation media. This package is part of a predefined selection, such as "Multimedia" or "Development," selected for update and will automatically be updated.

Delete Automatically (after selection)
This package is already installed, but a predefined selection (such as "Multimedia" or "Development") requires this package be deleted. This does not happen very often.

Additionally, decide to install or not to install the sources for a package. This information complements the current package status and cannot be toggled with the mouse or selected directly from the context menu. Instead, a check box at the end of the package line enables selection of the source packages. This option can also be accessed under 'Package'.

Install Source Also install the source code.

Do Not Install Source The sources will not be installed.

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 versions numbers are higher than those on the installation media are displayed in red. However, as 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 in the information window.

The Information Window

The tabs in the bottom right frame provide various information about the selected package. The description of the selected package is automatically active. Click the other tabs to view technical data (package size, group, etc.), the list of dependencies from other packages, or the version information.

The Resource Window

The resource window at the bottom left displays the disk space needed for your current selection of software on all currently 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.

The Menu Bar

The menu bar at the top left of the window provides access to most of the functions described above and a number of other functions that cannot be accessed in any other way. It contains the following four menus:

File Select 'File' → 'Export' to save a list of all installed packages in a text file. This is recommended if you want to replicate a specific installation scope at a later date or on another system. A file generated in this way can be imported with 'Import' and generates the same package selection as was saved. In both cases, define the location of the file or accept the suggestion.

To exit the package manager without saving changes to the package selection, click 'Exit — Discard Changes'. To save your changes, select 'Quit — Save Changes'. In this case, all changes are applied and the program is terminated.

Package The items in the 'Package' menu always refer to the package currently displayed in the individual package window. Although all status flags are displayed, you can only select those possible for the current package. Use the check boxes to determine whether to install the sources of the package. 'All in This List' opens a submenu listing all package status flags. However, these do not merely affect the current package, but all packages in this list.

Extras The 'Extras' menu offers options for handling package dependencies and conflicts. If you have already manually selected packages for installation, click 'Show Automatic Package Changes' to view the list of packages that the package manager automatically selected to resolve dependencies. If there are still unresolved package conflicts, an alert is displayed and solutions suggested.

If you set package conflicts to 'Ignore', this information is saved permanently in the system. Otherwise, you would have to set the same packages to 'Ignore' each time you start the package manager. To unignore dependencies, click 'Reset Ignored Dependency Conflicts'.

Help 'Help' → 'Overview' provides a brief explanation of the package manager functionality. A detailed description of the various package flags is available under 'Symbols'. If you prefer to operate programs without using the mouse, click 'Keys' to view a list of shortcuts.

Dependency Check

'Check Dependencies' and 'Autocheck' are located in the information window. If you click 'Check Dependencies', the package manger 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.

If you activate 'Autocheck', any change of a package status triggers an automatic check. This is a useful feature, as 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 autocheck is not activated by default. In either case, a consistency check is performed when you confirm your selection with 'Accept'.

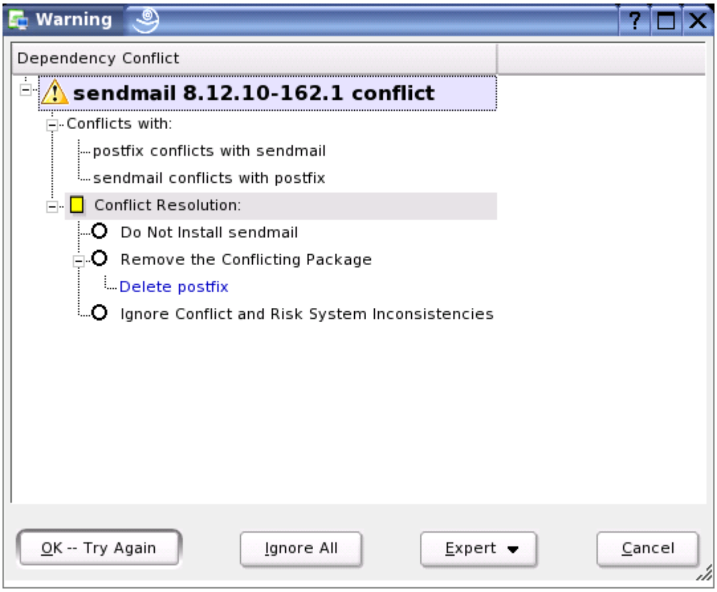
In the following example, `sendmail` and `postfix` may not be installed concurrently. Figure 2.5 on the next page shows the conflict message prompting you to make a decision. `postfix` is already installed. Accordingly, you can refrain from installing `sendmail`, remove `postfix`, or take the risk and ignore the conflict.

Caution

Handling Package Conflicts

It is advised to 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.

Caution



sendmail 8.12.10-162.1 conflict

Conflicts with:

- postfix conflicts with sendmail
- sendmail conflicts with postfix

Conflict Resolution:

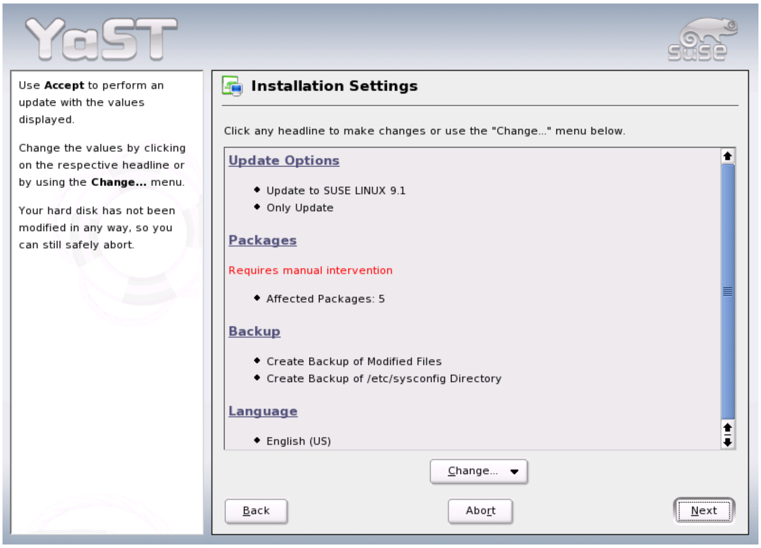
- ○ Do Not Install sendmail
- ○ Remove the Conflicting Package
  - Delete postfix
- ○ Ignore Conflict and Risk System Inconsistencies

Figure 2.5: Conflict Management of the Package Manager

2.3.5 System Update

This module enables an update of the version installed on your system. During operation, you can only update application software, not the SUSE LINUX base system. To update the base system, boot the computer from an installation medium, such as the CD. When selecting the installation mode in YaST, select 'Update an Existing System' instead of 'New Installation'.

The procedure for updating the system is similar to the new installation. Initially, YaST examines the system, determines a suitable update strategy, and presents the results in a suggestion dialog like that in Figure 2.6. Click the individual items with the mouse to change any details. Some items, such as 'Language' and 'Keyboard Layout', are covered in the section explaining the installation procedure. The following paragraphs only cover update-specific settings.



Use **Accept** to perform an update with the values displayed.

Change the values by clicking on the respective headline or by using the **Change...** menu.

Your hard disk has not been modified in any way, so you can still safely abort.

Click any headline to make changes or use the "Change..." menu below.

Update Options

- Update to SUSE LINUX 9.1
- Only Update

Packages

Requires manual intervention

- Affected Packages: 5

Backup

- Create Backup of Modified Files
- Create Backup of /etc/sysconfig Directory

Language

- English (US)

Figure 2.6: Suggestion Dialog for Updates

Selected for Update

If several versions of SUSE LINUX are installed on your system, this item enables selection of a partition for the update from the list.

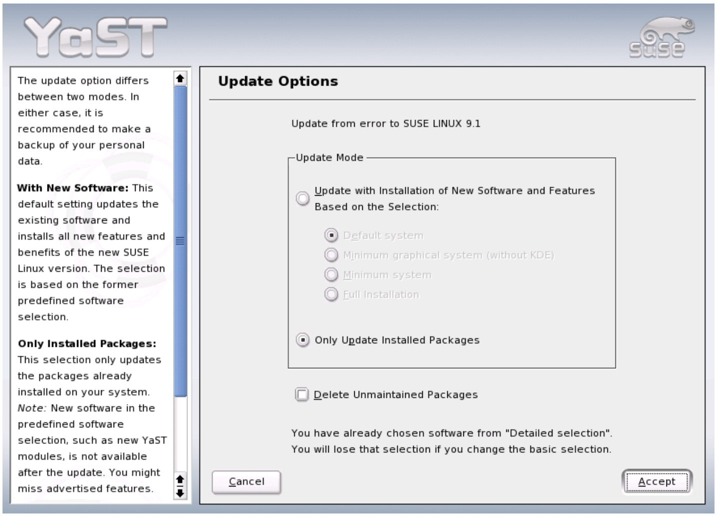
Update Options

Here, set the update method for your system. Two options are available. See Figure 2.7 on the next page.

Update with Installation of New Software
To update the entire system to the latest software versions, select one of the predefined selections. These selections are the same as those offered during the installation. They make sure new 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 will be 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.



The update option differs between two modes. In either case, it is recommended to make a backup of your personal data.

With New Software: This default setting updates the existing software and installs all new features and benefits of the new SUSE Linux version. The selection is based on the former predefined software selection.

Only Installed Packages: This selection only updates the packages already installed on your system. *Note:* New software in the predefined software selection, such as new YaST modules, is not available after the update. You might miss advertised features.

Update Options

Update from error to SUSE LINUX 9.1

Update Mode

○ Update with Installation of New Software and Features Based on the Selection:

● Default system
○ Minimum graphical system (without KDE)
○ Minimum system
○ Full Installation

● Only Update Installed Packages

☐ Delete Unmaintained Packages

You have already chosen software from "Detailed selection".
You will lose that selection if you change the basic selection.

Figure 2.7: Update Options

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 2.3.4 on page 53.

Backup

During the update, the configuration files of some packages may be replaced by those of the new version. As 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.

Note

Scope of the Backup

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

Note

Important Information about Updates

The system update is a very complex procedure. For each program package, YaST must check which version is installed on the computer and 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 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 completed.

This manual includes a chapter about updating (see Chapter 5 on page 157). All important changes from previous versions are listed, including alerts for possible update problems (see Section 5.2.1 on page 163).

2.3.6 SUSE Software Development Kit (SDK)

The SDK is a comprehensive tool kit that supports application development for SUSE Linux Enterprise Server 9 and Novell Linux Desktop. In fact, to provide you a comprehensive build system, SDK 9 includes all the open source tools that were used to build the SLES product. It provides you, as a developer, Independent Software Vendor (ISV), or Independent Hardware Vendor (IHV), all the tools you need to port applications to all of the platforms supported by the Enterprise Server and the Linux Desktop.

The SDK also contains integrated development environments (IDEs), debuggers, code editors, and other related tools. It supports most major programming languages (including C, C++, Java, and most scripting languages). For your convenience, the SDK includes multiple Perl packages that are not included in the Enterprise Server.

For detailed information, refer to `http://www.suse.com/sdk` Use the YaST package manager to install the SDK 9 software packages.

2.4 Hardware

New hardware must first be installed or connected as specified by the vendor. Turn on external devices and start the respective 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.

Note

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, as similar designations do not always indicate compatibility.

Note

2.4.1 CD-ROM Drives

Within the scope of the installation, all detected CD-ROM drives are integrated in the installed system by means of entries in the file `/etc/fstab`. The respective subdirectories are created in `/media`. Use this YaST module to integrate additional drives in the system.

When the module is started, a list of all detected drives is displayed. Mark your new drive using the check box at the beginning of the line and complete the integration with ‘Finish’. The new drive is then integrated in the system.

Note

s390, zseries: Connecting SCSI CD-ROM Drives

Connecting a SCSI CD-ROM drive to an IBM S/390 and zSeries system requires the some preparation.
First, connect the device to an FCP adapter. Then activate this device with YaST as for ZFCP hard disks, explained in Section 1.5 on page 12.
The rest of the procedure is as described above.

Note

2.4.2 Printer

A Linux system manages printers through print queues. Before any data is printed, it is sent to a print queue for temporary storage. From there, it is retrieved by a print spooler, which sends it to the printer device in the required order.

However, this data usually is not available in a form that can be processed by the printer. A graphical image, for instance, first needs to be converted into a format the printer can understand. This conversion into a printer language is achieved with a print filter, a program called by the print spooler to translate data as needed, so the printer can handle it.

Note

Further Reading

More detailed information about printing in Linux can be found in Chapter 13 on page 289.

Note

Configuration with YaST

To configure the printer, select 'Printer'. This opens the main printer configuration window in which the detected devices are listed in the upper part. The lower part lists any queues already configured. If your printer was not detected, restart the autodetection. If it fails again, select 'Configure' to configure the printer manually.

Note

In SUSE LINUX, the default printing system is CUPS, but you can switch from CUPS to LPRng and back at any time. To do so, select 'Change' → 'Advanced' in the YaST printer configuration module. Then select either printing system and configure it.

Note

Not every printer can be configured for both printing systems. Certain configurations are only supported by either CUPS or LPRng and lpdfilter. YaST informs you about this whenever necessary.

Automatic Configuration

YaST configures the printer automatically if the parallel or USB port can be set up automatically and the connected printer can be autodetected. The ID string of the printer, as supplied to YaST during hardware autodetection, must also be included in the printer database. As this ID may differ from the actual name of the model, you may need to select the model manually.

To make sure everything works properly, each configuration should be checked with the print test function of YaST. The YaST test page also provides important information about the configuration selected.

Manual Configuration

If the requirements for automatic configuration are not met or if you want a custom setup, configure the printer manually. Depending on how successful the autodetection is and how much information about the printer model is found in the database, YaST may be able to determine the right settings automatically or at least make a reasonable preselection.

The following parameters must be configured:

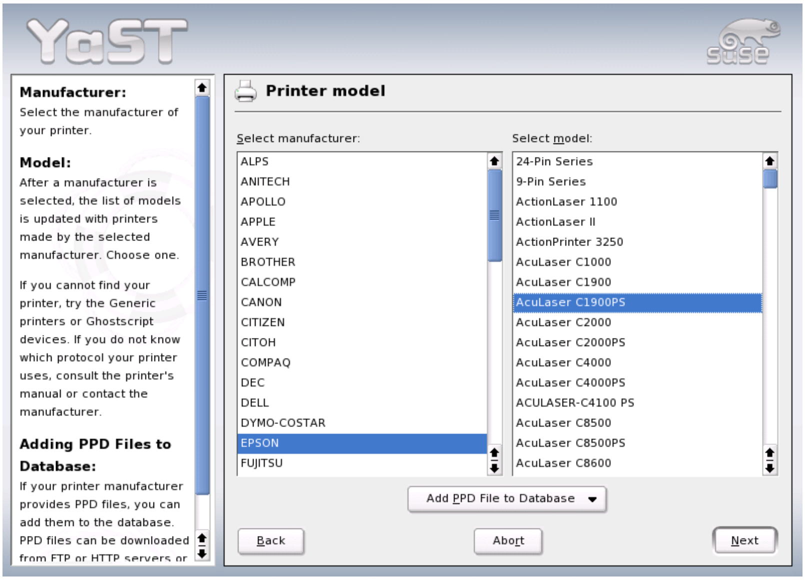
Hardware Connection (Port) The configuration of the hardware connection depends on whether YaST has been able to find the printer during hardware autodetection. If YaST is able to detect the printer model automatically, it can be assumed that the printer connection works on the hardware level and no settings need to be changed in this respect.

If YaST is unable to autodetect the printer model, there may be some problem on the hardware level and some manual intervention is required to configure the connection.

Name of the Queue The queue name is used when issuing print commands. The name should be relatively short and consist of lowercase letters and numbers only.

Ghostscript Driver or Printer Language (Printer Model)
The Ghostscript driver and the printer language depend on your printer model. YaST lets you select a predefined configuration suitable for the model. Selecting a manufacturer and a model basically means selecting a printer language and a Ghostscript driver suitable for this language with some default settings for the driver. These settings may then be changed in an additional dialog as needed.

For non-PostScript models, all printer-specific data is produced by the Ghostscript driver. The driver configuration — both choosing the right driver and the correct options for it — is the single most important factor determining the output quality. The settings made at this point affect the printout on a queue-by-queue basis.


YaST
Printer model
Select manufacturer:
ALPS
ANITECH
APOLLO
APPLE
AVERY
BROTHER
CALCOMP
CANON
CITIZEN
CITOH
COMPAQ
DEC
DELL
DYMO-COSTAR
EPSON
FUJITSU
Select model:
24-Pin Series
9-Pin Series
ActionLaser 1100
ActionLaser II
ActionPrinter 3250
AcuLaser C1000
AcuLaser C1900
AcuLaser C1900PS
AcuLaser C2000
AcuLaser C2000PS
AcuLaser C4000
AcuLaser C4000PS
ACULASER-C4100 PS
AcuLaser C8500
AcuLaser C8500PS
AcuLaser C8600
Add PPD File to Database
Back
Abort
Next

Manufacturer:
Select the manufacturer of your printer.

Model:
After a manufacturer is selected, the list of models is updated with printers made by the selected manufacturer. Choose one.

If you cannot find your printer, try the Generic printers or Ghostscript devices. If you do not know which protocol your printer uses, consult the printer's manual or contact the manufacturer.

Adding PPD Files to Database:
If your printer manufacturer provides PPD files, you can add them to the database. PPD files can be downloaded from FTP or HTTP servers or

Figure 2.8: *Selecting the Printer*

If your printer was autodetected (using the printer database) or if the model has been selected manually, YaST presents a choice of suitable Ghostscript drivers, usually with a number of predefined configurations for each of them, for instance:

- monochrome
- color 300 dpi
- photo 600 dpi

A predefined configuration includes a suitable Ghostscript driver and, if available, a set of options for the driver related to output quality. Not all selectable combinations of driver settings work with every printer model. This is especially true for higher resolutions.

If the printer model is not listed in the printer database, YaST offers a selection of standard drivers for the standard printer languages.

Advanced Settings Use this dialog to access some additional hardware-dependent (driver-specific) and hardware-independent options. If needed, change special queue settings or restrict access to the printer. However, normally there should be no need change anything here.

Troubleshooting

If there is some kind of error in the communication between the computer and the printer, the printer may no longer be able to interpret data correctly. This can cause the output to be garbled and use up large amounts of paper.

1. To stop the printing, take out all paper or open the paper trays.
2. As the print job will be removed from the printer queue only after having been sent completely to the printer, it will usually still be there. Even a reboot will not remove it. To delete the print job, use a tool like kprinter.
3. It is possible that some data is still transmitted to the printer even though the print job has been deleted from the queue. To stop all processes still using the printer device, use the `fuser` command. For a printer connected to the first parallel port, enter `fuser -k /dev/lp0`. To achieve the same for a USB printer, enter `fuser -k /dev/usb/lp0`.
4. Do a complete reset of the printer by switching it off. Wait a few seconds before putting the paper back into the trays and switching the device back on. The printer should now be ready to accept new jobs.

2.4.3 Hard Disk Controller

Normally YaST configures the hard disk controller of your system during the installation. If you add controllers, integrate these into the system with this YaST module. 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.

Caution

Configuration of the Hard Disk Controller

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

Caution

2.4.4 Graphics Card and Monitor (SaX2)

Note

S/390, zSeries: Configuring the Graphical User Interface

IBM S/390 and zSeries do not have any input and output devices supported by XFree. Therefore, none of the configuration procedures described in this section apply.
For more information relevant for IBM S/390 and zSeries, read from Section 2.5 on page 84.

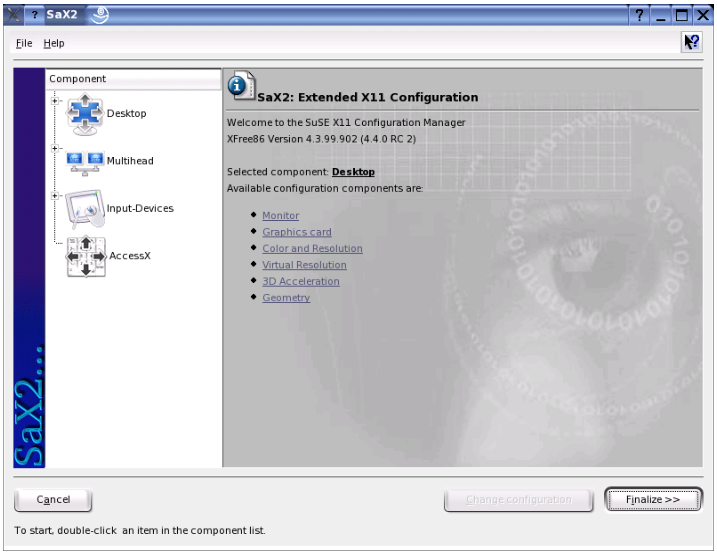
Note

The graphical user interface, or X server, handles the communication between hardware and software. Desktops, like KDE and GNOME, and the wide variety of window managers use the X server for interaction with the user.

The graphical user interface is initially configured during installation. To change the settings afterwards, run this YaST module. In the configuration dialog, choose between 'Text Mode Only' and the graphical user interface. The current settings are saved and you can reset to them at any time. The current values are displayed and offered for modification: the screen resolution, the color depth, the refresh rate, and the vendor and type of your monitor, if autodetected. If you have just installed a new graphics card, a small dialog appears asking whether to activate 3D acceleration for your graphics card.

Click 'Edit'. SaX2, the configuration tool for the input and display devices, starts in a separate window. This window is shown in Figure 2.9 on the facing page.

In the left navigation bar, there are four main items: 'Display', 'Input devices', 'Multihead', and 'AccessX'. Configure your monitor, graphics card, color depth, resolution, and the position and size of the screen under 'Display'. The keyboard, mouse, touchscreen monitor, and graphics tablet can



SaX2: Extended X11 Configuration

Welcome to the SuSE X11 Configuration Manager
XFree86 Version 4.3.99.902 (4.4.0 RC 2)

Selected component: **Desktop**
Available configuration components are:

- Monitor
- Graphics card
- Color and Resolution
- Virtual Resolution
- 3D Acceleration
- Geometry

To start, double-click an item in the component list.

Figure 2.9: *The Main Window of SaX2*

be configured under 'Input devices'. Use 'Multihead' to configure multiple screens (see Section 2.4.4 on page 76). 'AccessX' is a useful tool for controlling the mouse pointer with the number pad.

Select your monitor and graphics card. Usually, the monitor and graphics card are autodetected by the system. In this case, no manual settings are required.

If your monitor is not autodetected, automatically proceed to the monitor selection dialog. Select your monitor from the extensive list of vendors and devices or manually enter the monitor values specified in the monitor manual. Alternatively, select one of the preconfigured VESA modes.

Click 'Finish' in the main window following the completion of the settings for 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 Esc and reduce the refresh rate or the resolution and color depth. Regardless of whether you run a test, all modifications are only activated when you restart the X server.

Display

With ‘Edit configuration’ → ‘Properties’, a window with the tabs ‘Monitor’, ‘Frequencies’, and ‘Expert’ appears.

‘Monitor’ In the left part of the window, select the vendor. In the right part, select your model. If you have floppy disks with Linux drivers for your monitor, install these by clicking ‘Driver disk’.

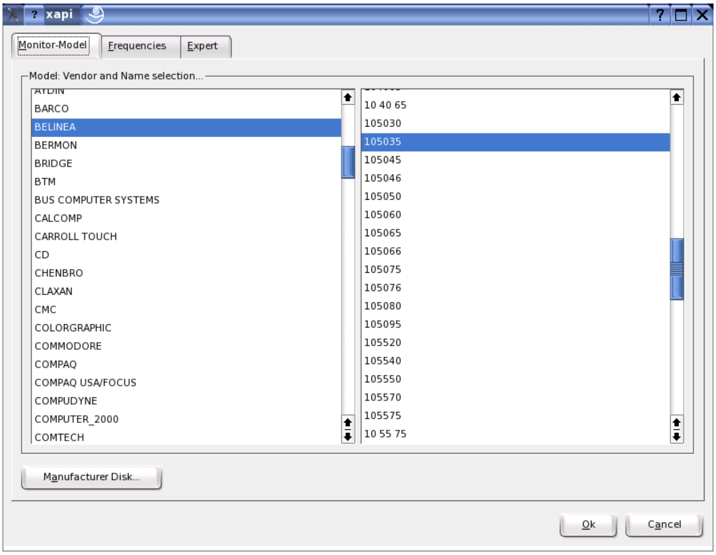

? xapi
Monitor-Model
Frequencies
Expert
Model: Vendor and Name selection...
BARCO
BELINEA
BERMON
BRIDGE
BTM
BUS COMPUTER SYSTEMS
CALCOMP
CARROLL TOUCH
CD
CHENBRO
CLAXAN
CMC
COLORGRAPHIC
COMMODORE
COMPAQ
COMPAQ USA/FOCUS
COMPUDYNE
COMPUTER_2000
COMTECH
10 40 65
105030
105035
105045
105046
105050
105060
105065
105066
105075
105076
105080
105095
105520
105540
105550
105570
105575
10 55 75
Manufacturer Disk...
Ok
Cancel

Figure 2.10: Monitor Selection

‘Frequencies’ Here, enter the horizontal and vertical frequencies for your screen. The vertical frequency is another designation for the image refresh rate. Normally, the acceptable value ranges are read from the model and entered here. Usually, they do not need to be changed.

‘Expert’ Here, enter some options for your screen. In the upper selection field, define the method to use for the calculation of the screen resolution and screen geometry. Do not change anything unless the monitor is addressed incorrectly and the display is not stable. Furthermore, you can change the size of the displayed image and activate the power saving mode DPMS.

Caution

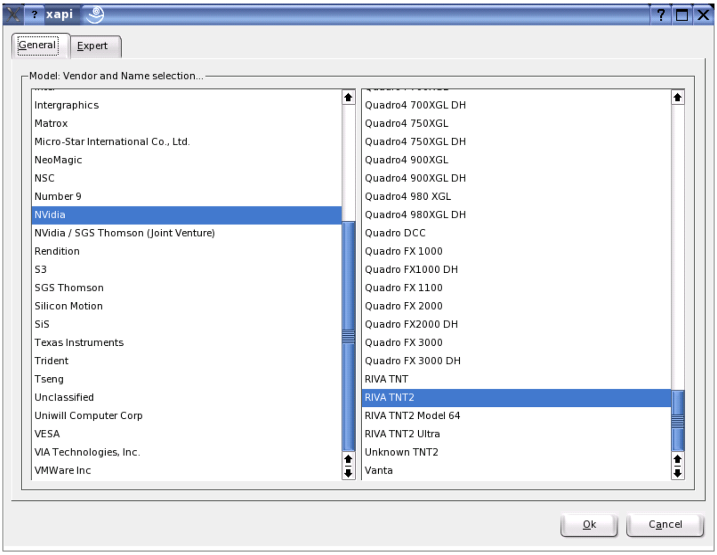
Configuring the Monitor Frequencies

There are safety mechanisms, but you should still be very careful when manually changing the allowed frequencies. False values may destroy your monitor. If in doubt, refer to the manual of the monitor.

Caution

Graphics Card

The graphics card dialog has two tabs: ‘General’ and ‘Expert’. In ‘General’, select the vendor of your graphics card on the left side and the model on the right.



Model: Vendor and Name selection...

- Intergraphics
- Matrox
- Micro-Star International Co., Ltd.
- NeoMagic
- NSC
- Number 9
- NVidia
- NVidia / SGS Thomson (Joint Venture)
- Rendition
- S3
- SGS Thomson
- Silicon Motion
- SiS
- Texas Instruments
- Trident
- Tseng
- Unclassified
- Uniwill Computer Corp
- VESA
- VIA Technologies, Inc.
- VMWare Inc

- Quadro4 700XGL DH
- Quadro4 750XGL
- Quadro4 750XGL DH
- Quadro4 900XGL
- Quadro4 900XGL DH
- Quadro4 980 XGL
- Quadro4 980XGL DH
- Quadro DCC
- Quadro FX 1000
- Quadro FX1000 DH
- Quadro FX 1100
- Quadro FX 2000
- Quadro FX2000 DH
- Quadro FX 3000
- Quadro FX 3000 DH
- RIVA TNT
- RIVA TNT2
- RIVA TNT2 Model 64
- RIVA TNT2 Ultra
- Unknown TNT2
- Vanta

Figure 2.11: Selecting the Graphics Card

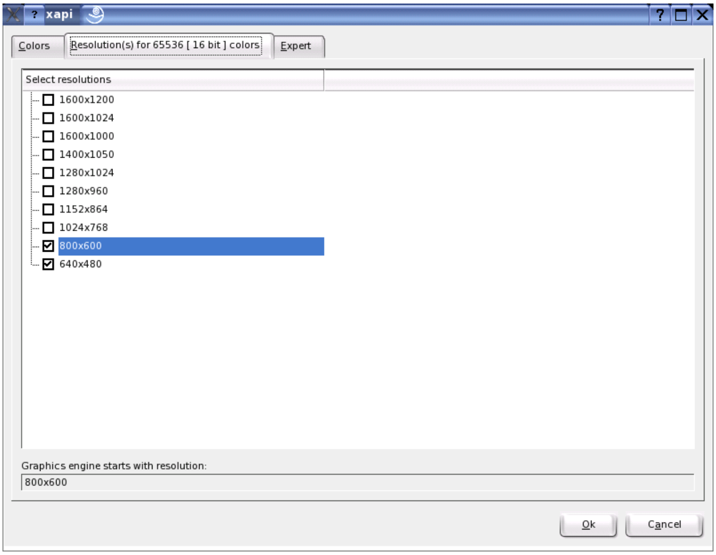
‘Expert’ offers more advanced configuration possibilities. On the right side, turn your screen to the left or to a vertical position (useful for some turnable TFT screens). The entries for the BusID are only relevant if you operate several screens. Normally, nothing needs to be changed here. You should not modify the card options unless you have experience in this field and know what the options mean. If necessary, check the documentation of your graphics card.

Colors and Resolutions

Here, three tabs, 'Colors', 'Resolution', and 'Expert', are available.

'Colors' Depending on the hardware used, select a color depth of 16, 256, 32768, 65536, or 16.7 million colors (4, 8, 15, 16, or 24 bit). For a reasonable display quality, set at least 256 colors.

'Resolution' When the hardware is detected, the resolution is queried. Therefore, the module usually only offers resolution and color depth combinations that your hardware can display correctly. This keeps the danger of damaging your hardware with incorrect settings very low in SUSE LINUX. If you change the resolution manually, consult the documentation of your hardware to make sure the value set can be displayed.



Select resolutions
- ☐ 1600x1200
- ☐ 1600x1024
- ☐ 1600x1000
- ☐ 1400x1050
- ☐ 1280x1024
- ☐ 1280x960
- ☐ 1152x864
- ☐ 1024x768
- ☑ 800x600
- ☑ 640x480

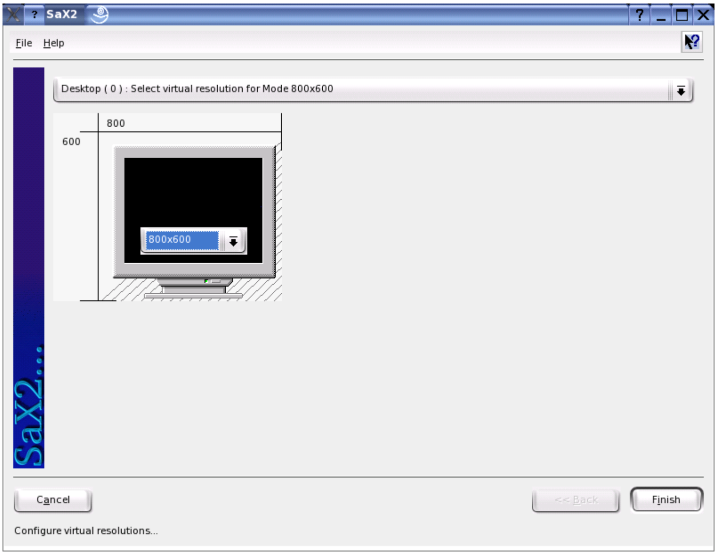
Graphics engine starts with resolution: 800x600

Figure 2.12: *Configuring the Resolution*

‘Expert’ In addition to the resolutions offered in the previous tab, this tab enables you to add your own resolutions, which will subsequently be included for selection in the tab.

Virtual Resolution

Every desktop has a certain resolution that is displayed over the full screen of the monitor. Additionally, it is possible to set the resolution larger than the visible area of the screen. If you move the mouse beyond the margins of the desktop, the virtual part of the desktop is displayed on screen. This increases the available work space.



Desktop (0) : Select virtual resolution for Mode 800x600

Configure virtual resolutions...

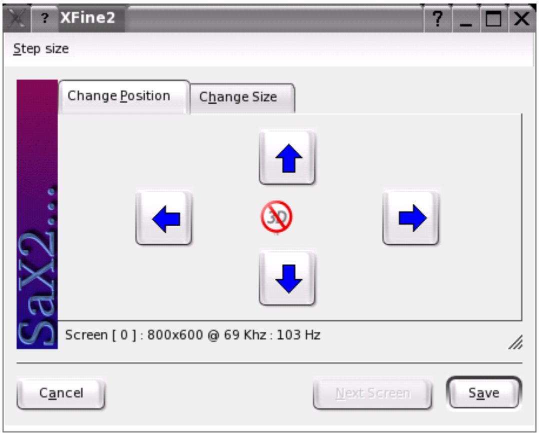
Figure 2.13: *Configuring the Virtual Resolution*

The virtual resolution can be set in two different ways. To set it using ‘By Drag&Drop’, move the mouse pointer over the monitor image so it turns into crosshairs. Keep the left mouse button pressed and move the mouse to enlarge the raster image, which corresponds with the virtual resolution. This method is best if you are not quite sure how much virtual space you want on your desktop.

For ‘By selection from the pop-up menu’, the pop-up menu in the middle of the raster image displays the currently used virtual resolution. To use one of the default virtual resolutions, select one from the menu.

Image Position and Size

Under these two tabs, precisely adjust the size and the position of the image with the arrows. See Figure 2.14. If you have a multihead environment (more than one screen), use 'Next screen' to switch to the other monitors to adjust their size and position. Press 'Save' to save your settings.



Screen [0] : 800x600 @ 69 Khz : 103 Hz

Figure 2.14: *Adjusting the Image Geometry*

Multihead

If you have installed more than one graphics card in your computer or a graphics card with multiple outputs, you can connect more than one screen to your system. If you operate two screens, this is referred to as *dualhead*. More than two is referred to as *multihead*. SaX2 automatically detects multiple graphics cards in the system and prepares the configuration accordingly. Set the multihead mode and the arrangement of the screens in the multihead dialog. Three modes are offered: 'Traditional' (default), 'One screen (Xinerama)', and 'Clone mode'.

Traditional Multihead Each monitor represents an individual unit. The mouse pointer can switch between the screens.

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.

The layout of a multihead environment describes the arrangement of and the relationship between the individual screens. 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. In the 'Layout' dialog of the multihead tool, determine the way the monitors are arranged by using the mouse to move the screen symbols in the grid. After completing the layout dialog, verify the new configuration by clicking 'Test'.

Linux currently does not offer 3D support for Xinerama multihead environments. In this case, SaX2 deactivates the 3D support.

Input Devices

Mouse If the mouse already works, you do not need to do anything. However, if the mouse does not work, control it with the number pad of the keyboard as described in Section 2.4.4 on the next page.

If the automatic detection fails, use this dialog to configure your mouse manually. Refer to the documentation of your mouse for a description of the model. Select your model from the list of supported mouse types and confirm by pressing ⑤ on the number pad.

Keyboard Use the selection field at the top of this dialog to specify the kind of keyboard to use. Then select the language for the keyboard layout (the country-specific position of the keys). Use the test field to check if special characters are displayed correctly.

The status of the check box used for activating and deactivating the entry of accented letters depends on the respective language and does not need to be changed. Click 'Finish' to apply the new settings to your system.

Touchscreen Currently, XFree86 only supports Microtouch and Elo TouchSystems touchscreens. SaX2 can only autodetect the monitor, not the toucher. The toucher is treated as an input device.

To configure the toucher, start SaX2 and select 'Input devices' → 'Touchscreens'. Click 'Add' and add a touchscreen. Save the configuration by clicking 'Finish'. You do not need to test the configuration.

Touchscreens feature a variety of options and usually must be calibrated first. Unfortunately, there is no general tool for this purpose in

Linux. The standard configuration contains suitable default values for the dimensions of the touchscreen. Normally, no additional configuration is required.

Graphics Tablet Currently, XFree86 only supports a limited number of graphics tablets. SaX2 enables the configuration of graphics tablets connected to the USB port or the serial port. From the configuration perspective, a graphics tablet is just an input device like a mouse.

Start SaX2 and select 'Input devices' → 'Graphics tablet'. Click 'Add', select the vendor from the following dialog, and add a graphics tablet from the selection list. Mark the check boxes to the right if you have connected a pen or eraser. 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. Save the configuration by clicking 'Finish'.

AccessX

If you do not use a mouse on your computer, start SaX2 and activate AccessX to be able to control the mouse pointer with the keys on the numeric keypad. (See Table 2.1).

***Table 2.1:** AccessX — Operating the Mouse with the Numeric Keypad*

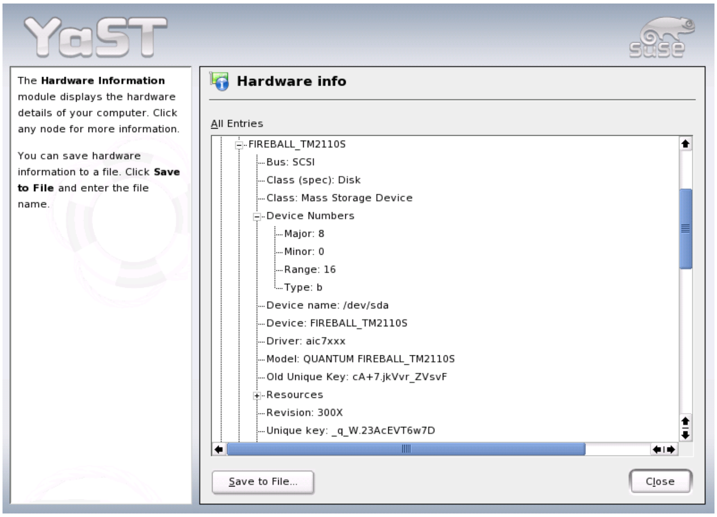
Key	description
÷	selects the left mouse button
×	selects the middle mouse button
−	selects the right mouse button
5	invokes a click event of the previously selected mouse button. The left mouse button is preset if no other button was selected. The selection is reset to its default after the event.
+	acts like 5 except is a double-click event
0	acts like 5 except is a click-and-hold event
Del	releases the click-and-hold event previously invoked with 0
7	moves the cursor toward the upper left
8	moves the cursor straight upwards

⑨	moves the cursor towards the upper right
④	moves the cursor towards the left
⑥	moves the cursor towards the right
①	moves the cursor towards the lower left
②	moves the cursor straight downwards
③	moves the cursor towards the lower right

With the slider, set the speed of the mouse pointer movement when a key is pressed.

2.4.5 Hardware Information

YaST detects hardware for the configuration of hardware components. The detected technical data is displayed in this screen. This is especially useful, for example, if you want to submit a support request for which you need information about your hardware.



The **Hardware Information** module displays the hardware details of your computer. Click any node for more information.

You can save hardware information to a file. Click **Save to File** and enter the file name.

```
FIREBALL_TM2110S
  Bus: SCSI
  Class (spec): Disk
  Class: Mass Storage Device
  Device Numbers
    Major: 8
    Minor: 0
    Range: 16
    Type: b
  Device name: /dev/sda
  Device: FIREBALL_TM2110S
  Driver: aic7xxx
  Model: QUANTUM FIREBALL_TM2110S
  Old Unique Key: cA+7.jkVvr_ZVsvF
  Resources
  Revision: 300X
  Unique key: _q_W.23AcEVT6w7D
```

Figure 2.15: Displaying Hardware Information

2.4.6 IDE DMA Mode

With this module, activate and deactivate the DMA mode for your IDE hard disks and your IDE CD and DVD drives in the installed system. 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 the installation, the current SUSE LINUX kernel automatically activates DMA for hard disks but not for CD drives, as default DMA activation for all drives often caused 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.

Note

2.4.7 Mouse

Configure your mouse with this YaST module. As the procedure for the selection of the mouse was already explained for installation, refer to Section 1.7.3 on page 16.

2.4.8 Scanner

If your scanner is connected and switched on, it should be detected automatically when this YaST module is started. In this case, the dialog for the installation of the scanner appears. If no scanner is detected, the manual configuration dialog appears. If you have already installed one or several scanners, a list of existing scanners that can be modified or deleted is displayed. Press 'Add' to configure a new device.

Next, an installation is performed with default settings. If the installation is successful, a corresponding message appears. Now, test your scanner by inserting a document and clicking 'Test'.

Scanner Not Detected

Only supported scanners can be autodetected. Scanners connected to another network host cannot be detected. The manual configuration distinguishes three types of scanners: USB scanners, SCSI scanners, and network scanners.

USB Scanner Specify the vendor and model. YaST then attempts to load USB modules. If your scanner is very new, the modules may not be loaded automatically. In this case, continue automatically to a dialog in which to load the USB module manually. Refer to the YaST help text for more information.

SCSI Scanner Specify the device (such as `/dev/sg0`). SCSI scanners should not be connected or disconnected when the the system is running. Shut the system down first.

Network Scanner Enter the IP address or the host name. To configure a network scanner, refer to the Support Database article *Scanning in Linux* (`http://sdb.suse.de/en/`, keyword *scanner*).

If your scanner was not detected, the device probably is not supported. However, sometimes even supported scanners are not detected. If that is the case, proceed with the manual scanner selection. If you can identify your scanner in the list of vendors and models, select it. If not, select 'Cancel'. Information about scanners that work with Linux is provided at `http://cdb.suse.de/index.php?LANG=en`, `http://sdb.suse.de/en/`, and `http://www.mostang.com/sane`.

Caution

Only assign the scanner manually if you are absolutely sure. Incorrect selection could damage your hardware.

Caution

Troubleshooting

Your scanner may not have been detected for one of the following reasons:

- The scanner is not supported. Check `http://sdb.suse.de/en/` for a list of Linux-compatible devices.
- Your SCSI controller was not installed correctly.

- There are termination problems with your SCSI port.
- Your SCSI cable is too long.
- Your scanner has a SCSI light controller that is not supported by Linux.
- Your scanner is defective.

Caution

SCSI scanners should not be connected or disconnected when the the system is running. Shut the system down first.

Caution

2.4.9 Sound

When the sound configuration tool is started, YaST tries to detect your sound card automatically. Configure one or multiple sound cards. To use multiple sound cards, start by selecting one of the cards to configure. Press 'Configure' to enter the 'Setup' dialog. 'Edit' opens a dialog in which to edit previously configured sound cards. 'Finish' saves the current settings and completes the sound configuration. If YaST is unable to detect your sound card automatically, press 'Add Sound Card' in 'Sound Configuration' to open a dialog in which to select a sound card and module.

Setup

With '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. With 'Normal Setup', you have the possibility to adjust the output volume and play a test sound. With 'Options', manually customize the sound card options.

Set up your joystick by clicking the respective check box. Select the joystick type in the following dialog and click 'Next'. The same dialog appears when you click 'Joystick' in the YaST Control Center.

Sound Card Volume

Test your sound configuration in this test screen. Use '+' and '–' to adjust the volume. Start at about ten percent to avoid damage to your speakers or hearing. A test sound should be audible when you press 'Test'. If you cannot hear anything, increase the volume. Press 'Continue' to complete the sound configuration. The volume setting will be saved.

Sound Configuration

Use 'Delete' to remove a sound card. Existing entries of configured sound cards are deactivated in the file `/etc/modprobe.d/sound`. Click 'Options' to open a dialog in which to customize the sound module options manually. In 'Volume', configure the individual settings for the input and output of each sound card. 'Next' saves the new values and 'Back' restores the default configuration. Under 'Add Sound Card...', configure additional sound cards. If YaST detects another sound card, continue to 'Configure a Sound Card'. If YaST does not detect a sound card, automatically be directed to 'Manual Sound Card Selection'.

If you use a `Creative Soundblaster Live` or `AWE` sound card, automatically copy SF2 sound fonts to your hard disk from the original Soundblaster driver CD-ROM with 'Install Sound Fonts'. The sound fonts are saved in the directory `/usr/share/sfbank/creative/`.

Enable or disable the start-up of ALSA when booting the machine with 'Start ALSA'. For playback of MIDI files, activate 'Start Sequencer'. This way, the sound modules required for sequencer support are loaded along with the ALSA modules.

The volume and configuration of all sound cards installed are saved when you click 'Finish'. The mixer settings are saved to the file `/etc/asound.conf` and the ALSA configuration data is appended at the end of the file `/etc/modules.conf`.

Configuring a Sound Card

If multiple sound cards were detected, select your preferred card under 'List of Automatically Recognized...'. Continue to 'Setup' with 'Next'. If the sound card was not automatically detected, click 'Select from List' and, with 'Next', proceed to 'Manual Sound Card Selection'.

Manual Sound Card Selection

If your sound card was not automatically detected, a list of sound card drivers and models are shown from which to choose. With 'All', see the entire list of supported cards.

Refer to your sound card documentation for the information required. A reference list of sound cards supported by ALSA with their corresponding sound modules is available in `/usr/share/doc/packages/alsa/cards.txt` and at `http://www.alsa-project.org/~goemon/`. After making your selection, click 'Next' to return to 'Setup'.

2.5 Network Devices

A description for configuring any supported types of network adapters in YaST including background information about connecting to networks is provided in Section 21.4 on page 437.

2.6 Network Services

2.6.1 DHCP Server

YaST can set up a custom DHCP server in only a few steps. Chapter 21.11 on page 522 provides basic knowledge about the subject as well as a step-by-step description of the configuration process in YaST.

2.6.2 Host Name and DNS

Use this module for configuration of host name and DNS, if these settings were not already been made while configuring the network devices.

2.6.3 NFS Client and NFS Server

NFS enables you to operate a file server that can be accessed by members of your network. On this file server, you can make programs, files, or storage space available for users. Use the 'NFS Server' module to set up your computer as an NFS server and to determine the directories to export for use by the network users.

Subsequently, any user (holding the required permissions) can mount these directories in his own file tree. The description of the YaST module and background information about NFS is provided in Section 21.10 on page 518.

2.6.4 Configuration of a Samba Server

Set up a Samba server to share resources such as files or printers with Windows hosts. In the first dialog, define the role of the Samba server. You can deactivate it, use it as a file and print server, or use it as a backup or primary domain controller. A file and print server makes directories and printers available. A domain controller enables its clients to log in to a Windows domain. The primary domain controller manages users and passwords. A backup domain controller uses another domain controller for authenticating the users. More information about Samba is available in Section 24.1 on page 588.

2.6.5 Configuration of Samba Clients

Configure a Samba client to access resources (files or printers) on the Samba server. In the 'Samba Workgroup' dialog, enter the domain or workgroup. Use 'Browse' to display all available groups and domains and select one of them with a mouse click. If you activate 'Also Use SMB Information for Linux Authentication', user authentication is conducted via the Samba server. After specifying all settings, click 'Finish' to complete the configuration.

2.6.6 NTP Client

NTP (Network Time Protocol) is a protocol for synchronizing the clocks of network hosts. In the respective YaST module, select a type with 'Add'. Several options are then displayed. 'Server' and 'Radio clock' are the most frequently-used options. 'Radio clock' requires special hardware.

If you select 'Server', enter the address of an NTP server when prompted. You can enter one of the public NTP servers listed at `http://www.eecis.udel.edu/~mills/ntp/servers.html`. Confirm with 'OK'.

To start the NTP daemon when the system is booted, select 'When booting system'. Save your settings with 'Finish'. More information about NTP is provided in Section 21.12 on page 535.

2.6.7 Routing

More information about routing is provided in Section 21.5 on page 450.

2.6.8 Mail Transfer Agent

This module configures your mail settings 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 server or IMAP server of your provider.

You can also use a mail program of your choice, such as KMail or Evolution, to set your POP and SMTP access data as usual (to receive mail with POP3 and send mail with SMTP). In this case, you do not need this module.

Connection Type

To configure your mail with YaST, specify the desired type of connection to the Internet in the first dialog of the e-mail configuration module. 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.

Furthermore, you can activate virus scanning for your incoming and outgoing e-mail with AMaViS by activating the respective check box. 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. If you use a dial-up connection, specify diverse POP or IMAP servers for mail reception by various users. By means of this dialog, you can also assign aliases, use masquerading, or set up virtual domains. Click 'Finish' to exit the mail configuration.

2.6.9 Mail Server

Note

LDAP-Based Mail Server Configuration

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

Note

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

Global Configuration Configures the identification of the local mail server as well as the maximum size of incoming or outgoing messages and the type of mail transport.

Local Delivery Configuration Configures the type of local mail delivery.

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

Basic UCE Configuration Configures the SPAM protection settings of the mail server. This activates the virus detection tool AMaViS after setting the type and strictness of the SPAM checking up to completely blocking acceptance of mail from certain hosts or clients.

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

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

Local Domain Configuration 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 different 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 those users with an explicit adress within a virtual domain receive mail. Virtual mail addresses are set up in the user management module of YaST.

2.6.10 Network Services (inetd)

This tool allows you to determine which network services (such as telnet, finger, talk, and ftp) should start when SUSE LINUX boots. These services enable external hosts to connect to your computer. You can also configure various parameters for each service. By default, the master service that manages the individual services (inetd or xinetd) is not started.

When this module starts, choose which of the two services to configure. The selected daemon can be started with a standard selection of network services. If desired, 'Add', 'Delete', or 'Edit' services to compose your own selection of services.

Caution

Configuration of Network Services (inetd)

The deployment and adjustment of network services on a system is a complex procedure which should be only undertaken by experts. The functionality of the system could be endangered otherwise.

Caution

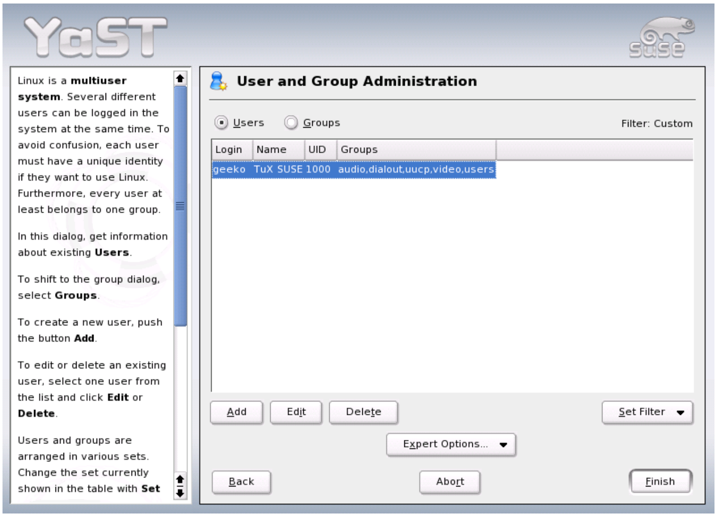
2.7 Security and Users

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.

2.7.1 User Administration

After you select to edit users, YaST provides an overview of all local users in the system. If you are part of an extensive network, click 'Set Filter' to list all system users (e.g., `root`) or NIS users. You can also create user-defined filter settings. Instead of switching between individual user groups, combine them according to your needs. To add new users, fill in the required blanks in the following screen. Subsequently, the new user can log in to the host with the login name and password. The user profile can be fine-tuned with 'Details'. You can manually set the user ID, the home directory, and the default login shell. Assign the new user to specific groups. Configure the validity of the password in 'Password Settings'. Click 'Edit' to change these settings whenever necessary. To delete a user, select the user from the list and click 'Delete'.

For advanced network administration, use 'Expert Options' to define the default settings for the creation of new users. Select the authentication method (NIS, LDAP, Kerberos, or Samba) and the algorithm for the password encryption. These settings are relevant for large (corporate) networks.



Linux is a **multiuser system**. Several different users can be logged in the system at the same time. To avoid confusion, each user must have a unique identity if they want to use Linux. Furthermore, every user at least belongs to one group.

In this dialog, get information about existing **Users**.

To shift to the group dialog, select **Groups**.

To create a new user, push the button **Add**.

To edit or delete an existing user, select one user from the list and click **Edit** or **Delete**.

Users and groups are arranged in various sets. Change the set currently shown in the table with **Set**

User and Group Administration

● Users ○ Groups

Filter: Custom

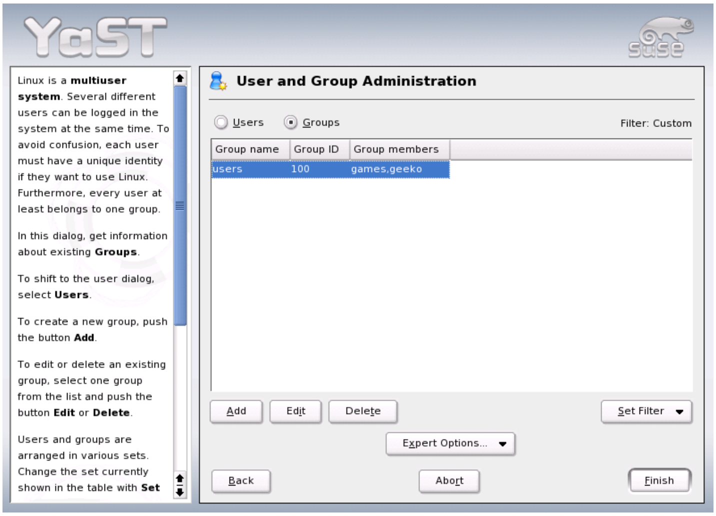
Login	Name	UID	Groups
geeko	TuX SUSE	1000	audio,dialout,uucp,video,users

Figure 2.16: User Administration

2.7.2 Group Administration

Start the group administration module from the YaST Control Center or click 'Groups' in the user administration. Both dialogs have the same functionality, allowing you to create, edit, or delete groups.

YaST provides a list of all groups. To delete a group, select it from the list (the line will be highlighted dark blue) and click 'Delete'. Under 'Add' and 'Edit', enter the name, group ID (gid), and members of the group in the respective YaST screen. If desired, set a password for the change to this group. The filter settings are the same as in the 'User Administration' dialog.



Linux is a **multiuser system**. Several different users can be logged in the system at the same time. To avoid confusion, each user must have a unique identity if they want to use Linux. Furthermore, every user at least belongs to one group.

In this dialog, get information about existing **Groups**.

To shift to the user dialog, select **Users**.

To create a new group, push the button **Add**.

To edit or delete an existing group, select one group from the list and push the button **Edit** or **Delete**.

Users and groups are arranged in various sets. Change the set currently shown in the table with **Set**

○ Users ● Groups

Filter: Custom

Group name	Group ID	Group members
users	100	games,geeko

Figure 2.17: *Group Administration*

2.7.3 Security Settings

In 'Local Security Configuration', which can be accessed under 'Security&Users', select one of the following four options: Level 1 is for standalone computers (preconfigured). Level 2 is for workstations with a network (preconfigured). Level 3 is for a server with a network (preconfigured). Use 'Custom Settings' for your own configuration.

If you click one of the first three items, incorporate one of the levels of preconfigured system security options. To do this, simply click 'Finish'. Under 'Details', access the individual settings that can be modified. If you choose 'Custom settings', proceed to the different dialogs with 'Next'. Here, find the default installation values.

'Password Settings' For new passwords to be checked by the system before they are accepted, mark 'Checking new passwords' and 'Plausibility test for password'. Set the minimum and maximum length of passwords 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' Specify how the key combination Ctrl-Alt-Del should be interpreted by selecting the action from the drop-down list.

Usually, this combination, 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.

Specify the 'Shutdown Behavior of KDM' by granting permission to shut down the system from the KDE Display Manager, the graphical login of KDE. 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 via 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 failed login attempts' and '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`. With 'Allow remote graphical login', other users are granted access to your graphical login screen via the network. However, as this access possibility represents a potential security risk, it is inactive by default.

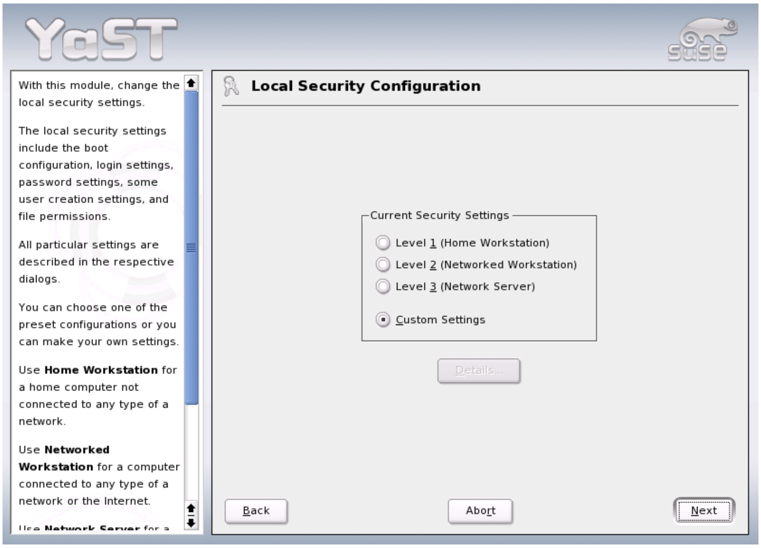
'Add User Settings' Every user has a numerical and an alphabetical user ID. The correlation between these is established via the file `/etc/passwd` and should be as unique as possible.

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. Proceed in the same way with the group ID settings.

'Miscellaneous Settings' For 'Setting of file permissions', there are three selection options: 'Easy', 'Secure', and 'Paranoid'. The first one should be sufficient for most users. The YaST help text provides information about the three security levels.

The setting 'Paranoid' is extremely restrictive and can serve as the basic level of operation for system administrator settings. If you select 'Paranoid', remember that some programs might not work or not work correctly, because users no longer have the permissions to access certain files. In this dialog, also define which user should start the `updatedb` program. 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 (locatedb can be searched by running the `locate` command). 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. Finally, make sure the option 'Current directory in root's path' is deactivated (default).

Press 'Finish' to complete your security configuration.



With this module, change the local security settings.

The local security settings include the boot configuration, login settings, password settings, some user creation settings, and file permissions.

All particular settings are described in the respective dialogs.

You can choose one of the preset configurations or you can make your own settings.

Use **Home Workstation** for a home computer not connected to any type of a network.

Use **Networked Workstation** for a computer connected to any type of a network or the Internet.

Use **Network Server** for a

Local Security Configuration

Current Security Settings

○ Level 1 (Home Workstation)
○ Level 2 (Networked Workstation)
○ Level 3 (Network Server)
● Custom Settings

Figure 2.18: *Security Settings*

2.7.4 Firewall

Use this module to configure SuSEfirewall2 to protect your machine against attacks from the Internet.Detailed information about SuSEfirewall2 can be found in Section 26.3 on page 657. The module consists of four dialogs. In the first dialog, select the interface to protect. 'External interface' selects the interface connected to the Internet. List multiple interfaces separated by spaces. 'Internal interface' is only required if you are located in an internal network and intend to use the firewall to protect your computer against internal attacks. In this case, your computer would be in a *demilitarized zone* (DMZ). Normally, a configuration with DMZ is only used for company networks.

2.8 System

Note

s390, zseries: Additional Steps.

For IBM S/390 and zSeries, continue with Section 2.8.5 on page 97.

Note

2.8.1 Backup Copy of the System Areas

The YaST backup module enables you to create a backup of your system. The backup created by the module does not comprise the entire system, but only saves information about changed packages and copies of critical storage areas and configuration files.

Define the kind of data to save in the backup. By default, the backup includes information about any packages changed since the last installation. In addition, it may include data that does not belong to packages themselves, such as many of the configuration files in `/etc` or the directories under `/home`. Apart from that, the backup can include important storage areas on your hard disk that may be crucial when trying to restore a system, such as the partition table or the master boot record (MBR).

2.8.2 Restoring the System

The restore module, shown in Figure 2.19 on the following page, enables restoration of your system from a backup archive. Follow the instructions in YaST. Press ‘Next’ to proceed to the individual dialogs. First, specify where the archives are located (removable media, local hard disks, or network file systems). A description and the contents of the individual archives are displayed, enabling you to decide what to restore from the archives.

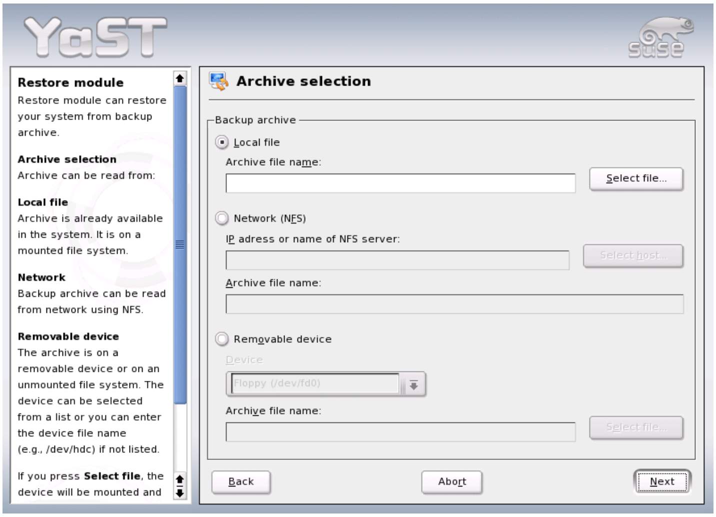
Additionally, there are two dialogs for uninstalling packages that were added since the last backup and for the reinstallation of packages that were deleted since the last backup. These two steps enable you to restore the exact system state at the time of the last backup.

Caution

System Restoration

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

Caution


YaST
Archive selection
Backup archive
Archive file name:
Select file...
IP adress or name of NFS server:
Select host...
Archive file name:
Device
Floppy (/dev/fd0)
Archive file name:
Select file...
Back
Abort
Next

Restore module
Restore module can restore your system from backup archive.

Archive selection
Archive can be read from:

Local file
Archive is already available in the system. It is on a mounted file system.

Network
Backup archive can be read from network using NFS.

Removable device
The archive is on a removable device or on an unmounted file system. The device can be selected from a list or you can enter the device file name (e.g., /dev/hdc) if not listed.

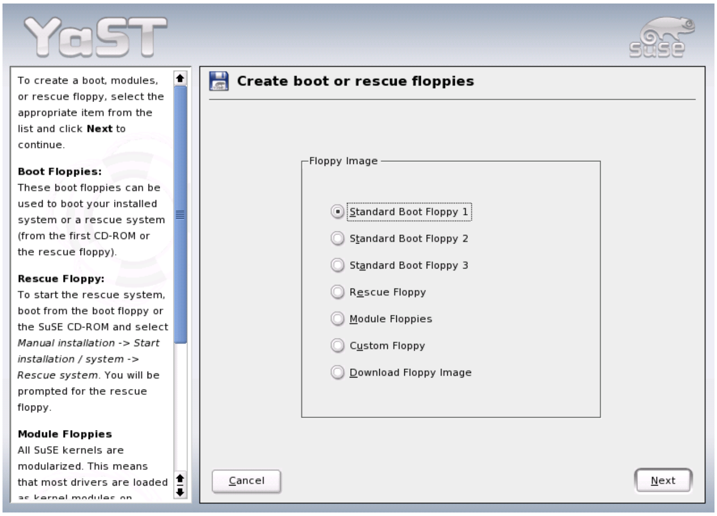
If you press **Select file**, the device will be mounted and

● Local file
○ Network (NFS)
○ Removable device

Figure 2.19: *Start Window of the Restore Module*

2.8.3 Creating a Boot, Rescue, or Module Disk

Use this YaST module to create boot disks, rescue disks, and module disks. These floppy disks are helpful if the boot configuration of your system is damaged. The rescue disk is especially necessary if the file system of the root partition is damaged. In this case, you might also need the module disk with various drivers to be able to access the system (e.g., to access a RAID system).



To create a boot, modules, or rescue floppy, select the appropriate item from the list and click **Next** to continue.

Boot Floppies:
These boot floppies can be used to boot your installed system or a rescue system (from the first CD-ROM or the rescue floppy).

Rescue Floppy:
To start the rescue system, boot from the boot floppy or the SuSE CD-ROM and select *Manual installation -> Start installation / system -> Rescue system*. You will be prompted for the rescue floppy.

Module Floppies
All SuSE kernels are modularized. This means that most drivers are loaded as kernel modules on

Create boot or rescue floppies

Floppy Image

- ● Standard Boot Floppy 1
- ○ Standard Boot Floppy 2
- ○ Standard Boot Floppy 3
- ○ Rescue Floppy
- ○ Module Floppies
- ○ Custom Floppy
- ○ Download Floppy Image

Figure 2.20: Creating a Boot, Rescue, or Module Disk

'Standard Boot Disk' Use this option to create a standard boot disk with which to boot an installed system. This disk is also needed for starting the rescue system.

'Rescue Disk' This disk contains a special environment that allows you to perform maintenance tasks in your installed system, such as checking and repairing the file system and updating the boot loader.

To start the rescue system, boot with the standard boot disk then select 'Manual Installation' → 'Start Installation or System' → 'Rescue System'.

You will then be prompted to insert the rescue disk. If your system was configured to use special drivers (such as RAID or USB), you might need to load the respective modules from a module disk.

'Module Disks' Module disks contain additional system drivers. The standard kernel only supports IDE drives. If the drives in your system are connected to special controllers (such as SCSI), load the needed drivers from a module disk. If you select this option and click 'Next', you will be taken to a dialog for creating various module disks.

The following module disks are available:

USB Modules This floppy disk contains the USB modules you might need if USB drives are connected.

IDE, RAID, and SCSI Modules As the standard kernel only supports normal IDE drives, you will need this module disk if you use special IDE controllers. Furthermore, all RAID and SCSI modules are provided on this disk.

Network Modules If you need access to a network, load the suitable driver module for your network card from this floppy disk.

PCMCIA, CD-ROM (non-ATAPI), FireWire, and File Systems This floppy disk contains all PCMCIA modules used especially for laptop computers. Furthermore, the modules for FireWire and some less common file systems are available here. Older CD-ROM drives that do not comply with the ATAPI standard can also be operated with drivers from this floppy disk.

To load drivers from a module disk to the rescue system, select 'Kernel Modules (hardware drivers)' and the desired module category (SCSI, ethernet, etc.). You are prompted to insert the respective module disk and the contained modules are then listed. Select the desired module. Watch the system messages carefully: 'Loading module <modulename> failed' indicates that the hardware could not be recognized by the module. Some older drivers require specific parameters to be able to address the hardware correctly. In this case, refer to the documentation of your hardware.

'User-Defined Disk' Use this to write any existing floppy disk image from the hard disk to a floppy disk.

'Download Disk Image' With this, enter a URL and authentication data to download a floppy disk image from the Internet.

To create one of these floppy disks, select the corresponding option and click 'Next'. Insert a floppy disk when prompted. If you click 'Next' again, the floppy disk is created.

2.8.4 Boot Loader Configuration

A detailed description of how to configure the boot loader with YaST is available in Section 8.6 on page 218.

2.8.5 LVM

The Logical Volume Manager (LVM) is a tool for custom partitioning of hard disks into logical drives. Information about LVM is available in Section 3.10 on page 131.

2.8.6 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 already existing memory and volume management systems, like DOS, Linux LVM, GPT (GUID Partition Table), S/390, Macintosh, and BSD partitions. More information is provided on `http://evms.sourceforge.net/`.

2.8.7 Partitioning

Although it is possible to modify the partitions in the installed system, this should be handled by experts who know exactly what they are doing, as otherwise the risk of losing data is very high. If you decide to use this tool, refer to the description in Section 1.7.4 on page 17 (the partitioning tool during the installation is the same as in the installed system).

2.8.8 Profile Manager (SCPM)

Note

s390, zseries: Profile Manager

This module is not relevant for SUSE LINUX Enterprise Server on IBM S/390 and zSeries.

Note

The SCPM (System Configuration Profile Management) module offers the possibility of creating, managing, and switching among system configurations. 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, as it enables the use of various hardware components or test configurations.

For more information about SCPM basics and handling, refer to the respective sections in Chapter 16 on page 325.

2.8.9 Runlevel Editor

SUSE LINUX can be operated in several runlevels. By default, the system boots to runlevel 5, which offers multiuser mode, network access, and the graphical user interface (X Window System). The other runlevels offer multiuser mode with network but without X (runlevel 3), multiuser mode without network (runlevel 2), single-user mode (runlevel 1 and S), system halt (runlevel 0), and system reboot (runlevel 6).

The various runlevels are useful if problems are encountered in connection with a particular service (X or network) in a higher runlevel. In this case, the system can be booted to a lower runlevel to repair the service. Many servers operate without a graphical user interface and must be booted in a runlevel without X, such as runlevel 3.

Usually home users only need the standard runlevel (5). However, if the graphical user interface freezes at any time, you can restart the X Window system by switching to a text console with Ctrl-Alt-F1, logging in as root, and switching to runlevel 3 with the command `init 3`. This shuts down your X Window System, leaving you with a text console. To restart the graphical system, enter `init 5`.

In a default installation, runlevel 5 is selected. To start a different runlevel when the system is booted, change the default runlevel here. With 'Runlevel properties', determine which services are started in which runlevel.

Caution

Runlevel Configuration

Incorrect settings for system services and runlevels can render your system useless. To retain the operability of your system, consider the possible consequences before modifying any of these settings.

Caution

More information about runlevels in SUSE LINUX can be found in Chapter 11 on page 259.

2.8.10 Sysconfig Editor

The directory `/etc/sysconfig` contains the files with the most important settings for SUSE LINUX (formerly centrally administered in the file

`/etc/rc.config`). The sysconfig editor displays all settings in a well-arranged form. The values can be modified and saved to the individual configuration files. Generally, manual editing is not necessary, as the files are automatically adapted when a package is installed or a service is configured.

Caution

System Configuration with /etc/sysconfig/

Do not edit the files in `/etc/sysconfig` if you do not know exactly what you are doing, as this could seriously inhibit the operability of your system.

Caution

More information about `/etc/sysconfig/` can be found in Chapter 11 on page 259.

2.8.11 Time Zone Selection

The time zone was already set during the installation, but you can make changes here. Click your country or region in the list and select 'Local time' or 'GMT' (Greenwich Mean Time). 'GMT' is often used in Linux systems. Machines with additional operating systems, such as Microsoft Windows, mostly use local time.

2.8.12 Language Selection

Here, select the language for your Linux system. The language can be changed at any time. The language selected in YaST applies to the entire system, including YaST and the desktop environment KDE.

2.8.13 Keyboard Layout Selection

Note

s390, zseries: Keyboard Layout

Because IBM S/390 and zSeries do not have a locally attached keyboard, this module has no relevance for these architectures.

Note

Note

Configuration of the Keyboard Layout

Only use this module if you work on a system without the X Window System and a graphical user interface. If you use a graphical system (such as KDE), set up the keyboard with the module 'Display and Input Devices'. See Section 2.4.4 on page 70.

Note

The desired keyboard layout usually matches the selected language. Use the test field to see if special characters, such as the pipe symbol `|`, are displayed correctly.

2.9 Miscellaneous

2.9.1 Submitting a Support Request

By purchasing SUSE LINUX, you are entitled to free installation support. For information about the support scope, address, and phone numbers, visit our web site at `www.suse.de/en/`.

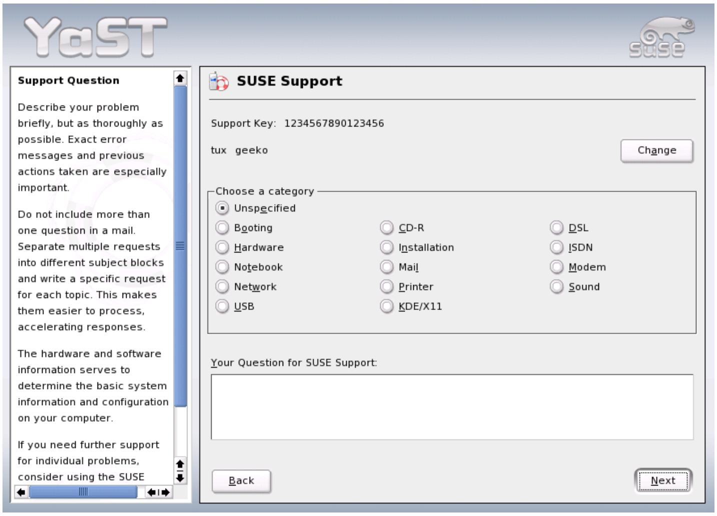
YaST offers the possibility to send a support request directly by e-mail to the SUSE team. Registration is required first. Start by entering the required data — your registration code is located at the back of the CD cover. Regarding your query, select the problem category in the following window and provide a description of the problem (Figure 2.21 on the facing page). Also read the YaST help text, which explains how best to describe the problem so the support team can help you.

2.9.2 Boot Log

The boot log contains the screen messages displayed when the computer is started. It is logged to `/var/log/boot.msg`. Use this YaST module to view the log, for example, to check if all services and functions were started as expected.

2.9.3 System Log

The system log logs the operations of your computer to `/var/log/messsages`. Kernel messages are recorded here, sorted according to date and time.



Describe your problem briefly, but as thoroughly as possible. Exact error messages and previous actions taken are especially important.

Do not include more than one question in a mail. Separate multiple requests into different subject blocks and write a specific request for each topic. This makes them easier to process, accelerating responses.

The hardware and software information serves to determine the basic system information and configuration on your computer.

If you need further support for individual problems, consider using the SUSE

Support Key: 1234567890123456

tux geeko

● Unspecified

○ Booting	○ CD-R	○ DSL
○ Hardware	○ Installation	○ ISDN
○ Notebook	○ Mail	○ Modem
○ Network	○ Printer	○ Sound
○ USB	○ KDE/X11	

Figure 2.21: Submitting a Support Request

2.9.4 Loading a Vendor's Driver CD

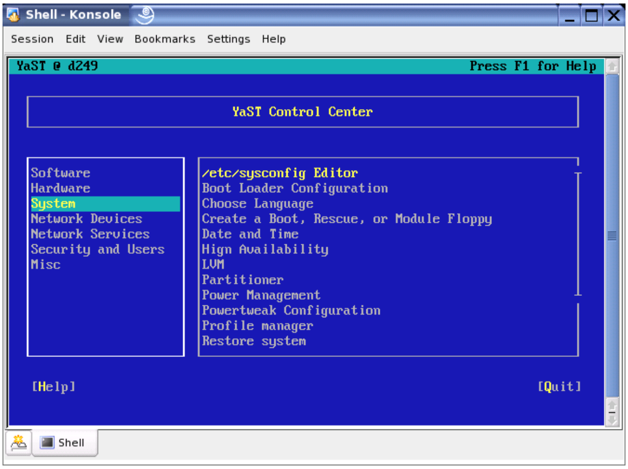
With this module, automatically install device drivers from a Linux driver CD that contains drivers for SUSE LINUX. When installing SUSE LINUX from scratch, use this YaST module to load the required drivers from the vendor CD after the installation.

2.10 YaST in Text Mode (ncurses)

When YaST is started in text mode, the YaST Control Center appears first. See Figure 2.22 on the next page.

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'.

When the YaST Control Center is started, the category 'Software' is selected automatically. Use ↓ and ↑ to change the category. To start a module from



```
YaST @ d249                                                  Press F1 for Help

                              YaST Control Center

 Software               /etc/sysconfig Editor
 Hardware               Boot Loader Configuration
 System                 Choose Language
 Network Devices        Create a Boot, Rescue, or Module Floppy
 Network Services       Date and Time
 Security and Users     Hign Availability
 Misc                   LVM
                        Partitioner
                        Power Management
                        Powertweak Configuration
                        Profile manager
                        Restore system

 [Help]                                                                [Quit]
```

Figure 2.22: Main Window of YaST in Text Mode

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 colorful 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 the 'Exit' button or by selecting 'Exit' in the category overview and pressing Enter.

2.10.1 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 2.10.2 on the facing page 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 (e.g., between the individual modules of a module group in the Control Center). If individual entries within a frame exceed its width, use Shift–→ or Shift–← to scroll horizontally to the right and to the 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. See Figure 2.23 on the next page.

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, as 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.

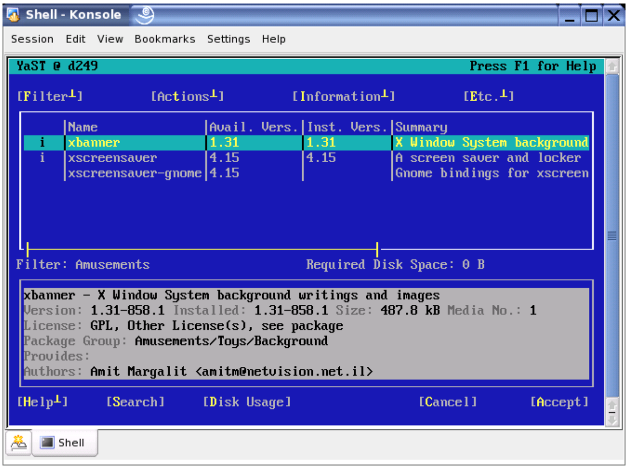
2.10.2 Restriction of Key Combinations

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.

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 F keys should always be fully available on a pure text console.



```
YaST @ d249                                                Press F1 for Help

[Filter┴]         [Actions┴]          [Information┴]         [Etc.┴]

      |Name               |Avail. Vers.|Inst. Vers.|Summary
   i  |xbanner            |1.31        |1.31       |X Window System background
   i  |xscreensaver       |4.15        |4.15       |A screen saver and locker
      |xscreensaver-gnome |4.15        |           |Gnome bindings for xscreen

Filter: Amusements                       Required Disk Space: 0 B

xbanner - X Window System background writings and images
Version: 1.31-858.1 Installed: 1.31-858.1 Size: 487.8 kB Media No.: 1
License: GPL, Other License(s), see package
Package Group: Amusements/Toys/Background
Provides:
Authors: Amit Margalit <amitm@netvision.net.il>

[Help┴]     [Search]     [Disk Usage]               [Cancel]       [Accept]
```

Figure 2.23: *The Software Installation Module*

2.10.3 Starting the Individual Modules

To save time, the individual YaST modules can be started directly. To a module, enter `yast <module_name>`. The network module, for example, is started with `yast lan`. A list of all module names available on your system can be viewed with the command `yast -l` or `yast --list`.

Special Installation Procedures

SUSE LINUX can be installed in a number of ways. The possibilities range from a graphical quick installation to a text-based installation allowing numerous manual adaptions. The following sections cover various installation procedures and the use of diverse installation sources (CD-ROM, NFS). This chapter also features information about resolving problems encountered during the installation, a detailed section about partitioning, and an introduction to SUSE LINUX Enterprise Server on iSCSI.

3.1 linuxrc 106
3.2 Installation with VNC 115
3.3 Text-Based Installation with YaST 116
3.4 Starting SUSE LINUX 118
3.5 Special Installation Procedures 119
3.6 Tips and Tricks 122
3.7 ATAPI CD-ROM Hangs While Reading 126
3.8 Permanent Device File Names for SCSI Devices . . . 127
3.9 Partitioning for Experts 128
3.10 LVM Configuration 131
3.11 Soft RAID 139
3.12 Mass storage via IP networks – iSCSI 142

3.1 linuxrc

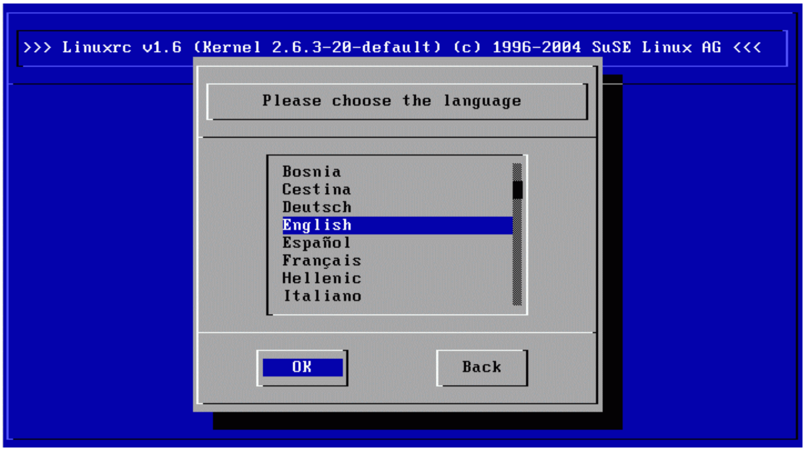
linuxrc is a program that runs in the start-up stage of the kernel prior to the actual boot process. This allows you to boot a small modularized kernel and to load the few drivers that are really needed as modules. linuxrc assists in loading relevant drivers manually. However, the automatic hardware detection performed by YaST is usually quite reliable. The use of linuxrc is not limited to the installation. You can also use it as a boot tool for an installed system and even for an independent RAM disk–based rescue system. Refer to Section 10.4 on page 249 for more details.

The linuxrc program is a tool to define installation settings, and to load hardware drivers (in the form of kernel modules). After doing so, linuxrc hands over control to YaST, which starts the actual installation of system software and applications.

Use ↑ and ↓ to select a menu item, as well as ← and → to select an action, such as 'OK' or 'Cancel'. Perform the selected by pressing Enter.

A more detailed description of linuxrc is available in Section 3.1.

After starting, linuxrc automatically prompts you to select your language and keyboard layout.



```
>>> Linuxrc v1.6 (Kernel 2.6.3-20-default) (c) 1996-2004 SuSE Linux AG <<<
```

Figure 3.1: Selecting the Language

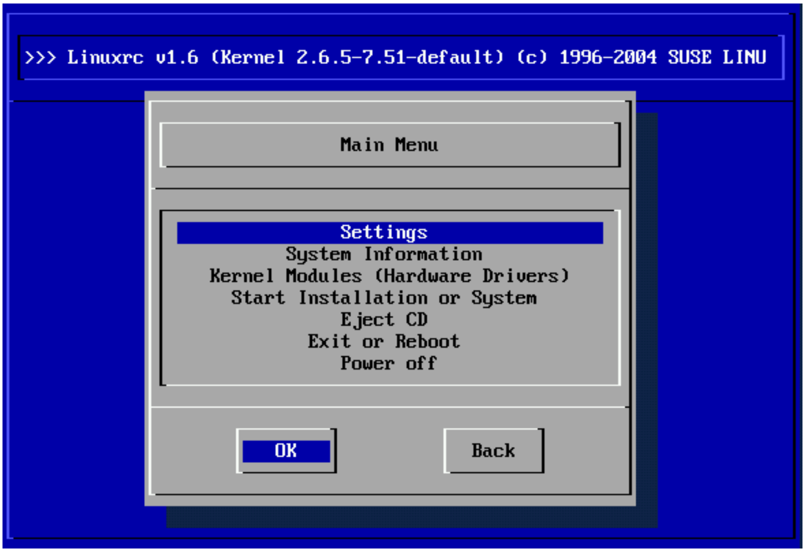
Select your desired installation language (such as 'English') and confirm with Enter. Next, select the layout of your keyboard (for example, 'English

(US)').

3.1.1 Main Menu

After selecting the language and keyboard, continue to the main menu of linuxrc (see Figure 3.2). Normally, linuxrc is used to start Linux, in which case you should select 'Start Installation or System'. You may be able to access this item directly, depending on the hardware and the installation procedure in general.

Refer to Section 3.3 on page 116 for more information.



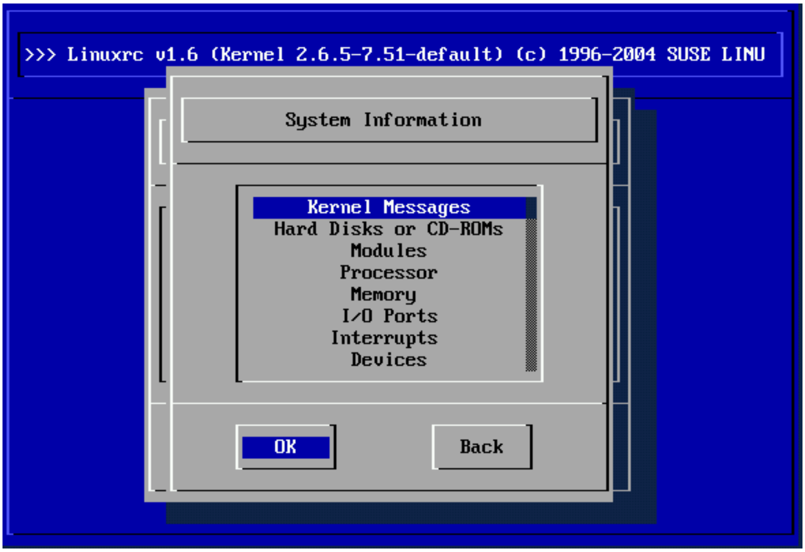
>>> Linuxrc v1.6 (Kernel 2.6.5-7.51-default) (c) 1996-2004 SUSE LINU

Main Menu
Settings
System Information
Kernel Modules (Hardware Drivers)
Start Installation or System
Eject CD
Exit or Reboot
Power off
OK
Back

Figure 3.2: The linuxrc Main Menu

3.1.2 System Information

With the 'System Information' menu, shown in Figure 3.3 on the following page, view kernel messages and other technical details. For example, check the I/O ports used by PCI cards and the memory size as detected by the Linux kernel.


```
>>> Linuxrc v1.6 (Kernel 2.6.5-7.51-default) (c) 1996-2004 SUSE LINU
```

System Information
Kernel Messages
Hard Disks or CD-ROMs
Modules
Processor
Memory
I/O Ports
Interrupts
Devices
OK
Back

Figure 3.3: *System Information*

The next lines show how a hard disk and a CD-ROM connected to an (E)IDE controller announce their start. In this case, you do not need to load additional modules:

```
hda: ST32140A, 2015MB w/128kB Cache, LBA, CHS=1023/64/63
hdb: CD-ROM CDR-S1G, ATAPI CD-ROM drive
Partition check: hda: hda1 hda2 hda3 < hda5 >
```

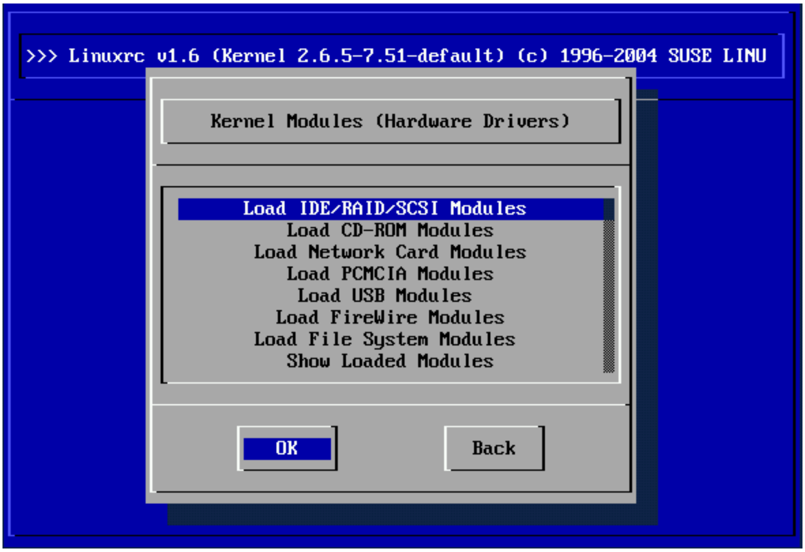
If you have booted a kernel with a SCSI driver already compiled into it, also skip loading a SCSI driver module. When detected, SCSI adapters and connected devices announce themselves like this:

```
scsi: 1 host.
Started kswapd v 1.4.2.2
scsi0: target 0 accepting period 100ns offset 8 10.00MHz FAST SCSI-II
scsi0: setting target 0 to period 100ns offset 8 10.00MHz FAST SCSI-II
  Vendor: QUANTUM   Model: VP32210           Rev: 81H8
  Type:   Direct-Access                      ANSI SCSI revision: 02
Detected scsi disk sda at scsi0, channel 0, id 0, lun 0
scsi0: target 2 accepting period 236ns offset 8 4.23MHz synchronous SCSI
scsi0: setting target 2 to period 248ns offset 8 4.03MHz synchronous SCSI
```

```
  Vendor: TOSHIBA   Model: CD-ROM XM-3401TA  Rev: 0283
  Type:   CD-ROM                             ANSI SCSI revision: 02
scsi: detected 1 SCSI disk total.
SCSI device sda:hdwr sector=512 bytes.Sectors=4308352 [2103 MB] [2.1 GB]
Partition check:
 sda: sda1 sda2 sda3 sda4 < sda5 sda6 sda7 sda8 >
```

3.1.3 Loading Modules

Select the modules (drivers) needed. linuxrc offers the available drivers in a list. The name of the respective module is displayed to the left and a brief description of the hardware supported by the driver is displayed to the right. For some components, linuxrc offers several drivers or newer alpha versions of them.



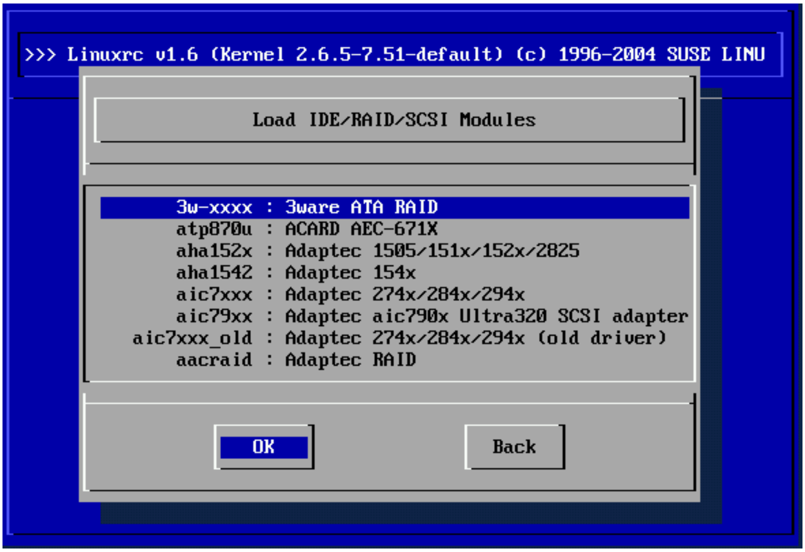
```
>>> Linuxrc v1.6 (Kernel 2.6.5-7.51-default) (c) 1996-2004 SUSE LINU
```

Kernel Modules (Hardware Drivers)
Load IDE/RAID/SCSI Modules
Load CD-ROM Modules
Load Network Card Modules
Load PCMCIA Modules
Load USB Modules
Load FireWire Modules
Load File System Modules
Show Loaded Modules
OK
Back

Figure 3.4: Loading Modules

3.1.4 Entering Parameters

Locate a suitable driver for your hardware and press Enter. This opens a dialog in which to enter additional parameters for the module. Separate multiple parameters for one module with spaces.



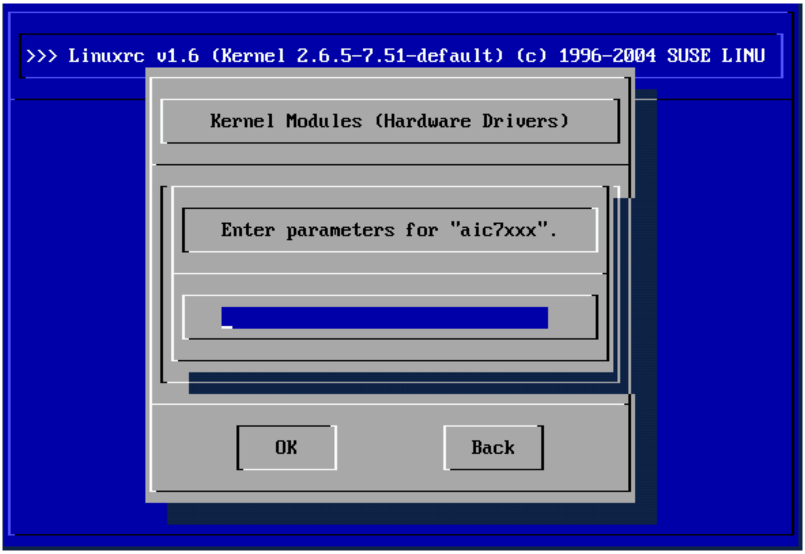
```
>>> Linuxrc v1.6 (Kernel 2.6.5-7.51-default) (c) 1996-2004 SUSE LINU

                  Load IDE/RAID/SCSI Modules

          3w-xxxx : 3ware ATA RAID
          atp870u : ACARD AEC-671X
          aha152x : Adaptec 1505/151x/152x/2825
          aha1542 : Adaptec 154x
          aic7xxx : Adaptec 274x/284x/294x
          aic79xx : Adaptec aic790x Ultra320 SCSI adapter
      aic7xxx_old : Adaptec 274x/284x/294x (old driver)
         aacraid : Adaptec RAID
```

Figure 3.5: *Selecting SCSI Drivers*

In many cases, it is not necessary to specify the hardware in detail, as most drivers find their components automatically. Only network cards and older CD-ROM drives with proprietary controller cards may require parameters. If unsure, try pressing Enter.

For some modules, the detection and initialization of the hardware can take some time. Switch to virtual console 4 (Alt-F4) to watch the kernel messages while loading. SCSI drivers especially take some time, as they wait until all attached devices respond.

If the module is loaded successfully, linuxrc displays the kernel messages, allowing you to verify that everything worked smoothly. In the event of a problem, the messages may indicate the reason.



```
>>> Linuxrc v1.6 (Kernel 2.6.5-7.51-default) (c) 1996-2004 SUSE LINU
```

Kernel Modules (Hardware Drivers)

Enter parameters for "aic7xxx".

Figure 3.6: Entering Parameters for a Module

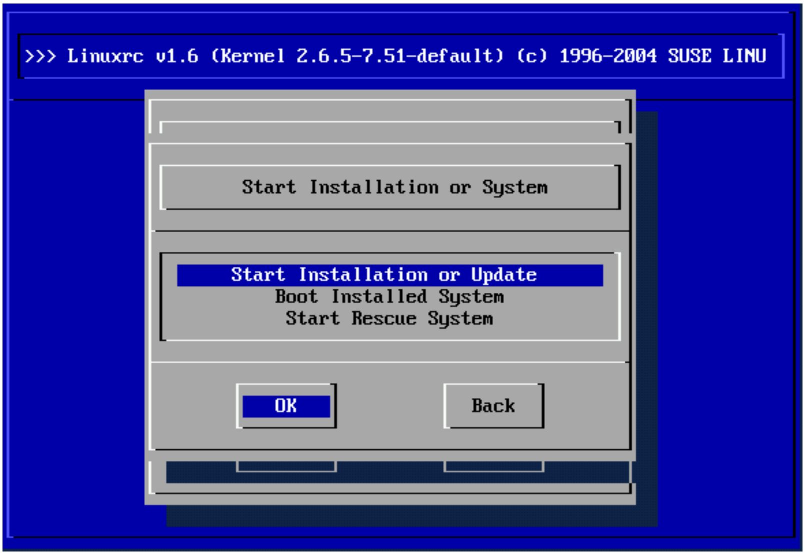
Note

If it turns out that no driver is included for your installation device (proprietary or parallel port CD-ROM drive, network card, PCMCIA) among the standard modules, you may be able to use one of the drivers of an extra module disk (to learn how to make such a floppy, refer to Section 3.6 on page 122). To do so, scroll down to the end of the menu then select 'Other Modules'. `linuxrc` then prompts you to insert the corresponding disk.

Note

3.1.5 Start Installation or System

After setting up hardware support via modules, proceed to 'Start Installation or System'. From this menu, a number of procedures can be started: 'Start Installation or Update', 'Boot Installed System' (the root partition must be known), 'Start Rescue System' (see Section 10.4 on page 249), and 'Eject CD'.



```
>>> Linuxrc v1.6 (Kernel 2.6.5-7.51-default) (c) 1996-2004 SUSE LINU
```

Start Installation or System
Start Installation or Update
Boot Installed System
Start Rescue System
OK
Back

Figure 3.7: The linuxrc Installation Menu

'Start LiveEval CD' is only available if you booted a *LiveEval CD*. Download ISO images from the FTP server (`live-eval-<VERSION>`) at `ftp://ftp.suse.com/pub/suse/i386/`

To begin the installation, select 'Start Installation or Update' from the menu and press Enter. You are then prompted to select the installation source. In most cases, you can leave this at the preselected 'CD-ROM' item.

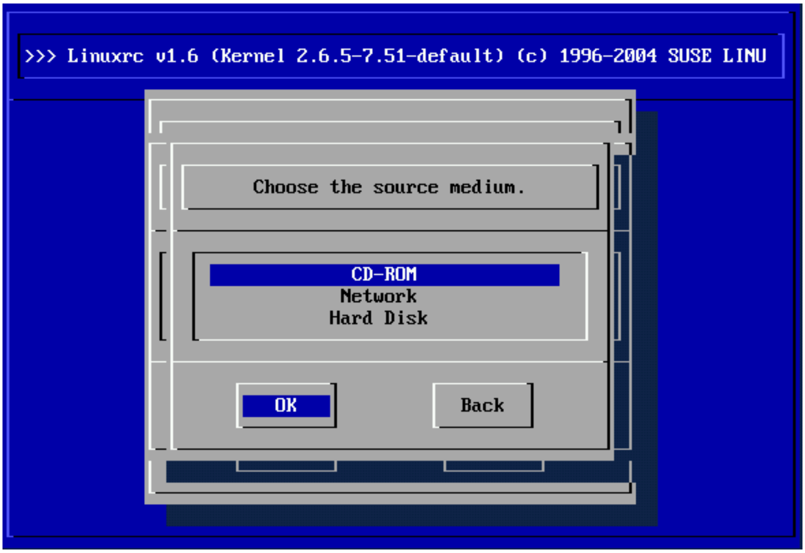
You are now ready to go. After pressing Enter, the installation environment loads from CD 1 or from the DVD. As soon as this process is completed, YaST starts with its text (ncurses-based) interface and the installation begins.

Note

'Start LiveEval CD' is very useful for testing the compatibility of a computer or laptop without installing the system on the hard disk.

Note

It is possible to select among different source media for the installation (Figure 3.8 on the next page), and in a similar way for the rescue system (Figure 10.1 on page 251).



```
>>> Linuxrc v1.6 (Kernel 2.6.5-7.51-default) (c) 1996-2004 SUSE LINU
```

Choose the source medium.

Figure 3.8: Selecting the Source Medium in linuxrc

3.1.6 Potential Problems

- The desired keyboard layout is not offered by linuxrc. To solve this, select an alternative, such as 'English (US)'. After the installation is completed, adjust this setting with YaST.

- The SCSI adapter of your machine is not recognized.

  Try loading the module of a compatible adapter. Also check whether a disk with a driver update for your adapter has been made available.

- Your ATAPI CD-ROM drive hangs when the system tries to read from it. See Section 3.7 on page 126.

- In some cases, there may be a problem loading the data into the RAM disk, making it impossible for YaST to start. If this happens, try the following steps, which should fix the error.

  From the linuxrc main menu, select 'Settings' → 'Debug (Expert)'. In the dialog that opens, set 'Force Root Image' to 'no'. Then return to the main menu and try starting the installation again.

3.1.7 Passing Parameters to linuxrc

If linuxrc does not run in manual mode, it looks for an info file on a floppy disk or in the `initrd` in `/info`. Subsequently, linuxrc loads the parameters at the kernel prompt. You can edit the default values in the file `/linuxrc.config`. However, the recommended method is to implement changes in the info file.

An info file consists of keywords and values in the format `key: value`. These pairs of keys and values can also be entered at the boot prompt provided by the installation medium using the format `key=value`. A list of all keys is available in the file `/usr/share/doc/packages/linuxrc/linuxrc.html`. The following list shows some of the most important keys with example values:

Install: URL (nfs, ftp, hd, etc.) Lets you specify the installation source as a URL. Possible protocols include `cd`, `hd`, `nfs`, `smb`, `ftp`, `http`, and `tftp`. The URL syntax corresponds to the common form as used in web browsers, for example:

- `nfs://<server>/<directory>`
- `ftp://[user[:password]@]<server>/<directory>`

Netdevice: eth0 The `Netdevice:` keyword specifies the interface linuxrc should use, if there are several ethernet interfaces available on the installation host.

HostIP: 10.10.0.2 Specifies the IP address of the host.

Gateway: 10.10.0.128 This specifies the gateway through which the installation server can be reached, if it is not located in the subnetwork of the host.

Proxy: 10.10.0.1 The `Proxy:` keyword defines a proxy for the FTP and HTTP protocols.

ProxyPort: 3128 This specifies the port used by the proxy, if it does not use the default port.

Textmode: 0|1 This keyword enables starting YaST in text mode.

AutoYast: ftp://autoyastfile The `AutoYast` keyword can be used to initiate an automatic installation. The value must be a URL pointing to an AutoYaST installation file.

VNC: 0|1 The `VNC` parameter controls the installation process via VNC, which makes the installation more convenient for hosts that do not have a graphical console. If enabled, the corresponding service is activated on the installation host. Also see the `VNCPassword` keyword.

VNCPassword: password This sets a password for a VNC installation to control access to the session.

UseSSH: 0|1 This keyword enables access to linuxrc via SSH when performing the installation with YaST in text mode.

SSHPassword: password This sets the password for the `root` user launching linuxrc.

Insmod: module parameters This specifies a module the kernel should load, together with any parameters needed for it. Module parameters must be separated by blank spaces.

AddSwap: 0|3|/dev/hda5 If set to 0, the system does not try to activate a swap partition. If set to a positive number, the partition corresponding to the number is activated as a swap partition. Alternatively, specify the full device name of a partition.

3.2 Installation with VNC

VNC (*virtual network computing*) is a client-server solution that allows a remote X server to be managed

via a slim and easy-to-use client. This client is available for a variety of operating systems, including Microsoft Windows, Apple's MacOS, and Linux.

The VNC client, vncviewer, is used to ensure the graphical display and handling of YaST during the installation process. Before booting (IPL) the system to install, prepare a remote computer so it can access the system to install over the network.

3.2.1 Preparing for the VNC Installation

► **s390, zseries**
As described in the Architecture-specific Information manual, it is only necessary to choose the `VNC` connection option in the installation process for S/390 and zSeries. This option allows any VNC client to be connected

to the installation system and ensures that the installation process can be carried out with the graphical YaST. ◄

To perform a VNC installation, pass certain parameters to the kernel. This must be done before the kernel is launched. To do this, enter the following command at the boot prompt:

```
vnc=1 vncpassword=<xyz> install=<Quelle>
```

`vnc=1` signals that the VNC server should be launched on the installation system. `vncpassword` is the password to use later. The installation source (`install`) can either be specified manually (enter the protocol and URL for the directory concerned) or it can contain the instruction `slp:/`. In the latter case, the installation source is automatically determined by SLP query. Information about SLP is contained in Section 21.6 on page 452.

3.2.2 Clients for the VNC Installation

The connection to the installation computer and the VNC server running on it is established via a VNC client. Under SUSE LINUX, use vncviewer. This is part of the `XFree86-Xvnc` package. To establish a connection to the installation system from a Windows client, install the tightvnc program on the Windows system. This program is on the first SUSE LINUX CD, in the `/dosutils/tightvnc/` directory.

Launch the VNC client of your choice. Then, when prompted, enter the IP address of the installation system along with the VNC password.

Alternatively, establish VNC connections using a Java-capable browser. To do this, enter the following into the address field of the browser:

```
http://<IP-Adresse des Installationssystems>:5801/
```

Once the connection has been established, YaST launches and the installation can start.

3.3 Text-Based Installation with YaST

In addition to installing with the assistance of a graphical interface, SUSE LINUX can also be installed with the help of the text version of YaST (console mode). All YaST modules are also available in this text mode. The text mode is especially useful if you do not need a graphical interface (e.g., for server systems) or if the graphics card is not supported by the X Window System. The visually impaired can also benefit from this text mode.

First, set the boot sequence in the BIOS to enable booting from the CD-ROM drive. Insert the DVD or CD 1 in the drive and reboot the machine. The start screen is displayed after a few seconds.

Use ↑ and ↓ to select 'Manual Installation' within ten seconds to prevent YaST from starting automatically. If your hardware requires special parameters, which is not usually the case, enter these in `Boot Options`. The parameter `textmode=1` can be used to force YaST to run in text mode.

Use F2 ('Video Mode') to set the screen resolution for the installation. If you expect your graphics card to cause problems during the installation, select 'Text Mode'. Then press Enter. A box appears with the progress display `Loading Linux kernel`. The kernel boots and linuxrc starts. Proceed with the installation using the menus of linuxrc.

Other boot problems can usually be circumvented with kernel parameters. If DMA causes difficulties, use the start option 'Installation — Safe Settings'. If your CD-ROM drive (ATAPI) crashes when booting the system, refer to Section 3.7 on page 126.

The following kernel parameters may be used if you experience problems with ACPI (advanced configuration and power interface).

`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=oldboot` Switch off ACPI for everything but those parts that are necessary to boot.

`acpi=force` Always enables 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`.

`pci=noacpi` Prevents ACPI from doing the PCI IRQ routing.

Also refer to the SDB article `http://portal.suse.com/sdb/en/2002/10/81_acpi.html`.

If unexplainable errors occur when the kernel is loaded or during the installation, select 'Memory Test' in the boot menu to check the memory. Linux requires the hardware to meet high standards, which means the memory and its timing must be set correctly. More information is available at `http://portal.suse.com/sdb/en/2001/05/thallma_memtest86.html`. If possible, run the memory test overnight.

3.4 Starting SUSE LINUX

Following the installation, decide how to boot Linux for daily operations. The following overview introduces various alternatives for booting Linux. The most suitable method depends on the intended purpose.

Boot Disk You can boot Linux from a *boot disk*. This approach will always work and is easy. The boot disk can be created with YaST. See the *User Guide* manual, 'YaST — Configuration': 'Creating a Boot, Rescue, or Module Disk'.

The boot disk is a useful interim solution if you have difficulties configuring the other possibilities or if you want to postpone the decision regarding the final boot mechanism. A boot disk may also be a suitable solution in connection with OS/2 or Windows NT.

Linux Boot Loader The most versatile and technically elegant solution for booting your system is the use of a Linux boot manager like GRUB (GRand Unified Bootloader) or LILO (LInux LOader), which both allow selection from different operating systems prior to booting. The boot loader can either be configured during installation or later with the help of YaST.

Caution

There are BIOS variants that check the structure of the boot sector (MBR) and erroneously display a virus warning after the installation of GRUB or LILO. This problem can be easily solved by entering the BIOS and looking for corresponding adjustable settings. For example, you should switch off 'virus protection'. You can switch this option back on again later. It is unnecessary, however, if Linux is the only operating system you use.

Caution

A detailed discussion of various boot methods, especially of GRUB and LILO, can be found in Section 8 on page 199.

3.4.1 The Graphical SUSE Screen

Starting with SUSE LINUX 7.2, the graphical SUSE screen is displayed on the first console if the option "vga=<value>" is active as kernel parameter. If you install using YaST, this option is automatically activated in accordance with the selected resolution and the graphics card.

Disabling the SUSE Screen

Basically there are three ways to achieve this:

- Disabling the SUSE screen whenever necessary. Enter the command `echo 0 >/proc/splash` on the command line to disable the graphical screen. To activate it again, enter `echo 0x0f01 >/proc/splash`.
- Disabling the SUSE screen by default. Add the kernel parameter `splash=0` to your boot loader configuration. 8 on page 199 provides more information about this. However, if you prefer the text mode which was the default in previous 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 the menu 'framebuffer support'.

> **Note**
>
> Disabling framebuffer support in the kernel will automatically disable the splash screen as well. SUSE cannot provide any support for your system if you run it with a custom kernel.
>
> **Note**

3.5 Special Installation Procedures

3.5.1 Automatic Installation with AutoYaST

If the installation needs to be performed on many similar machines, it makes sense to use Autoyast for the task.

AutoYaST relies on the hardware detection mechanism of YaST and normally uses default settings, but it can also be configured to suit your needs. Therefore, installation hosts need not be strictly identical. It is sufficient for them to have a similar hardware setup. You still need to take into account the limitations of the hardware itself, which cannot be circumvented by AutoYaST.

YaST includes an AutoYaST module, which can be used to create the necessary configuration. This configuration is written to an XML file, so can also be edited or even created manually.

Further information and extensive documentation for AutoYaST is included in the `autoyast2` package. When installed, the documentation is at `/usr/share/doc/packages/autoyast2/html/index.html`.

3.5.2 Installation from a Network Source

No installation support is available for this approach. Therefore, the following procedure should only be attempted by experienced computer users.

To install SUSE LINUX from a network source, two steps are necessary:

1. The data required for the installation (CDs, DVD) must be made available on a machine that will serve as the installation source.
2. The system to install must be booted from floppy disk or CD and the network must be configured.

Configuring a Network Installation Source

Prepare the network share by copying the installation CDs to individual directories and make these available on a system with an NFS server. For example, on an existing SUSE LINUX machine, copy the individual CDs with `cp -a /mnt/cdrom /suse-share/`.

Then rename the directory, for example, to *CD1*, with a command like `mv /suse-share/cdrom /suse-share/CD1`. Repeat this procedure for the other CDs. Then export the `/suse-share` directory via NFS. For information, see Section 21.10 on page 518.

Booting for the Network Installation

Insert the boot medium in the drive. The creation of the boot disk is described in Section 3.6.1 on the next page and Section 3.6.2 on page 124. The boot menu appears after a short time. Select 'Manual Installation'. You can also specify additional kernel parameters. Confirm the selection with Enter. The kernel loads and you are prompted to insert the first module disk.

After a short time, linuxrc appears and asks you to enter a number of parameters:

1. Select the language and the keyboard layout in linuxrc.
2. Select 'Kernel modules (hardware drivers)'.
3. Load the IDE, RAID, or SCSI drivers required for your system.
4. Select 'Load network card modules' and load the needed network card driver (e.g., `eepro100`).
5. Select 'Load file system driver' and load the needed drivers (e.g., `reiserfs`).
6. Select 'Back' then 'Start installation / system'.

7. Select 'Start installation/update'.
8. Select 'Network' then the network protocol (e.g., `NFS`).
9. Select the network card to use.
10. Specify the IP addresses and other network information.
11. Specify the IP address of the NFS server providing the installation data.
12. Enter the path of the NFS share (e.g., `/suse-share/CD1`).

linuxrc then loads the installation environment from the network source and starts YaST. Proceed with the installation as described in the *User Guide* in the chapter *Installation*.

Troubleshooting

- The installation is terminated before it actually starts. The installation directory of the *other* machine

  was not exported with `exec` permissions. Do this now.

- The server does not recognize the host on which to install SUSE LINUX. Enter the name and IP address of the host to install in the `/etc/hosts` file of the server.

3.6 Tips and Tricks

3.6.1 Creating a Boot Disk in DOS

You need formatted 3.5″ HD floppy disks and a bootable 3.5″ floppy disk drive. The `boot` directory on CD 1 contains a number of disk images. With a suitable utility, these images can be copied to floppy disks. A floppy disk prepared in this way is referred to as a boot disk.

The disk images also 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.

Creating a Boot Disk with rawwritewin

In Windows, boot disks can be created with the graphical utility rawwritewin. In Windows, find this utility in the directory `dosutils/rawwritewin/` on CD 1.

On start-up, specify the image file. The image files are located in the `boot` directory on CD 1. At the least, you will need the images "bootdisk" and "modules1". To list these images in the file browser, set the file type to "all files". Then insert a floppy disk in your floppy disk drive and click "write". To create several floppy disks, repeat the same procedure.

Creating a Boot Disk with rawrite

The DOS utility `rawrite.exe` (CD 1, directory `dosutils/rawrite` can be used for creating SUSE boot and module disks. To use this utility, you need a computer with DOS (such as FreeDOS) or Windows.

In Windows XP, proceed as follows:

1. Insert SUSE LINUX CD 1.
2. Open a DOS window (in the start menu, select 'Accessories' → 'Command Prompt').
3. Run rawrite.exe with the correct path specification for the CD drive. The example assumes that you are in the directory `Windows` on the hard disk `C:` and your CD drive is `D:`.

   ```
   d:\dosutils\rawrite\rawrite
   ```

4. On start-up, the utility asks for the source and destination of the file to copy. The image of the boot disk is located in the directory `boot/` on CD 1. The file name is `bootdisk`. Remember to specify the path for your CD drive.

   ```
   d:\dosutils\rawrite\rawrite
   RaWrite 1.2 - Write disk file to raw floppy diskette

   Enter source file name: d:\boot\bootdisk
   Enter destination drive: a:
   ```

After you enter the destination drive `a:`, rawrite prompts you to insert a formatted floppy disk and press Enter. Subsequently, the progress of the copy action is displayed. The process can be terminated with Ctrl + C.

The other disk images (`modules1`, `modules2`, `modules3`, and `modules4`) can be created in the same way. These floppy disks are required if you have USB or SCSI devices or a network or PCMCIA card that you want to address during the installation. A module disk may also be needed if using a special file system during the installation.

3.6.2 Creating a Boot Disk in a UNIX-Type System

On a UNIX or Linux system, you need a CD-ROM drive and a formatted floppy disk. Proceed as follows to create boot disks:

1. If you need to format the disks first, use:

   ```
   fdformat /dev/fd0u1440
   ```

2. Mount CD 1 (for example, to `/media/cdrom`):

   ```
   mount -tiso9660 /dev/cdrom /media/cdrom
   ```

3. Change to the `boot` directory on the CD:

   ```
   cd /media/cdrom/boot
   ```

4. Create the boot disk with the following command:

   ```
   dd if=/media/cdrom/boot/bootdisk of=/dev/fd0 bs=8k
   ```

The `README` file in the `boot` directory provides details about the floppy disk images. You can read these files with `more` or `less`.

The other disk images, `modules1`, `modules2`, `modules3`, and `modules4`, can be created in the same way. These floppy disks are required if you have USB or SCSI devices or a network or PCMCIA card that you want to address during the installation. A module disk may also be needed to use a special file system during the installation.

To use a custom kernel during the installation, the procedure is a bit more complex. In this case, write the default image `bootdisk` to the floppy disk then overwrite the kernel `linux` with your own kernel (see Section 9.6 on page 233):

```
dd if=/media/cdrom/boot/bootdisk of=/dev/fd0 bs=8k
mount -t msdos /dev/fd0 /mnt
cp /usr/src/linux/arch/i386/boot/vmlinuz /mnt/linux
umount /mnt
```

3.6.3 Booting from a Floppy Disk (SYSLINUX)

The boot disk can be used for handling special installation requirements (for example, if the CD-ROM drive is not available). See Section 3.6.1 on page 122 or Section 3.6.2 on the preceding page for more information about creating boot disks.

The boot procedure is initiated by the boot loader SYSLINUX (`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 supports 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.

Note

If you keep Shift pressed when SYSLINUX is started, all these steps will be skipped. For troubleshooting purposes: if you insert the line

```
verbose 1
```

in `syslinux.cfg`, the boot loader tells you which action is currently being performed.

Note

If the machine does not boot from the floppy disk, you may have to change the boot sequence in the BIOS to `A,C,CDROM`.

3.6.4 Using CD 2 for Booting

CD 2 is also bootable. In contrast to CD 1, which uses a bootable ISO image, CD 2 is booted by means of 2.88 MB disk image. Use CD 2 if you are sure you can boot from CD, but it does not work with CD 1 (fallback solution).

3.6.5 Supported CD-ROM Drives

Most CD-ROM drives are supported.

- ATAPI drives should work smoothly.
- The support of SCSI CD-ROM drives depends on whether the SCSI controller to which the CD-ROM drive is connected is supported. Supported SCSI controllers are listed in the Hardware Database at `http://cdb.suse.de`. If your SCSI controller is not supported and your hard disk is also connected to the controller, you have a problem.
- Many vendor-specific CD-ROM drives are supported in Linux. Nevertheless, problems may be encountered with this kind of drives. If your drive is not explicitly listed, try using a similar type from the same vendor.
- USB CD-ROM drives are also supported. If the BIOS of your machine does not support booting from USB devices, start the installation by means of the boot disks. For details, refer to Section 3.6.3 on the preceding page. Before you boot from the floppy disk, make sure all needed USB devices are connected and powered on.

3.7 ATAPI CD-ROM Hangs While Reading

If your ATAPI CD-ROM is not recognized or it hangs while reading, this is most frequently due to incorrectly installed hardware. All devices must be connected to the EIDE controller in the correct order. The first device is master on the first controller. The second device is slave on the first controller. The third device should be master on the second controller, and so forth.

It often occurs that there is only a CD-ROM besides the first device. The CD-ROM drive is sometimes connected as master to the second controller (secondary IDE controller). This is wrong and can cause Linux not to know what to do with this *gap*. Try to fix this by passing the appropriate parameter to the kernel (`hdc=cdrom`).

Sometimes one of the devices is just *misjumpered*. This means it is jumpered as slave, but is connected as master, or vice versa. When in doubt, check your hardware settings and correct them where necessary.

In addition, there is a series of faulty EIDE chipsets, most of which have now been identified. There is a special kernel to handle such cases. See the `README` in `/boot` of the installation CD-ROM.

If booting does not work immediately, try using the following kernel parameters:

`hdx=cdrom` `x` stands for `a`, `b`, `c`, `d`, etc., and is interpreted as follows:

- `a` — Master on the first IDE controller
- `b` — Slave on the first IDE controller
- `c` — Master on the second IDE controller

An example of `parameter to enter` is `hdb=cdrom`. With this parameter, specify the CD-ROM drive to the kernel, if it cannot find it itself and you have an ATAPI CD-ROM drive.

`idex=noautotune` `x` stands for `0`, `1`, `2`, `3`, etc., and is interpreted as follows:

- `0` — First IDE controller
- `1` — Second IDE controller

An example of `parameter to enter` is `ide0=noautotune`. This parameter is often useful for (E)IDE hard disks.

3.8 Assigning Permanent Device File Names to SCSI Devices

When the system is booted, SCSI devices are assigned device file names in a more or less dynamic way. This is no problem as long as the number or configuration of the devices does not change. However, if a new SCSI hard disk is added and the new hard disk is detected by the kernel before the old hard disk, the old disk will be assigned new names and the entries in the mount table `/etc/fstab` will no longer match.

To avoid this problem, use `boot.scsidev`. `boot.scsidev` handles the setup of the SCSI devices during the boot procedure and enters permanent device names under `/dev/scsi/`. These names can be used in `/etc/fstab`.

In the expert mode of the runlevel editor, activate `boot.scsidev` for level `B`. The links needed for generating the names during the boot procedure are then created in `/etc/init.d/boot.d`.

3.9 Partitioning for Experts

This section provides detailed information for tailoring system partitioning to your needs. This information is mainly of interest for those who want to optimize a system for security and speed and who are prepared to reinstall the entire existing system if necessary.

The procedures described here require a basic understanding of the functions of a UNIX file system. You should be familiar with mount points and physical, extended, and logical partitions.

First, consider the following questions:

- How will the machine be used (file server, application server, compute server, stand-alone machine)?
- How many people will work with this machine (concurrent logins)?
- How many hard disks are installed? What is their size and type (EIDE, SCSI, or RAID controllers)?

3.9.1 Size of the Swap Partition

Many sources state the rule that the swap size should be at least twice the size of the main memory. This is a relic of times when 8 MB RAM was considered a lot. In the past, the aim was to equip the machine with about 30 to 40 MB of virtual memory (RAM plus swap). Modern applications require even more memory. For normal users, 512 MB of virtual memory is currently a reasonable value. Never configure your system without any swap memory.

3.9.2 Partitioning Proposals for Special Purposes

File Server

Here, hard disk performance is crucial. Use SCSI devices if possible. Keep in mind the performance of the disk and the controller. A file server is used to save data, such as user directories, a database, or other archives, centrally. This approach greatly simplifies the data administration.

Optimizing the hard disk access is vital for file servers in networks of more than twenty users. Suppose you want to set up a Linux file server for the home directories of 25 users. If the average user requires 100–150 MB for personal data, a 4 GB partition mounted under `/home` is probably sufficient. For fifty users, you would need 8 GB. If possible, split `/home/` to two 4 GB hard disks that share the load (and access time).

Note

Web browser caches should be stored on local hard disks.

Note

Compute Server

A compute server is generally a powerful machine that carries out extensive calculations in the network. Normally, such a machine is equipped with a large main memory (more than 512 RAM). Fast disk throughput is only needed for the swap partitions. If possible, distribute swap partitions to multiple hard disks.

3.9.3 Optimization

The hard disks are normally the limiting factor. To avoid this bottleneck, combine the following three possibilities:

- Distribute the load evenly to multiple disks.
- Use an optimized file system, such as `reiserfs`).
- Equip your file server with a sufficient amount of memory (at least 256 MB).

Parallel Use of Multiple Disks

The total amount of time needed for providing requested data consists of the following elements:

1. Time elapsed until the request reaches the disk controller.
2. Time elapsed until this request is send to the hard disk.
3. Time elapsed until the hard disk positions its head.
4. Time elapsed until the media turns to the respective sector.
5. Time elapsed for the transmission.

The first item depends on the network connection and must be regulated there. Item two is a relatively insignificant period that depends on the hard disk controller itself. Items three and four are the main parts. The positioning time is measured in ms. Compared to the access times of the main memory, which are measured in ns, this represents a factor of one million. Item four depends on the disk rotation speed, which is usually several ms. Item five depends on the rotation speed, the number of heads, and the current position of the head (inside or outside).

To optimize the performance, the third item should be improved. For SCSI devices, the *disconnect* feature comes into play. When this feature is used, the controller sends the command *Go to track x, sector y* to the connected device (in this case, the hard disk). Now the inactive disk mechanism starts moving. If the disk is smart (if it supports disconnect) and the controller driver also supports this feature, the controller immediately sends the hard disk a disconnect command and the disk is disconnected from the SCSI bus. Now, other SCSI devices can proceed with their transfers. After some time (depending on the strategy or load on the SCSI bus) the connection to the disk is reactivated. In the ideal case, the device will have reached the requested track.

On a multitasking, multiuser system like Linux, these parameters can be optimized effectively. For example, examine the following excerpt of the output of the command `df` (see Example 3.1).

Example 3.1: *Example df Output*

```
Filesystem Size Used Avail Use% Mounted on
/dev/sda5 1.8G 1.6G 201M 89% /
/dev/sda1 23M 3.9M 17M 18% /boot
/dev/sdb1 2.9G 2.1G 677M 76% /usr
/dev/sdc1 1.9G 958M 941M 51% /usr/lib
shmfs 185M 0 184M 0% /dev/shm
```

To demonstrate the advantages, consider what happens if `root` enters the following in `/usr/src`:

```
tar xzf package.tgz -C /usr/lib
```

This command extracts `package.tgz` to `/usr/lib/package`. To do this, the shell runs tar and gzip (both located in `/bin` on `/dev/sda`) then `package.tgz` is read by `/usr/src` (on `/dev/sdb`). Finally, the extracted data is written to `/usr/lib` (on `/dev/sdc`). Thus, the positioning as well as the reading and writing of the disks' internal buffers can be performed almost concurrently.

This is only one of many examples. As a general rule, if you have several hard disks (with the same speed), `/usr` and `/usr/lib` should be placed on separate disks. `/usr/lib` should have about seventy percent of the capacity of `/usr`. Due to the frequency of access, `/` should be placed on the disk containing `/usr/lib`.

Speed and Main Memory

In Linux, the size of main memory is often more important than the processor speed. One reason, if not the main reason, for this is the ability of Linux to create dynamic buffers containing hard disk data. For this purpose, Linux uses various tricks, such as *read ahead* (reading of sectors in advance) and *delayed write* (postponement and bundling of write access). The latter is the reason why you should not simply switch off your Linux machine. Both factors contribute to the fact that the main memory seems to fill up over time and that Linux is so fast. See Section 10.2.6 on page 243.

3.10 LVM Configuration

This professional partitioning tool enables you to edit and delete existing partitions and create new ones. Access the Soft RAID and LVM configuration from here.

Note

Background information and partitioning tips can be found in Section 3.9 on page 128.

Note

In normal circumstances, partitions are set up during installation. However, it is possible to integrate a second hard disk in an existing Linux system. First, the new hard disk must be partitioned. Then it must be mounted and entered into the `/etc/fstab` file. It may be necessary to copy some of the data to move an `/opt` partition from the old hard disk to the new one.

Use caution repartitioning the hard disk in use — this is essentially possible, but you will have to reboot the system right afterwards. It is a bit safer to boot from CD then repartition it.

'Experts. . . ' opens a pop-up menu containing the following commands:

Reread Partition Table Rereads the partitioning from disk. For example, you need this for manual partitioning in the text console.

Adopt Mount Points from Existing /etc/fstab
This is only relevant during installation. Reading the old `fstab` is useful for completely reinstalling your system rather than just updating it. In this case, it is not necessary to enter the mount points by hand.

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.

3.10.1 Logical Volume Manager (LVM)

Starting from kernel version 2.6, you can use LVM version 2, 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. LVM2 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 chapter always refer to LVM version 2.

Instead of LVM2, you can also use EVMS (Enterprise Volume Management System), which offers a uniform interface for logical volumes as well as RAID volumes. Like LVM2, EVMS makes use of the device mapper in kernel 2.6.

The Logical Volume Manager (LVM) enables flexible distribution of hard disk space over several file systems. As it is difficult to modify partitions on a running system, LVM was developed. It provides a virtual pool (Volume Group — VG for short) of memory space from which logical volumes (LV) can be generated if needed. The operating system accesses these instead of the physical partitions.

Features:

- Several hard disks or partitions can be combined to a large logical partition.
- Provided the configuration is suitable, a LV (such as `/usr`) can be enlarged when the free space is exhausted.

- Using LVM, even add hard disks or LVs in a running system. However, this requires hot-swappable hardware that is capable of such actions.
- Several hard disks can be used with improved performance in the RAID 0 (striping) mode.
- The snapshot feature enables consistent backups (especially for servers) in the running system.

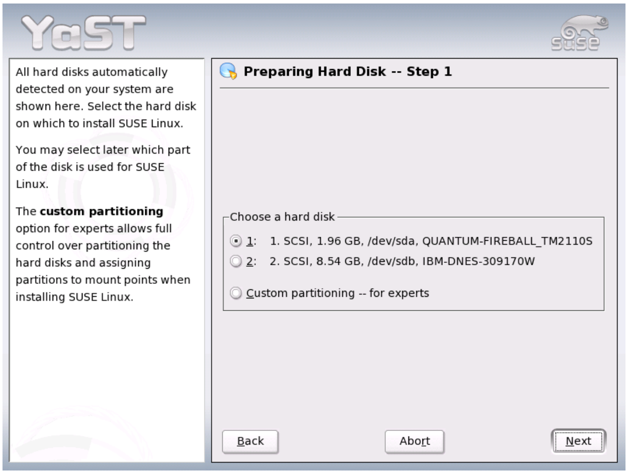
Implementing LVM already makes sense for heavily used home PCs or small servers. If you have a growing data stock, as in the case of databases, MP3 archives, or user directories, LVM is just the right thing for you. This would allow file systems that are larger than physical hard disk. Another advantage of the LVM is that up to 256 LVs can be added. Keep in mind that working with LVM is very different than 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/`.

3.10.2 LVM Configuration with YaST

Prepare the LVM configuration in YaST by creating an LVM partition when installing. To do this, click 'Partitioning' in the suggestion window then 'Discard' or 'Change' in the screen that follows. Next, create a partition for LVM by first clicking 'Add' → 'Do not format' in the Partitioner then clicking '0x8e Linux LVM'. Continue partitioning with LVM immediately afterwards or wait until after the system is completely installed. To do this, highlight the LVM partition in the partitioner then click 'LVM...'.

3.10.3 LVM — Partitioning

After selecting 'LVM...' in the partitioning section, continue automatically to a dialog in which to repartition your hard disks. Delete or modify existing partitions here or add new ones. A partition to use for LVM must have the partition label `8E`. These partitions are indicated by "Linux LVM" in the partition list.



All hard disks automatically detected on your system are shown here. Select the hard disk on which to install SUSE Linux.

You may select later which part of the disk is used for SUSE Linux.

The **custom partitioning** option for experts allows full control over partitioning the hard disks and assigning partitions to mount points when installing SUSE Linux.

Preparing Hard Disk -- Step 1

Choose a hard disk

● 1: 1. SCSI, 1.96 GB, /dev/sda, QUANTUM-FIREBALL_TM2110S

○ 2: 2. SCSI, 8.54 GB, /dev/sdb, IBM-DNES-309170W

○ Custom partitioning -- for experts

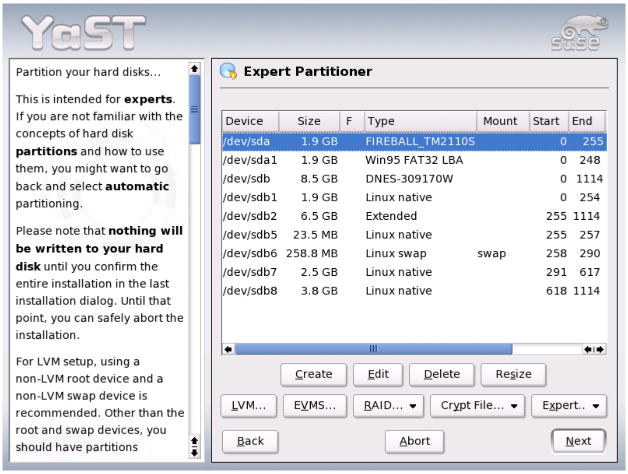
Figure 3.9: *Activating LVM During Installation*

Note

Repartitioning Logical Volumes

At the beginning of the physical volumes (PVs), information about the volume is written to the partition. In this way, a PV "knows" to which volume group it belongs. To repartition, it is advisable to delete the beginning of this volume. In VG "system" and PV "/dev/sda2", this can be done with the command `dd if=/dev/zero of=/dev/sda2 bs=512 count=1`.

Note



Partition your hard disks...

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

Please note that **nothing will be written to your hard disk** until you confirm the entire installation in the last installation dialog. Until that point, you can safely abort the installation.

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

Device	Size	F	Type	Mount	Start	End
/dev/sda	1.9 GB		FIREBALL_TM2110S		0	255
/dev/sda1	1.9 GB		Win95 FAT32 LBA		0	248
/dev/sdb	8.5 GB		DNES-309170W		0	1114
/dev/sdb1	1.9 GB		Linux native		0	254
/dev/sdb2	6.5 GB		Extended		255	1114
/dev/sdb5	23.5 MB		Linux native		255	257
/dev/sdb6	258.8 MB		Linux swap	swap	258	290
/dev/sdb7	2.5 GB		Linux native		291	617
/dev/sdb8	3.8 GB		Linux native		618	1114

Figure 3.10: *YaST: LVM Partitioner*

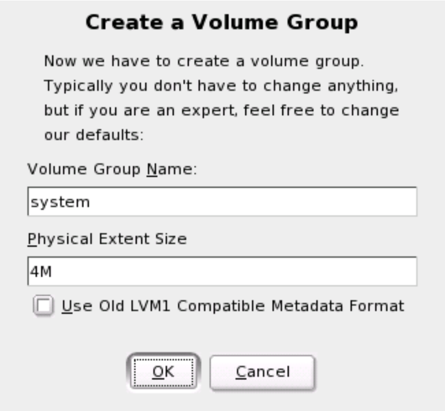
You do not need to set the 8E label for all partitions designated for LVM. If needed, YaST automatically sets the partition label of a partition assigned to an LVM volume group to 8E. For any unpartitioned areas on your disks, create LVM partitions in this dialog. These partitions should then be designated the partition label 8E. They do not need to be formatted and no mount point can be entered.

If a working LVM configuration already exists on your system, it is automatically activated as soon as you begin configuring the LVM. If this is successfully activated, any disks containing a partition belonging to an activated volume group can no longer be repartitioned. The Linux kernel will refuse to read the modified partitioning of a hard disk as long as only one partition on this disk is used.

Repartitioning disks not belonging to an LVM volume group is not a problem at all. If you already have a functioning LVM configuration on your system, repartitioning is usually not necessary. In this screen, configure all mount points not located on LVM Logical Volumes. The root file system in YaST must be stored on a normal partition. Select this partition from the list and specify this as root file system using the 'Edit' button. In view of the flexibility of LVM, we recommend that you place all additional file systems in LVM logical volumes. After specifying the root partition, exit this dialog.

3.10.4 LVM — Configuring Physical Volumes

In the dialog 'LVM', the LVM volume groups are managed. If no volume group exists on your system yet, you will be prompted to add one. `system` is suggested as a name for the volume group in which the SUSE LINUX system files are located. Physical extent size (PE Size) defines the maximum size of a physical and logical volume in this volume group. This value is normally set to four megabytes. This allows for a maximum size of 256 GB for physical and logical volumes. The physical extent size should only be increased if you need logical volumes larger than 256 GB (e.g., to 8, 16, or 32 MB).



Create a Volume Group

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

Volume Group Name: system

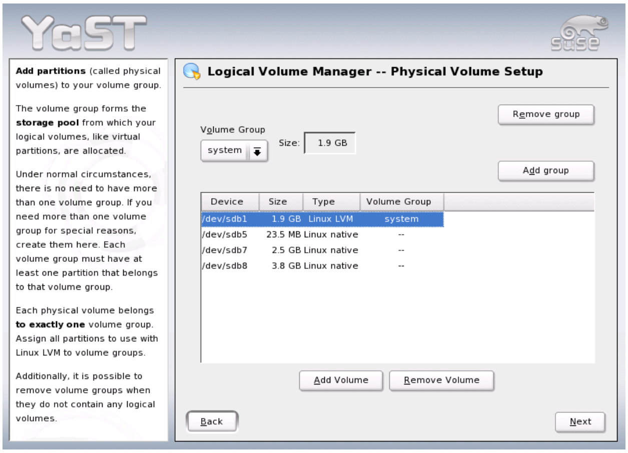
Physical Extent Size: 4M

☐ Use Old LVM1 Compatible Metadata Format

Figure 3.11: Adding a Volume Group

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 by "--".

The volume group currently being edited can be modified 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 to which no partitions are assigned can be deleted. No more than one volume group needs to be created for a normally installed SUSE LINUX system. A partition assigned to a volume group is also referred to as a physical volume (PV).



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

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

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

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

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

Logical Volume Manager -- Physical Volume Setup

Volume Group: system

Size: 1.9 GB

Device	Size	Type	Volume Group
/dev/sdb1	1.9 GB	Linux LVM	system
/dev/sdb5	23.5 MB	Linux native	--
/dev/sdb7	2.5 GB	Linux native	--
/dev/sdb8	3.8 GB	Linux native	--

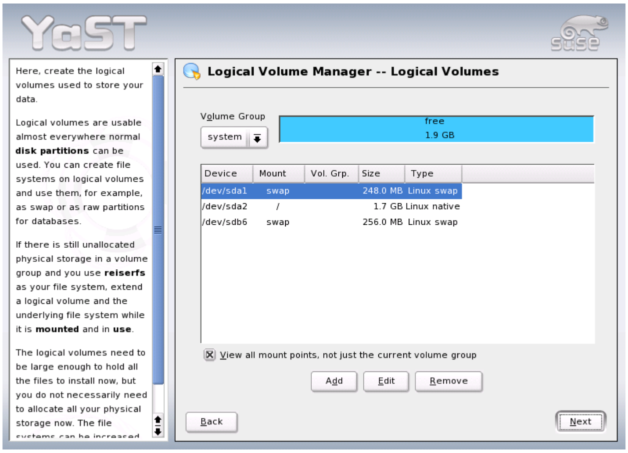
Figure 3.12: Partition List

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 will remain unused. Before exiting the dialog, every volume group must be assigned at least one physical volume.

3.10.5 Logical Volumes

This dialog is responsible for managing logical volumes. Assign one logical volume to each volume group. To create a striping array when you create the logical volumes, first create the LV with the largest number of stripes. 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 only two PVs are available, an LV with three stripes is impossible.

Normally, a file system is created on a logical volume (e.g., reiserfs, ext2) 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. All normal Linux partitions to which a mount point is assigned, all swap partitions, and all already existing logical volumes are listed here.


YaST
Logical Volume Manager -- Logical Volumes
Volume Group
system
free
1.9 GB
☒ View all mount points, not just the current volume group
Add
Edit
Remove
Back
Next

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

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

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

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

Device	Mount	Vol. Grp.	Size	Type
/dev/sda1	swap		248.0 MB	Linux swap
/dev/sda2	/		1.7 GB	Linux native
/dev/sdb6	swap		256.0 MB	Linux swap

Figure 3.13: Logical Volume Management

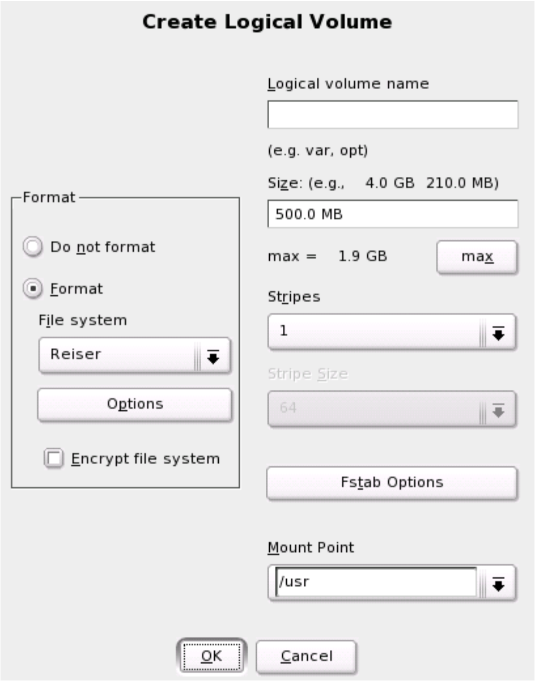
Caution

Using LVM might be associated with increased risk factors, 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.

Caution

If you have already configured LVM on your system, the existing logical volumes must be entered now. Before continuing, assign the appropriate mount point to these logical volumes. If you are configuring LVM on a system for the first time, no logical volumes are displayed in this screen yet. A logical volume must be generated for each mount point (using ‘Add’). Also set the size, the file system type (e.g., reiserfs or ext2), and the mount point (e.g., `/var/`, `/usr/`, `/home/`).

If you have created several volume groups, switch between individual volume groups by means of the selection list at the top left. Added logical volumes are listed in the volume group displayed there. After creating all the logical volumes required, exit the dialog. If you are still in the installation process, you can proceed with the software selection.



Create Logical Volume

Logical volume name

(e.g. var, opt)

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

500.0 MB

max = 1.9 GB

Stripes: 1

Stripe Size: 64

Mount Point: /usr

Format

○ Do not format

● Format

File system: Reiser

☐ Encrypt file system

Figure 3.14: Creating Logical Volumes

3.11 Soft RAID

The purpose of RAID (redundant array of inexpensive disks) is to combine several hard disk partitions into one large *virtual* hard disk for the optimization of performance and data security. Using this method, however, one advantage is sacrificed for another. *RAID level* defines the pool and common triggering device of the all hard disks, the RAID controller. A RAID controller mostly uses the SCSI protocol, because it can drive more hard disks better than the IDE protocol. It is also better able to process commands running in parallel.

Like a RAID controller, which can often be quite expensive, soft RAID is also able to take on these tasks. SUSE LINUX 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.

3.11.1 Common RAID Levels

RAID 0 This level improves the performance of your data access. 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 hard disks are pooled together. The performance is very good — although the RAID system will be destroyed and your data lost if even one of hard disks fails.

RAID 1 This level provides adequate security for your data, as 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. The writing performance suffers a little in the copying process when using RAID 1 (ten to twenty percent 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.

RAID 5 RAID 5 is an optimized compromise between the two other levels in terms of performance and redundancy. The hard disk space equals the number of disks used minus one. The data is distributed over the hard disks as with RAID 0. *Parity blocks*, created on one of the partitions, are there for security reasons. They are linked to each other with XOR — enabling the contents, via XDR, 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.

3.11.2 Soft RAID Configuration with YaST

Access Soft RAID configuration with the 'RAID' module under 'System' or via the partitioning module under 'Hardware'.

First Step: Partitioning

First, see a list of your partitions under 'Expert Settings' in the partitioning tool. If the Soft RAID partitions have already been set up, they appear here. Otherwise, set them up from scratch. 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 various 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.

Second Step: Setting up RAID

Click 'RAID' to open a dialog in which to choose between RAID levels 0, 1, and 5. In the following screen, assign the partition to the new RAID. 'Expert Options' opens the settings options for the *chunk size* — for fine-tuning the performance. Checking 'Persistent Superblock' ensures that the RAID partitions are recognized as such when booting. After completing the configuration, you will then see the `/dev/md0` device and others indicated with *RAID* on the expert page in the partitioning module.

3.11.3 Troubleshooting

Find out whether a RAID partition has been destroyed by the file contents `/proc/mdstats`. The basic procedure in case of system failure is to shut down your Linux system and replace the defective hard disk with a new one partitioned the same way. Then restart your system and give the `raidhotadd /dev/mdX /dev/sdX` command. This enables the hard disk to be integrated automatically into the RAID system and be fully reconstructed.

3.11.4 For More Information

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

- `/usr/share/doc/packages/raidtools/Software-RAID-HOWTO.html`
- `http://en.tldp.org/HOWTO/Software-RAID-HOWTO.html`

Linux RAID mailing lists are also available, such as `http://www.mail-archive.com/linux-raid@vger.rutgers.edu`.

3.12 Mass storage via IP networks – iSCSI

One of the central problems in computer centers and when operating servers is the provision of 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 PC servers are not connected to central storage solutions.

linux-iSCSI provides a simple and reasonably inexpensive solution for connecting Linux computers to central storage systems.

In principle, iSCSI represents a transfer of SCSI commands on 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. These packages are then transferred to the corresponding iSCSI remote station.

In order to use iSCSI you need the `linux-iscsi` package. The connection data must be entered in the `/etc/iscsi.conf` file. If you have an iSCSI storage device, this configuration file might look like this:

```
DiscoveryAddress=10.10.222.222
TargetName=iqn.1987-05.com.cisco:00.3b8334455c55.disk1
```

In this very simple example the storage system does not use authentication. Many properties of iSCSI can be set in `/etc/iscsi.conf`. You will find further details in the manual page for iSCSI.

After iSCSI has been configured, the iSCSI subsystem must be started. This is achieved with the `rciscsi start` command. The system should output the following messages:

```
rciscsi start
Starting iSCSI: iscsi iscsid fsck/mount                  done
```

The `/etc/initiatorname.iscsi` file is set up at the first initialization and will be used by the computer in the future to log on to iSCSI storage. This file cannot simply be copied. It must be created from scratch for every host.

If the start has been successful, then the system messages will indicate which devices have been recognized. This is achieved using the `dmesg` command. The various devices are now available under `/dev/sda` or `/dev/sdb` for example and can be partitioned and formatted as required. The mountpoints for the recognized devices should be entered in `/etc/fstab.iscsi`. These are executed when iSCSI is started.

Further publications relating to iSCSI can be found on the project website under `http://linux-iscsi.sourceforge.net/`

4

Central software installation and updating

If you want to install a whole pool of systems within a network with SUSE LINUX Enterprise Server, you can use YaST for the central provision of installation data and for the central management of software updates.

4.1 Setting up a central installation server 146
4.2 Managing software updates with the YOU server . 150
4.3 Booting From The Network 152

4.1 Setting up a central installation server

Instead of installing every individual computer to be configured with a set of installation media, you can also provide the installation data on a dedicated installation server in your network and fetch it from there for the purpose installing clients. The YaST installation server supports HTTP, FTP and NFS. With the help of the *Service Location Protocols* (SLP), this server can be made known to all clients in the network. This means that there is no need to select the installation source manually on the clients.

Note

Further information about SLP

Detailed information about SLP under SUSE LINUX is available in 21.6 on page 452.

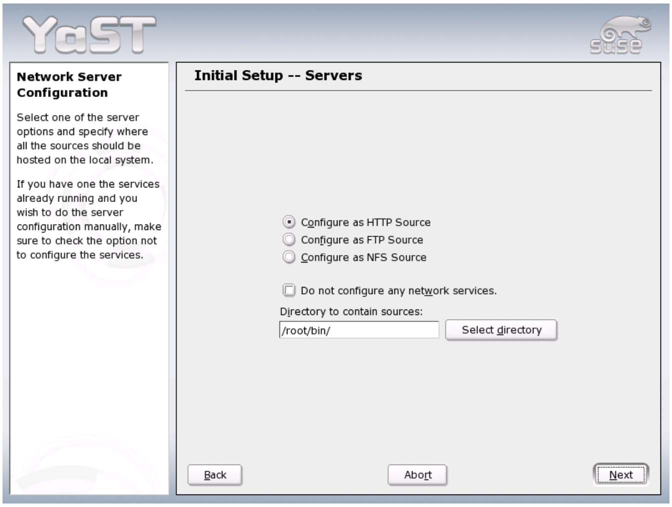
Note

4.1.1 Configuration with YaST

Start the 'Installation-Server' module from the 'Miscellaneous' menu. The new installation server will now be configured in four steps:

Selecting the server type YaST supports three types of installation server: HTTP, FTP and NFS. Select the server type by means of a radio button. From now on the selected server service will be started automatically every time you start up the system. If a service of the selected type is already running on your system, then deactivate the automatic configuration of the server service with YaST and configure the server service after you have completed YaST configuration itself. In both cases you should define the directory in which the installation data is to be made available on the server (see 4.1 on the next page).

Detailed configuration of the required server type
This step relates to the automatic configuration of server services. This dialog is skipped when deactivating automatic configuration (see above). Define an alias for the root directory of the FTP or HTTP server, where the installation data will be found. The installation source will later be located under `ftp://<Server-IP>/<Alias>/<Name>` (FTP) or under



Network Server Configuration

Select one of the server options and specify where all the sources should be hosted on the local system.

If you have one the services already running and you wish to do the server configuration manually, make sure to check the option not to configure the services.

Initial Setup -- Servers

● Configure as HTTP Source
○ Configure as FTP Source
○ Configure as NFS Source

☐ Do not configure any network services.

Directory to contain sources: /root/bin/

Figure 4.1: *YaST installation server: selecting the server type*

`http://<Server-IP>/<Alias>/<Name>` (HTTP). ⟨*Name*⟩ stands for the name of the installation source, which you define in the following step. If you have selected NFS in the previous step, you should define wildcards and `exports` options. The NFS server will be accessible under `nfs://<Server-IP>/<Name>`. Details of NFS and `exports` will be found under 21.10.4 on page 519.

Configuring the installation source

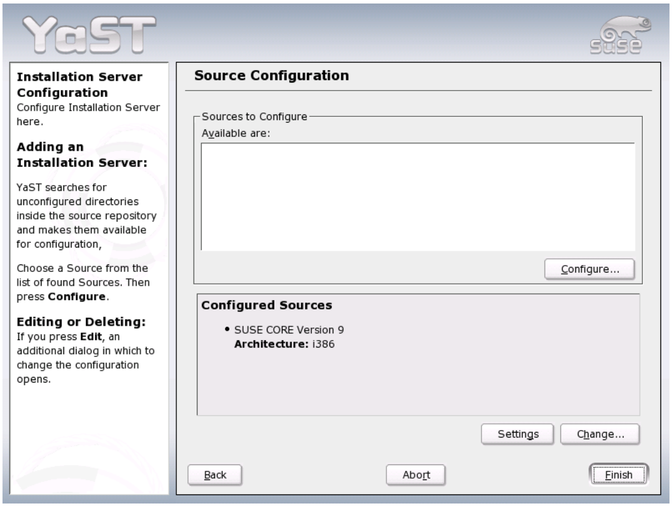
Before the installation media are copied to their destination, you should define the name of the installation source (ideally an easily remembered abbreviation of the product and version name). YaST allows you provide ISO images of the media instead of copies of the SUSE LINUX CDs. If you wish to take this route, activate the relevant checkbox and specify the directory path under which the ISO files will be found locally. Depending on which product you want to distribute using this installation server, it may be that you will require more add-on CDs or service pack CDs in order completely to install the product. If you activate the 'Prompt for additional CDs', YaST will automatically remind you to supply these media.

If you want to announce your installation server in the network via SLP, activate the relevant checkbox.

Uploading the installation data The most lengthy step in configuring an installation server is the copying of the actual SUSE LINUX 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, you will return to the overview of existing information sources and close the configuration by selecting 'Finish'.

Your configuration server is now fully configured and ready for service. It is automatically started every time the system is started. No further intervention is required from you. 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.

If your installation server is to provide the installation data for more than one product of product version, start the YaST installation server module and select 'Configure' in the overview of existing installation sources (see figure 4.2) to configure the new installation source.



Installation Server Configuration
Configure Installation Server here.

Adding an Installation Server:

YaST searches for unconfigured directories inside the source repository and makes them available for configuration,

Choose a Source from the list of found Sources. Then press **Configure**.

Editing or Deleting:
If you press **Edit**, an additional dialog in which to change the configuration opens.

Source Configuration

Sources to Configure
Available are:

Configured Sources

- SUSE CORE Version 9
  Architecture: i386

Figure 4.2: YaST Installation server: overview of installation sources

If you wish to deactivate an installation source, select the 'Change', button in the overview to reach a list of all available installation sources. Choose the entry you want to remove here and select 'Delete'. This delete procedure only relates to the deactivation of the server service. The installation data itself remains in the directory you have chosen. However you can remove it manually.

4.1.2 Client installation via the installation server

► **s390, zseries**
The use of installation media available via NFS or FTP to install SUSE LINUX Enterprise Server for IBM S/390 and zSeries is explained in the Architecture-specific Information manual in the *IPL of SUSE LINUX installation system* chapter. ◄

As soon as the installation server is available with the required installation data in the network, all computers in the local network can access the data. If a client is to be installed from scratch, all you need is a bootable medium to initialize the process. At the boot prompt — as described in 3.1.7 on page 114 — enter the name of the server from which the installation data is to be obtained:

```
install=<URL>
```

Afterwards your network interface will be automatically configured, preferably via DHCP. If this is not possible, perform manual configuration with linuxrc or specify the `HostIP` parameter at the boot prompt. The installation kernel is then started and YaST begins installation. Details of linuxrc can be found in 3.1 on page 106.

If your installation server is announced in the network via SLP, this will simplify the installation procedure:

1. Use function key (F3) and the cursor keys in the graphical splash screen to select the `SLP` option and confirm the selection with (Enter). Alternatively, enter `install=slp` at the boot prompt. In both cases, linuxrc will start an SLP inquiry for an installation server in the network.
2. Now select 'Installation' in the boot menu and confirm with (Enter). The installation kernel boots and YaST starts the installation. If several installation sources can be found with SLP, select the required source in linuxrc before YaST starts work.

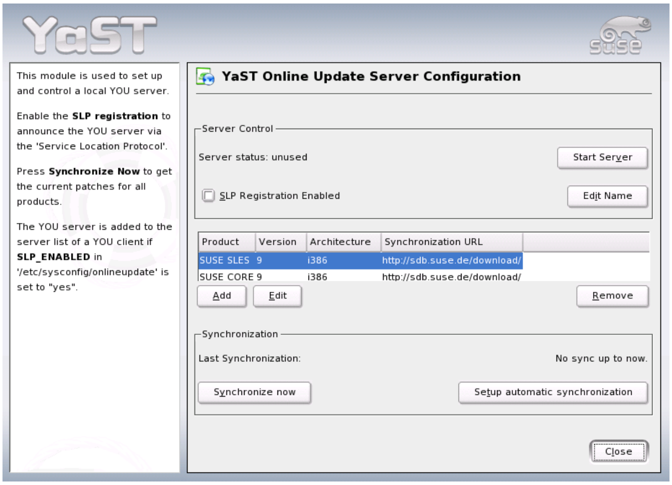
The rest of the installation procedure continues as described in the previous chapters. For detailed information about the SLP protocol and its applications see 21.6 on page 452.

4.2 Managing software updates with the YOU server

YaST uses the 'YOU server' module to allow users to employ their own local update server, which always has the current software updates available for all YOU clients contained in the network. This centralizes the updating of all systems contained in the network. The YOU server is either compared manually or automatically with one of the update servers in the Internet authorized by SUSE. Depending on the product, these are either the SUSE maintenance web (`http://sdb.suse.de/download`) or one of the mirrors of the SUSE FTP server (`ftp://ftp.suse.com/pub/suse`). The local clients call off the updates via HTTP. The YOU server can be configured so that it can be recognized via SLP (*Service Location Protocol*) and can be recognized automatically by all clients in the network.

4.2.1 Configuring the local YOU server

Start the 'Software' → 'YOU Server Configuration' module from the YaST control center. The dialog mask is divided into three areas: server control, product list and synchronization.


YaST
SUSE
YaST Online Update Server Configuration
Server Control
Start Server
SLP Registration Enabled
Edit Name
Add
Edit
Remove
Synchronization
Synchronize now
Setup automatic synchronization
Close

This module is used to set up and control a local YOU server.

Enable the **SLP registration** to announce the YOU server via the 'Service Location Protocol'.

Press **Synchronize Now** to get the current patches for all products.

The YOU server is added to the server list of a YOU client if **SLP_ENABLED** in '/etc/sysconfig/onlineupdate' is set to "yes".

Server status: unused

☐ SLP Registration Enabled

Product	Version	Architecture	Synchronization URL
SUSE SLES	9	i386	http://sdb.suse.de/download/
SUSE CORE	9	i386	http://sdb.suse.de/download/

Last Synchronization: No sync up to now.

Figure 4.3: The YOU server

The dialog part of the 'Server Control' offers an overview of the status of the YOU server and the configuration of its basic properties. 'Start Server' starts the YOU server. The web server (apache2) that distributes the updates to the subordinate servers via HTTP is installed, configured and started after this button is selected. If the 'SLP Registration' checkbox is activated, then the YOU server registers its service on the local SLP server. The 'Change Name' button allows you to change the name under which your local YOU server appears among the YOU clients.

The product list shows the names of all products for which the YOU server currently provides updates and indicates the URL from which the updates are derived. The product running on the machine on which the server is set up is displayed as the default. The 'Add', 'Change' and 'Delete' buttons are used to edit the product list. 'Add' and 'Change' call up a dialog in which the product name, hardware architecture, version and URL of origin are entered. In the case of update servers that require authentication (SUSE Maintenance Web), enter the user name and password in this dialog.

Note

Product data

The product name, version and architectural designation of the hardware are used internally by YOU to form the path under which the updates are searched for on the source server. Please make sure that the correct designations are used here, as otherwise YOU cannot find the required updates on the source server. You will receive the data either via the YOU client dialog or by entering `online_update -c` on the command line.

Note

The status of synchronization with the source server or the date of the last synchronization process is displayed in the synchronization area. The 'Synchronize Now' button activates a synchronization process. All updates for the listed products are downloaded and stored under `/var/lib/YaST2/you/mnt/`. From here the updates are made available to all associated YOU clients for installation purposes. The 'Configure Automatic Synchronization' button is used to call up a dialog in which you can conveniently automate synchronization by means of a Cronjob.

4.2.2 Configuring the clients

The local YOU clients that you call using 'YaST Control Center' → 'Online Update' should either be configured manually so that they obtain the updates from your YOU server or use the SLP functionality of YaST to determine the server address automatically.

Manual configuration Enter the URL that refers to your local server in the URL field of the YOU client: `http://<servername>/YOU` or, alternatively, enter `/etc/youservers` in this path.

Searching for servers via SLP Activate the SLP search on your YOU client by setting the `SLP_ENABLED` variable to `yes` in the `/etc/sysconfig/onlineupdate` file.

Clients can be configured by means of Cronjob so that you regularly search the local server for updates. Alternatively, you can activate the update procedure manually. There is a description of the YOU client in 2.3.2 on page 50.

4.3 Booting From The Network

To boot the system with a bootable network card, you need three services:

1. First of all you need a service which answers queries from the network card. On Intel-based computers, this is generally PXE. During the PXE boot process, first of all a special PXE boot image is loaded. This controls the boot process from this point onwards. Itanium Processor Family computers are started up for the network boot via elilo.
2. tftpd is responsible for making the kernel and the first system available. To configure tftpd, a YaST module is provided.
3. If you want to boot with the boot server for the installation, then you will need a third service, an installation server. This service is described in detail in section 4.1 on page 146.
4. With the aid of this service, you can also operate a system entirely without any local hard disks. In this case, it is recommended that the `root` file system is made available through NFS.

4.3.1 Configuration Of tftpd

The actual boot process entails two stages. The first boot image, which is loaded by the computer, varies according to the architecture.

► **x86**
The PXE image `pxelinux.0` is loaded by BIOS. This takes over control of the boot process from this point onwards. First of all PXE fetches a configuration file from the tftp server. ◄

► **ipf**
The computer firmware starts by loading the boot image `elilo.efi` from the tftp server. This then loads a configuration file from the tftp server, which controls the boot process from this point. ◄

First of all you need to create the main directory for tftpd. By definition, this is the `/tftpboot/` directory:

```
mkdir /tftpboot
```

Preparing tftp For PXE Boot

The boot image which is needed to operate PXE can be found in the `syslinux` package under `/usr/share/syslinux/pxelinux.0`. Copy this file to the `/tftpboot/` directory:

```
cp /usr/share/syslinux/pxelinux.0 /tftpboot
```

The configuration file for PXE is stored in the `/tftpboot/pxelinux.cfg/` directory. Here it is possible to create a standard configuration file for all computers which are to be booted. However, it is also possible to use a separate configuration file for each IP address. If, for example, you want to create a separate configuration file for the IP address 192.168.0.0, you can find out its name with the command `gethostip`. If no special configuration file is found, then PXE will try to open a file called `default`.

In the configuration file for PXE, a number of options are available. Typically, a configuration file for the installation of a computer looks like this:

```
default linux
label linux
  kernel linux
  append initrd=initrd ramdisk_size=65536 install=slp:

implicit        0
```

```
display         message
prompt          1
timeout         200
notice          2
```

For this configuration to work, you need to copy the `linux` and `initrd` files from the first installation CD to the `/tftpboot/` directory. On the CD, you will find these files under `/boot/loader/`. You also need to define an installation source which is available through SLP. The procedure for this is described in section 4.1 on page 146.

Preparing tftp For Booting Itanium Processor Family Computers

The image that is required for network booting of Itanium Processor Family computers is contained on the first CD for SUSE LINUX Enterprise Server, in the `/boot/image` file. To prepare for the boot process, it is necessary to extract file `bootia64.efi` from this image and copy it as `elilo.efi` to the `/tftpboot/` directory. To launch the boot process, files `linux`, `initrd`, `textmenu` and `elilo.conf` must also be copied to `/tftpboot/`. Assuming that the first CD is available in the `/media/cdrom/` directory, you can achieve this by entering the following commands:

```
mount -o loop,ro /media/cdrom/boot/image /mnt
cd /mnt/efi/boot
cp -p linux initrd textmenu elilo.conf /tftpboot
cp -p bootia64.efi /tftpboot/elilo.efi
umount /mnt
```

The final step is to now supplement the `append` lines in file `/tftpboot/elilo.conf` by adding another parameter, `install=slp:`. This tells the installation system that it should search for its installation source using the "SLP" protocol.

4.3.2 Configuration Of dhcpd

dhcpd is responsible for telling the computer where it can find the boot image. Moreover, the computer, which functions as a network client, is assigned an IP address. To configure dhcpd, a YaST module is provided. You tell a client the location of the boot image with the aid of the parameter `filename`.

► **x86, amd64, em64t**

```
filename "pxelinux.0";
```

◀

▶ **ipf**

```
filename "elilo.efi";
```

◀

If tftpd is not running on the same server as dhcpd, then you will have to also enter the address of the tftpd server in the configuration:

```
next-server sun
```

4.3.3 Launching The Boot Process

During the boot process, the computer's BIOS automatically searches for a boot source. The dhcp server answers this query and passes across the data necessary to boot with either PXE or elilo.

Before the actual boot process gets under way, a "boot:" prompt is displayed, following which, if necessary, you can enter any additional parameters for kernel and linuxrc.

Updating the System and Package Management

SUSE LINUX provides the option of updating an existing system without completely reinstalling it. There are two types of updates: *updating individual software packages* and *updating the entire system*. Packages can also be installed by hand using the package manager RPM.

5.1 Updating SUSE LINUX 158
5.2 Software Changes from Version to Version 162
5.3 RPM — the Package Manager 168

5.1 Updating SUSE LINUX

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, the software selected, and the version numbers of SUSE LINUX.

Note

Read the `README` file on the CD. This file contains any changes made *after* this manual went to print.

Note

5.1.1 Preparations

Before updating, copy the old configuration files to a separate medium (streamer, removable hard disk, 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`. Furthermore, you may want to write the user data in `/home` (the `HOME` directories) to a backup medium. Back up this data as `root`. Only `root` has read permission for all local files.

Before starting your update, make note of the root partition. The command `df /` lists the device name of the root partition. In Example 5.1, the root partition to write down is `/dev/hda2` (mounted as `/`).

***Example 5.1:** List with df -h*

```
Filesystem  Size  Used  Avail Use% Mounted on
/dev/hda1   1.9G  189M  1.7G  10%  /dos
/dev/hda2   8.9G  7.1G  1.4G  84%  /
/dev/hda5   9.5G  8.3G  829M  92%  /home
```

5.1.2 Possible Problems

Checking passwd and group in /etc

Before updating the system, make sure `/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, of course, only necessary if you actually used PostgreSQL prior to your update.

x86: Promise Controller

The hard disk controller manufactured by Promise is currently found on high-end motherboards in numerous computer models, either as a pure IDE controller (for UDMA 100) or as an IDE-RAID controller. As of SUSE LINUX 8.0, these controllers are directly supported by the kernel and treated as a standard controller for IDE hard disks. The additional kernel module `pdcraid` is required for RAID functionality.

For some updates, hard disks on the Promise controller may be detected before disks on the standard IDE controller. If so, the system no longer boots following a kernel update and usually exits with `Kernel panic: VFS: unable to mount root fs`. In this case, the kernel parameter `ide=reverse` must be passed when booting to reverse this disk detection process. To apply this parameter permanently when using YaST, enter it in the boot configuration.

Caution

Only the controllers activated in the BIOS are detectable. In particular, subsequently activating or deactivating the controllers in the BIOS has a direct effect on the device names. Use caution or risk being unable to boot the system.

Caution

Technical Explanation

The controller sequence depends on the motherboard. Each manufacturer wires its supplementary controllers differently. The `lspci` shows this sequence. If the Promise controller is listed before the standard IDE controller, the kernel parameter `ide=reverse` is required after updating. With the previous kernel (without direct Promise support), the controller was ignored so the standard IDE controller was detected first. The first disk was then `/dev/hda`. With the new kernel, the Promise controller is detected immediately and its (up to four) disks are registered as `/dev/hda`, `/dev/hdb`, `/dev/hdc`, and `/dev/hdd`. The previous `/dev/hda` disk becomes `/dev/hde` so is no longer detectable in the boot process.

5.1.3 Updating with YaST

Following the preparation procedure outlined in Section 5.1.1 on page 158, you can now update your system:

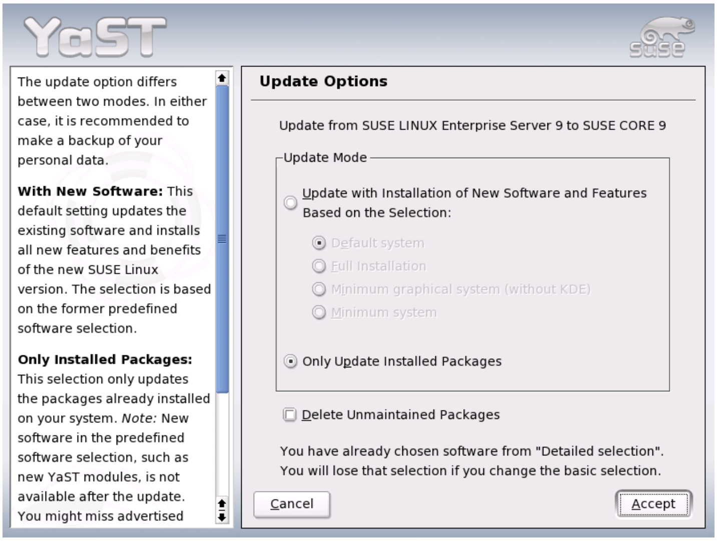
1. Boot the system as for the installation. In YaST, choose a language and select 'Update Existing System'. Do not select 'New Installation'.
2. YaST determines whether there are multiple root partitions. If there is only one, continue with the next step. If there are several, select the right partition and confirm with 'Next' (`/dev/hda7` was selected in the example in Section 5.1.1 on page 158). YaST reads the *old* `fstab` on this partition to analyze and mount the file systems listed there.
3. Then you have the possibility to make a backup copy of the system files during the update. This option slows down the update process. Use this option if you do not have a recent system backup.
4. In the following dialog, either choose to update only the software that is already installed or to add new software components to the system (upgrade mode). It is advisable to accept the suggested composition (e.g., 'Standard System'). Adjustments can be made later with YaST.

5.1.4 Manual Update

Updating the Base System

As basic system components, such as libraries, must be exchanged when updating a base system, an update cannot be run from within a currently running Linux system. First, set up the update environment. This is normally done using the CD or DVD or with a custom boot disk. If you are carrying out manual modifications during the update or prefer to perform the entire update with YaST in text mode, follow the steps described in Section 3.3 on page 116. Below is a summary of this procedure.

1. Immediately after booting the kernel from the boot disk or from the CD or DVD, linuxrc automatically starts.
2. In linuxrc, specify the language and keyboard settings under 'Settings' and click 'OK' to confirm each setting.
3. You might need to load the required hardware and software drivers via 'Kernel Modules'. See Section 3.1 on page 106 for more details of how to proceed and Section 3.1.3 on page 109 for a description of linuxrc.



The update option differs between two modes. In either case, it is recommended to make a backup of your personal data.

With New Software: This default setting updates the existing software and installs all new features and benefits of the new SUSE Linux version. The selection is based on the former predefined software selection.

Only Installed Packages: This selection only updates the packages already installed on your system. *Note:* New software in the predefined software selection, such as new YaST modules, is not available after the update. You might miss advertised

Update Options

Update from SUSE LINUX Enterprise Server 9 to SUSE CORE 9

Update Mode

○ Update with Installation of New Software and Features Based on the Selection:

- ● Default system
- ○ Full Installation
- ○ Minimum graphical system (without KDE)
- ○ Minimum system

● Only Update Installed Packages

☐ Delete Unmaintained Packages

You have already chosen software from "Detailed selection". You will lose that selection if you change the basic selection.

Figure 5.1: Updating the Software

4. Go to 'Start Installation or System' → 'Start Installation or Update' to select the source medium (see Section 3.1.5 on page 111).
5. The installation environment is loaded from linuxrc then YaST starts.

Following the selection of a language and the hardware detection by YaST, select 'Update Existing System' in the YaST opening screen.

Next, YaST attempts to determine the root partition and displays the result for selection or confirmation. Select your root partition from the list (example: `/dev/hda2`). In this way, prompt YaST to read the *old* `fstab` from this partition. YaST analyzes and mounts the file systems listed there.

Then you have the possibility to make a backup copy of the system files during the update. In the following dialog, either choose to update only the software already installed or to add important new software components to the system (*upgrade mode*). It is advisable to accept the suggested composition (e.g., 'Standard system'). Adjustments can be made later with YaST.

In the warning dialog, select 'Yes' to start the installation of the new software from the source medium to the system hard disk. First, the RPM database is checked, then the main system components are updated. YaST automatically creates backups of files modified in the running system since the last installation. In addition, old configuration files are backed up with the endings `.rpmorig` and `.rpmsave`. The installation or update procedure is logged in `/var/log/YaST2/y2log*` and can be viewed later at any time.

Updating the Rest of the System

After the base system is updated, you are switched to YaST's update mode. This mode allows you to tailor the rest of the system update to your needs. Complete the procedure as you would a new installation. Among other things, select a new kernel. The available options are presented by YaST.

Possible Problems

If certain shell environments no longer behave as expected after the update, check to see if the current *dot* files in the home directory are still compatible with your system. If not, use the current versions in `/etc/skel`. For example, `cp /etc/skel/.profile  /.profile`.

5.2 Software Changes from Version to Version

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

5.2.1 From SLES8 to SLES9

Upgrading to Kernel 2.6

SUSE LINUX is now based entirely on kernel 2.6. The predecessor version 2.4 should no longer be used, as the enclosed applications may not work with kernel 2.4. Moreover, note the following details:

- The loading of modules is now configured by means of the file `/etc/modprobe.conf`. The file `/etc/modules.conf` is obsolete. YaST will try to convert the file (see also script `/sbin/generate-modprobe.conf`).
- Modules now have the suffix `.ko`.
- The module `ide-scsi` is no longer needed for burning CDs.
- The prefix `snd_` has been removed from the ALSA sound module options.
- `sysfs` now complements the `/proc` file system.
- Power management (especially ACPI) has been improved and can now be configured by means of a YaST module.

Mounting VFAT Partitions

When mounting VFAT partitions, the parameter `code=` must be changed to `codepage=`. If you have difficulties mounting a VFAT partition, check if the file `/etc/fstab` contains the old parameter name.

Native POSIX Thread Library and glibc 2.3.x

Applications linked against NGPT (*Next Generation POSIX Threading*) do not work with glibc 2.3.x. All affected applications that are not shipped with SUSE LINUX must be compiled with linuxthreads or with NPTL (*Native POSIX Thread Library*). NPTL is preferred, as this is the standard for the future.

If NPTL causes difficulties, the older linuxthreads implementation can be used by setting the following environment variable (replace ⟨*kernel-version*⟩ with the version number of the respective kernel):

```
LD_ASSUME_KERNEL=kernel-version
```

The following version numbers are possible:

2.2.5 (i386, i586): linuxthreads without floating stacks

2.4.1 (AMD64, IPF, s390x, i586, i686):
linuxthread with floating stacks

Notes regarding the kernel and linuxthreads with floating stacks:

Applications using `errno`, `h_errno`, and `_res` must include the header files (`errno.h`, `netdb.h`, and `resolv.h`) with `#include`. For C++ programs with multithread support that use *thread cancellation*, the environment variable `LD_ASSUME_KERNEL=2.4.1` must be used to prompt the use of the linuxthreads library.

Adaptions for Native POSIX Thread Library

NPTL (*Native POSIX Thread Library*) is included as the thread package. NPTL is binary-compatible with the older linuxthreads library. However, areas in which linuxthreads violates the POSIX standard require NPTL adaptions. This includes the following: signal handling, `getpid` returns the same value in all threads, and thread handlers registered with pthread_atfork do not work if `vfork` is used.

Network Interface Configuration

The configuration of the network interface has changed. Formerly, the hardware was initialized following the configuration of a non-existing interface. Now, the system searches for new hardware and initializes it immediately, enabling the configuration of the new network interface.

New names have been introduced for the configuration files. As the name of a network interface is generated dynamically and the use of hotplug devices is increasing steadily, a name like `eth`⟨*X*⟩ is no longer suitable for configuration purposes. For this reason, unique designations, like the MAC address or the PCI slot, are now used for naming interface configurations. Of course, you can use interface names as soon as they appear. Commands like `ifup eth0` or `ifdown eth0` are still possible.

The device configurations are located in `/etc/sysconfig/hardware`. The interfaces provided by these devices are usually located in `/etc/sysconfig/network` (with different names).

See the detailed description in `/usr/share/doc/packages/sysconfig/README`.

Top-Level Domain .local as link-local Domain

The resolver library treats the top-level domain `.local` as "link-local" domain and sends multicast DNS queries to the multicast address `224.0.0.251`, port `5353`, instead of normal DNS queries. This is an incompatible change. If the domain `.local` is already used in the name server configuration, use a different domain name. For more information about multicast DNS, see `http://www.multicastdns.org`.

Systemwide UTF-8 Encoding

Currently, the default encoding for the system is `UTF-8`. Thus, when performing a standard installation, a locale is set with `.UTF-8` encoding (e.g., `en_US.UTF-8`). For more information, see `http://www.suse.de/~mfabian/suse-cjk/locales.html`.

Converting File Names to UTF-8

Files in previously created file systems do not use UTF-8 encoding for the file names (unless specified otherwise). If these files names contain non-ASCII characters, they will be garbled. To correct this, use the `convmv` script which converts the encoding of file names to UTF-8.

Shell Tools Compatible with POSIX Standard of 2001

In the default setting, shell tools from the `coreutils` package (`tail`, `chown`, `head`, `sort`, etc.) no longer comply with the POSIX standard of 1992 but with the POSIX standard of 2001 (*Single UNIX Specification, version 3 == IEEE Std 1003.1-2001 == ISO/IEC 9945:2002*). The old behavior can be forced with an environment variable:

```
_POSIX2_VERSION=199209
```

The new value is 200112 and is used the default for `_POSIX2_VERSION`. The SUS standard can be reviewed at the following URL (free of charge, but registration is required):

`http://www.unix.org`

Table 5.1: *Comparison POSIX 1992 vs. POSIX 2001*

POSIX 1992	POSIX 2001
chown tux.users	chown tux:users

tail +3	tail -n +3
head -1	head -n 1
sort +3	sort -k +3
nice -10	nice -n 10
split -10	split -l 10

Note

Third-party software may not yet comply with the new standard. In this case, set the environment variable as described above: `_POSIX2_VERSION=199209`.

Note

/etc/gshadow Obsolete

`/etc/gshadow` has been abandoned and removed, as this file is superfluous for the following reasons:

- It is not supported by glibc.
- There is no official interface for this file; even the shadow suite does not contain such an interface.
- Most tools that check the group password do not support the file and ignore it for the said reasons.

OpenLDAP

- As the database format has changed, the databases must be regenerated. During the update, the system attempts to perform this conversion automatically. However, there will certainly be cases in which the conversion fails.
- The schema check has undergone substantial improvement. Therefore, a number of (non-standard compliant) operations that were possible with the former LDAP server are no longer possible.
- The syntax of the configuration file has partly changed with a view to ACLs.

Following the installation, further information regarding the update is available in the file `/usr/share/doc/packages/openldap2/README.update`.

Apache 1.3 Replaced with Apache 2

The Apache web server (version 1.3) has been replaced with Apache 2. On a system with an HTTP server installation, an update will remove the Apache package and install Apache 2. Subsequently, the system must be adapted with YaST or manually. The configuration files in `/etc/httpd` are now located in `/etc/apache2`. Apache 2 needs the `apache2-prefork` package (recommended for stability) or the `apache2-worker` package.

From samba 2.x to samba 3.x

Following the update from samba 2.x to samba 3.x, `winbind` authentication is no longer available. The other authentication methods can continue to be used. For this reason, the following programs have been removed:

```
/usr/sbin/wb_auth
/usr/sbin/wb_ntlmauth
/usr/sbin/wb_info_group.pl
```

See also `http://www.squid-cache.org/Doc/FAQ/FAQ-23.html#ss23.5`.

OpenSSH Update (Version 3.8p1)

`gssapi` support has been replaced with `gssapi-with-mic` to prevent potential MITM attacks. These two versions are not compatible. This means that you cannot authenticate with Kerberos tickets from older distributions, as other authentication methods are used.

SSH and Terminal Applications

When establishing a connection from a remote host (especially via SSH, telnet, and RSH) between version 9 (standard configuration with activated UTF-8) and older systems (SLES, SLES 8, or SUSE LINUX 9.0 and earlier versions in which UTF-8 is not activated by default or not supported), terminal applications may display faulty characters.

This is because OpenSSH does not forward local settings. Therefore, the default system settings that may not match the remote terminal settings are used. This affects YaST in text mode and applications executed from a remote host as a normal user (not `root`). The applications started by `root` are only affected if the user changes the standard locales for `root` (only `LC_CTYPE` is set by default).

libiodbc Discarded

Users of FreeRADIUS must now link against unixODBC, as libiodbc has been discarded.

XML Resources in /usr/share/xml

FHS (see Section 10.1.2 on page 238) now requires XML resources (DTDs, stylesheets, etc.) to be installed in `/usr/share/xml`. Therefore, some directories are no longer available in `/usr/share/sgml`. If you encounter problems, modify your scripts or makefiles or use the official catalogs (especially `/etc/xml/catalog` or `/etc/sgml/catalog`).

Removable Media with subfs

Removable media are now integrated with subfs. Media no longer need to be mounted manually with `mount`. The command `cd /media/*` launches the automatic mounting process. Media cannot be ejected as long as they are accessed by a program.

Printer Configuration

Information about the changes in the print system is available in Section 13.1 on page 290.

5.3 RPM — the Package Manager

In SUSE LINUX, RPM (Red Hat Package Manager) is used for managing the software packages. Its main programs are `rpm` and `rpmbuild`. The powerful RPM database can be queried by the users, the 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 checks 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`.

`rpm` can be used to administer LSB-compliant packages. Refer to Section 10.1.1 on page 238 for more information about LSB.

Note

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 `kdelibs-devel`.

Note

5.3.1 Verifying Package Authenticity

SUSE LINUX RPM packages have a GnuPG signature:

```
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 apache-1.3.12.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/`. Since version 8.1, the key is additionally located in the directory `/usr/lib/rpm/gnupg/` to enable normal users to verify the signature of RPM packages.

5.3.2 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 will be 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.

```
rpm -F <package>.rpm
```

This command will remove the files of the old version and immediately install 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` will install 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.

Note

Following every update, check the backup copies with the extension `.rpmorig` or `.rpmsave`, which are created by `rpm`. These are your old configuration files. If necessary, adopt your modifications from the backup copies to the new configuration file then delete the old files with the extension `.rpmorig` or `.rpmsave`.

Note

To remove a package, enter `rpm -e ⟨package⟩`. `rpm` will only delete 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`.

5.3.3 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 small bugs could easily result in large amounts of data. However, since SUSE 8.1, the SUSE RPM offers a new 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 appear 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 `man rpm` and in `man rpmbuild`.

5.3.4 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 5.2).

Table 5.2: *The Most Important RPM Query Options (-q [-p] . . . package)*

`-i`	Package information
`-l`	File list
`-f` ⟨*FILE*⟩	Query a package owned by ⟨*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 Output 5.2.

Example 5.2: *rpm -q -i wget*

```
Name        :wget                   Relocations: (not relocateable)
Version     :1.8.2                  Vendor: SuSE Linux AG, Nuernberg, Germany
Release     :301                    Build Date: Di 23 Sep 2003 20:26:38 CEST
Install date:Mi 08 Okt 2003 11:46:31 CEST Build Host: levi.suse.de
Group       :Productivity/Networking/Web/Utilities
Source RPM  :wget-1.8.2-301.src.rpm
Size        :1333235                          License: GPL
Signature   :DSA/SHA1, Di 23 Sep 2003 22:13:12 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 file name with its full path. Provide as many file names as desired. For example, the following command

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

results in:

```
rpm-3.0.3-3
wget-1.5.3-55
```

If only part of the file name is known, use a shell script as shown in File 5.3. Pass the partial file name to the script shown as a parameter when running it.

Example 5.3: *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 information (updates, configuration, modifications, etc.) about a specific package. This example shows information about the package `rpm`. However, only the last five change entries in the RPM database are listed. All entries (dating back the last two years) are included in the package itself. This query only works if CD 1 is mounted at `/media/cdrom/`:

```
rpm -qp --changelog /media/cdrom/suse/i586/rpm-3*.rpm
```

With the help of the installed RPM database, verification checks can be made. These checks are initiated with the option `-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 5.3: 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. 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 3 MB for 1 GB in `/usr`.

5.3.5 Installing and Compiling Source Packages

All source packages of SUSE LINUX carry a `.src.rpm` extension (source RPM).

Note

Source packages can be installed with YaST, like any other package. They will not, however, be 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.

Note

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.gz` files, etc.) and for distribution-specific adjustments (`.dif` 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 will be installed in `/usr/src/packages/`: the sources and the adjustments in `SOURCES/` and the relevant `.spec` file in `SPECS/`.

Caution

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

Caution

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

```
/usr/src/packages/SPECS/wget.spec
/usr/src/packages/SOURCES/wget-1.4.5.dif
/usr/src/packages/SOURCES/wget-1.4.5.tar.gz
```

`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 Allows skipping specific 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.

5.3.6 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. The respective position is specified with `build --rpms <path>`. Unlike `rpm`, the `build` command looks for the SPEC file in the source directory. To build wget anew (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 will be established at `/var/tmp/build-root`. The package will be 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 can be accessed with `build --help` and `man build`.

5.3.7 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: the `HEADER` information can be displayed with F3, the archive structure can be viewed with the cursor keys and Enter, and archive components can be copied with F5.

A front-end for rpm is also available for Emacs. KDE offers the kpackage tool. GNOME offers gnorpm.

Using the Alien (`alien`) Perl script, it is possible to convert or install an *alien* binary package. This tries to convert *old* TGZ archives to RPM before installing. This way, the RPM database can keep track of such a package after it has been installed. Beware: `alien` is still *alpha* software, according to its author — even if it already has a high version number.

System Repair

In addition to numerous YaST modules for system installation and configuration, SUSE LINUX Enterprise Server also offers a feature for repairing the installed system. This chapter describes the various types and steps of system repair.

6.1 Starting YaST System Repair 180
6.2 Automatic Repair 180
6.3 User-Defined Repair 181
6.4 Expert Tools 182
6.5 Using initrd as a Rescue System 183

6.1 Starting YaST System Repair

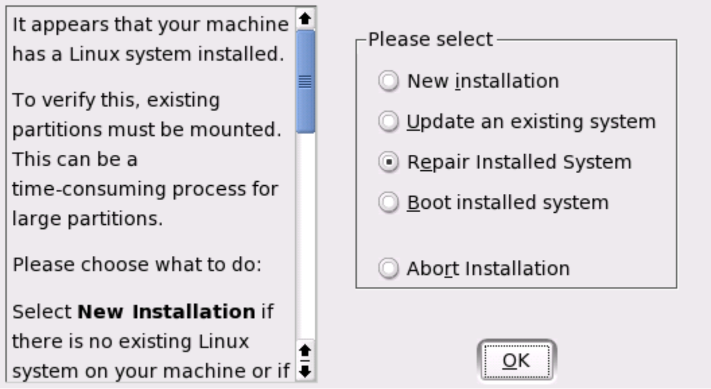
Because it cannot be assumed that a damaged system can boot by itself and a running system cannot be easily repaired, the YaST System Repair utility is run from the SUSE LINUX installation CD or DVD. Follow the steps outlined in Chapter 1 on page 7 to get to the dialog page offering the various installation options and select 'Repair Installed System'. See Figure 6.1.

Note

Using the Appropriate Installation Medium

Because the test and repair procedure is loaded from CD or DVD, it is essential to run it from an installation medium that *exactly* corresponds to your installed version of SUSE LINUX.

Note



It appears that your machine has a Linux system installed.

To verify this, existing partitions must be mounted. This can be a time-consuming process for large partitions.

Please choose what to do:

Select **New Installation** if there is no existing Linux system on your machine or if

Please select

○ New installation
○ Update an existing system
● Repair Installed System
○ Boot installed system
○ Abort Installation

Figure 6.1: Selecting the YaST System Repair Utility

In the next step, choose how the system repair should be performed. Automatic repair, custom repair, and expert tools are available and are described below.

6.2 Automatic Repair

This method is best suited to restoring a damaged system with unknown cause. Selecting it starts an extensive analysis of the installed system, which will take quite some time due to the large amount of tests and examinations. 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 process. The log window above allows tracking of the currently running activity and its test result. See Figure 6.2 on the next page. The following main test runs are performed with every run. They contain, in turn, a number of individual subtests.

Partition tables of all hard disks The validity and coherence of the partition tables of all detected hard disks are checked.

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 consistence. All valid partitions are mounted.

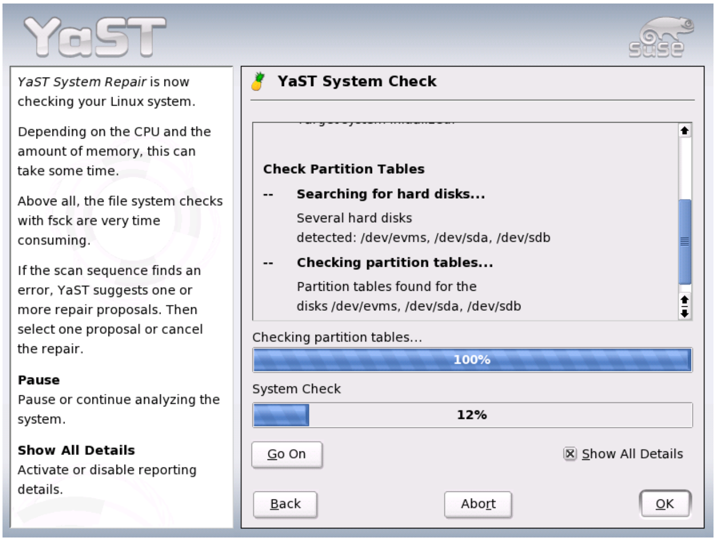
Boot loader configuration The boot loader configuration of the installed system (GRUB or LILO) is checked for completeness and coherence. Boot and root devices are examined and the availability of the initrd modules is checked.

Package database This checks whether all packages necessary for the operation of a minimal installation are present. While it is optionally possible to also analyze the base packages, this takes a long time because of their vast number.

Whenever an error is encountered, the procedure is stopped and a dialog opens, offering details and possible solutions. It is not possible to describe all these cases. Read the notifications on the screen carefully and choose the desired action from the list options. It is also possible to decline the offered repair action in cases of doubt. The system remains unaltered in this aspect and no repair is ever performed automatically and without prompting the user.

6.3 User-Defined Repair

The automatic repair explained in the preceding section categorically performs all tests. This is useful if the extent of the system damage is unkown.



YaST System Repair is now checking your Linux system.

Depending on the CPU and the amount of memory, this can take some time.

Above all, the file system checks with fsck are very time consuming.

If the scan sequence finds an error, YaST suggests one or more repair proposals. Then select one proposal or cancel the repair.

Pause
Pause or continue analyzing the system.

Show All Details
Activate or disable reporting details.

YaST System Check

Check Partition Tables

-- **Searching for hard disks...**
Several hard disks detected: /dev/evms, /dev/sda, /dev/sdb

-- **Checking partition tables...**
Partition tables found for the disks /dev/evms, /dev/sda, /dev/sdb

Checking partition tables...
100%

System Check
12%

Figure 6.2: *Automatic Repair Mode*

However, if you already know what part of the system is affected, the range of the applied tests can be narrowed. Choosing ‘User-defined 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 ‘Continue’ then starts a narrower test procedure that probably has a significantly shorter running time.

Not all test groups are applicable individually. The analysis of the fstab entries is always bound to an examination of the file systems, including existing swap partitions. YaST automatically satisfies such dependencies by selecting the smallest amount of necessary test runs.

6.4 Expert Tools

If you are knowledgeable with SUSE LINUX and already have a very clear idea of what needs to be repaired in your system, directly apply the tools necessary for repairing it by choosing ‘Expert tools’.

Install new boot loader This starts the YaST boot loader configuration module. Details can be found in Section 8.6 on page 218.

Run partitioning tool This starts the expert partitioning tool in YaST. Details can be found in Section 1.7.5 on page 23.

Fix file system This allows checking 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.

Restore lost partitions It is possible to attempt a reconstruction of 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.

Note

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 upon successful recognition. This is, however, not successful in all imaginable cases.

Note

Save system settings to disk This option allows saving important system files to a floppy disk. Should one of these files become damaged, it can be restored from disk.

Check 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.

6.5 Using initrd as a Rescue System

If the kernel of the SUSE LINUX Enterprise Server for S/390 and zSeries 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. This chapter describes a method of using the standard installation package as a rescue system from which the zipl program can be executed to update the IPL record.

6.1 Starting YaST System Repair 180
6.2 Automatic Repair 180
6.3 User-Defined Repair 181
6.4 Expert Tools 182
6.5 Using initrd as a Rescue System 183

6.5.1 IPLing the Rescue System

Note

For this method to work, the SUSE LINUX Enterprise Server for S/390 and zSeries installation data must be available. For details, refer to Chapter *Making the Installation Data Available* 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.

Note

First, IPL the SUSE LINUX Enterprise Server for S/390 and zSeries installation system as described in Architecture-specific Information. A list of choices for the network adapter to use is then presented.

Example 6.1: *Network Type Selection*

```
=                                                                 =
==-  Welcome to SuSE Linux Enterprise Server 9 for zSeries      -==
=                                                                 =

Please select the type of your network device:
 0) no network
 1) OSA Token Ring
 2) OSA Ethernet
 3) OSA-Gigabit Ethernet or OSA-Express Fast Ethernet
 4) Channel To Channel
 5) Escon
 6) IUCV
 8) Hipersockets
 9) OSA High speed Token Ring (qdio)
10) Show subchannels and detected devices
Enter your choice (0-10):
```

Select `0` for *no network*. The installation program is terminated and the following messages are printed:

Example 6.2: *Output for No Network Adapter Selected*

```
*** OK, NETWORK ACCESS WILL _NOT_ BE AVAILABLE. ***


You should be able to login via telnet/ssh now.
To restart network setup, enter:

netsetup

To continue the installation, enter:

./inst_source

bash: no job control in this shell
SuSE Instsys suse:/ #
```

This opens a root shell from which to issue all necessary commands directly.

6.5.2 Loading DASD Modules

To access the root device, the required kernel modules must be loaded. First, the DASD modules must be loaded. They consist of a middle layer module `dasd_mod` and a low-level module depending on the DASD type. Here, ECKD devices are assumed, so the module is called `dasd_eckd_mod`. The `dasd_mod` module requires the channel ranges of the DASDs to access (e.g., `0150`) as an argument. The modules are loaded with the commands:

Example 6.3: *Loading DASD Modules*

```
SuSE Instsys suse:/ # insmod dasd_mod dasd=0150
Using /lib/modules/version/kernel/drivers/s390/block/dasd_mod.o
dasd: initializing...
debug: dasd: new level 3
dasd: Registered successfully to major no 94
dasd: initialization finished
SuSE Instsys suse:/ # insmod dasd_eckd_mod
Using /lib/modules/version/kernel/drivers/s390/block/dasd_eckd_mod.o
dasd(eckd): ECKD discipline initializing
[ ... ]
```

```
Partition check:
 dasda:VOL1/  0X0150: dasda1 dasda2
dasd(eckd): We are interested in: CU 3880/00
dasd(eckd): We are interested in: CU 3990/00
dasd(eckd): We are interested in: CU 2105/00
dasd(eckd): We are interested in: CU 9343/00
```

If the line `Partition check` is printed, all modules have been loaded properly and the DASD device is now available for mounting.

6.5.3 Mounting the Root Device

If the modules have loaded correctly, 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

Example 6.4: *Mounting the Root Device*

```
    SuSE Instsys suse:/ # mount /dev/dasda2 /mnt
```

Note

If the installed system has not been shutdown properly, it may be advisable to check the file system consistency prior to mounting. This prevents any accidental loss of data. In our example, issue the command `fsck /dev/dasda2` to ensure that the file system is in a consistent state.

Note

By just issuing the command `mount`, it is possible to check whether the file system could be mounted correctly.

Example 6.5: *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)
```

6.5.4 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. This is done with the `chroot` command:

Example 6.6: *chroot to the Mounted File System*

```
SuSE Instsys suse:/ # cd /mnt
SuSE Instsys suse:/mnt # chroot /mnt
```

6.5.5 Executing zipl

Now execute the command `zipl` to rewrite the IPL record with the correct values:

Example 6.7: *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
```

6.5.6 Exiting the Rescue System

To exit the rescue system, first leave the shell opened by the `chroot` command:

Example 6.8: *Exit from the chrooted Shell*

```
sh-2.05b# exit
exit
SuSE Instsys suse:/mnt #
```

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 sls; for S/390 and zSeries installation.

***Example 6.9:** 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 system can now be IPLed as described in Chapter 1.7.11 on page 34.

Part II

System

32-bit and 64-bit applications in a 64-bit system environment

SUSE LINUX Enterprise Server is available for several 64-bit platforms. This does not necessarily mean that all the applications contained there have already been ported to 64-bit platforms. SUSE LINUX Enterprise Server supports the use of 32-bit applications in a 64-bit system environment. This section offers a brief overview of how this support is implemented on 64-bit SUSE LINUX Enterprise Server platforms.

7.1 Introduction 194
7.2 Runtime support 194
7.3 Software development 195
7.4 Software Compilation on Biarch Platforms 196
7.5 Kernel specifications 197

7.1 Introduction

SUSE LINUX Enterprise Server for the ipf, ppc64, s390x, sparc64, amd64 and em64t 64-bit platforms is designed so that existing 32-bit applications will run in the 64-bit environment “out-of-the-box”. The corresponding 32-bit platforms are x86 for ipf, ppc for ppc64, s390 for s390x and x86 for amd64 and em64t. This support means that you can continue to use your preferred 32-bit applications without having to wait for a corresponding 64-bit port to become available. The current ppc64 system runs in 32-bit mode, although you can use 64-bit applications.

To understand 32-bit support, we must consider the following issues:

Runtime support How can 32-bit applications be executed?

Development support How should 32-bit applications be compiled to enable them to run both in 32-bit and 64-bit system environments?

Kernel API How can 32-bit applications run under a 64-bit kernel?

7.2 Runtime support

Note

Conflicts between the 32-bit and 64-bit version of an application

If an application is available both for 32-bit and 64-bit environments, the parallel installation of both versions is bound to lead to problems. In such cases, you must decide on one of the two versions and install and use this.

Note

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, sparc64, amd64 and em64t: 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 environment.

All 64-bit libraries and object files are located in directories called `lib64/`, so that the 64-bit object files you would normally expect to find under `/lib/`, `/usr/lib/` and `/usr/X11R6/lib/` are now to be 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 that the file name for both versions can remain unchanged.

No subdirectories of the object directories whose data content does not depend on the word size are moved. For example, the X11 fonts will still be found in the usual location under `/usr/X11R6/lib/X11/fonts`.

This scheme conforms to the LSB (Linux Standards Base) and the FHS (File System Hierarchy Standard).

▶ **ipf**
The 64-bit libraries for both ipf and the 64-bit alpha platform are located in the standard `lib/` directories. In such cases there is neither a `lib64/` directory nor a `lib32/` directory. Instead, ipf 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/`. ◀

7.3 Software development

All 64-bit architectures support the development of 64-bit objects. However, 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 toolchain. The compiling 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 s390x). The flags for the binutils are architecture-dependent, however, GCC will transfer the correct flags to linkers and assemblers. A biarch development toolchain currently exists for sparc64 (supports sparc and sparc64 development), 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 does not support the direct development of 32-bit software on all platforms. If you wish to develop applications for x86 under ipf, you must use the corresponding 32-bit version of SUSE LINUX Enterprise Server.

Please note that all header files must be written in an architecture-independent form and that the installed 32-bit and 64-bit libraries must have an API (Application Programming Interface) that matches the installed header files. The normal SUSE environment is designed according to this principle – in the case of manually updated libraries, you must look after these issues yourself.

7.4 Software Compilation on Biarch Platforms

In order to develop binaries for the other architecture on a biarch architecture, the respective libraries for the second architecture must be installed additionally. These packages are called `rpmname-32bit` if the second architecture is a 32-bit architecture or `rpmname-64bit` if the second architecture is a 64-bit architecture.

Moreover, you 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 64-bit architecture, you need the following RPMs:

libaio 32-bit runtime package

libaio-devel Headers and libraries for the 32-bit development

libaio-64bit 64-bit runtime package

libaio-devel-64bit 64-bit development 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 with a call of the configure script with additional environment variables.

The following example refers to a ppc system with ppc64 as second architecture:

- Set autoconf to use the 64-bit compiler:

  ```
  CC="gcc -m64"
  ```

- Instruct the linker to process 64-bit objects:

  ```
  LD="ld -m elf64ppc"
  ```

- Set the assembler to generate 64-bit objects:

  ```
  AS="gcc -c -m64"
  ```

- Determine that the libraries for libtool and so on come from `/usr/lib64/`:

  ```
  LDFLAGS="-L/usr/lib64"
  ```

- Determine that the libraries are stored in the `lib64` subdirectory:

  ```
  --libdir=/usr/lib64
  ```

- Determine that the 64-bit X libraries are used:

  ```
  --x-libraries=/usr/X11R6/lib64/
  ```

Not all of these variables are needed for every program. Adapt them to the respective program.

An example `configure` call could look as follows:

```
CC="gcc -m64"            \
LDFLAGS="-L/usr/lib64;"  \
        .configure       \
               --prefix=/usr
               --libdir=/usr/lib64
make
make install
```

7.5 Kernel specifications

The 64-bit kernels for ppc64, s390x, amd64 and em64t 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.

Please note that the 32-bit emulation of system calls for a 64-bit kernel does not support a number of APIs used by system programs. This depends on the platform. For this reason, a small number of applications like `lspci` or the LVM administration programs do not have to exist on non-ppc64 platforms as 64-bit programs in order to function correctly.

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.

Note

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 SUSE 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.

Note

8

Booting and Boot Managers

This chapter introduces various methods for booting the installed system. First, some of the technical details of the boot process are explained to help with understanding the various methods. This is followed by a detailed description of the default boot manager GRUB.

8.1 Booting a PC 200
8.2 Boot Concepts 201
8.3 Map Files, GRUB, and LILO 202
8.4 Booting with GRUB 203
8.5 Booting with LILO 211
8.6 Configuring the Boot Loader with YaST 218
8.7 Uninstalling the Linux Boot Loader 221
8.8 Creating Boot CDs 223
8.9 The Boot Loader ZIPL 224

8.1 Booting a PC

After turning on your computer, the first thing that happens is that the BIOS (Basic Input Output System) takes control, initializes the screen and keyboard, and tests the main memory. At this point, no storage media or external devices are known to the system.

After that, the system reads the current date and time as well as information about the most important peripheral devices from the CMOS setup. After reading the CMOS, the BIOS should recognize the first hard disk, including details such as its geometry. It can then start to load the operating system (OS) from there.

To load the OS, the system loads a 512-byte data segment from the first hard disk into main memory and executes the code stored at the beginning of this segment. The instructions contained in it determine the rest of the boot process. This is why the first 512 bytes of the hard disk are often called the *Master Boot Record* (MBR).

Up to this point (loading the MBR), the boot sequence is independent of the installed operating system and is identical on all PCs. Also, all the PC has to access peripheral hardware are those routines (drivers) stored in the BIOS.

8.1.1 Master Boot Record

The layout of the MBR always follows a standard that is independent of the operating system. The first 446 bytes are reserved for program code. The next 64 bytes offer space for a partition table for up to four partitions (see Section 3.9 on page 128). Without the partition table, no file systems exist on the hard disk — the disk would be virtually useless without it. The last two bytes must contain a special *magic number* (`AA55`). An MBR containing a different number would be considered invalid by the BIOS and any PC operating system.

8.1.2 Boot Sectors

Boot sectors are the first sectors on a hard disk partition, except in the case of extended partitions, which are just *containers* for other partitions. Boot sectors offer 512 bytes of space and are designed to contain code capable of launching an operating system on this partition. Boot sectors of formatted

DOS, Windows, and OS/2 partitions do exactly that (in addition, they contain some basic data about the file system structure). In contrast, the boot sector of a Linux partition is empty (even after creating a file system on it). Thus, a Linux partition cannot bootstrap itself, even if it contains a kernel and a valid root file system. A boot sector with a valid start code contains the same magic number as the MBR in its last two bytes (AA55).

8.1.3 Booting DOS or Windows

The DOS MBR of the first hard disk contains information that determines which partition of a hard disk is active (bootable). The active partition is searched for the operating system to boot. Therefore, DOS must be installed on the first hard disk. The DOS program code in the MBR is the first stage of the boot loader. It checks if the specified partition contains a valid boot sector.

If this is the case, the code in this boot sector can be loaded as the second stage of the boot loader, which in turn loads the system programs. Subsequently, the DOS prompt appears or the Windows user interface is started. In DOS, only one primary partition can be marked as active. This is why you cannot install the DOS system on logical drives in an extended partition.

8.2 Boot Concepts

The simplest boot concept involves only one machine with one operating system. The boot process for this case has already been outlined. The same boot concept can be used for a Linux-only machine. Theoretically, you do not need to install a boot loader for such a system. However, in this case you would not be able to pass additional parameters to the kernel at boot time. For a machine with multiple operating systems, the following boot concepts are possible:

Booting other operating systems from a floppy disk:
One operating system is booted from the hard disk. Other operating systems can be booted from the floppy disk drive.

- *Requirements:* bootable floppy disk drive
- *Example:* installation of Linux alongside Windows; booting of Linux from a boot disk

- *Advantage:* no boot loader needs to be installed
- *Disadvantage:* requires working boot disks and the boot process takes longer
- Depending on the purpose of the computer, it is an advantage or disadvantage that Linux cannot be booted without a disk.

Booting another operating system from a USB storage device: The system can also use a USB storage device to drive the boot process. This is very similar to the floppy method, except the necessary data is fetched from the USB memory stick.

Installing a boot manager: This allows you to use several operating systems on a single machine and to choose among the installed systems at boot time. Switching to another operating system requires a reboot. However, the boot manager must be compatible with all the operating systems installed on the machine. The boot managers of SUSE LINUX (LILO and its successor GRUB) can boot all common operating systems. By default, SUSE LINUX installs the preferred boot manager in the MBR, unless this setting is changed during the installation.

8.3 Map Files, GRUB, and LILO

The main obstacle for booting an operating system is that the kernel is usually a file within a file system on a partition on a disk. These concepts are unknown to the BIOS. To circumvent this, maps and map files were introduced. These maps simply note the physical block numbers on the disk that comprise the logical files. When such a map is processed, the BIOS loads all the physical blocks in sequence as noted in the map, building the logical file in memory.

In contrast to LILO, which relies entirely on maps, GRUB tries to gain independence from the fixed maps at an early stage. GRUB achieves this by means of the file system code, which enables access to files by way of the path specification instead of the block numbers.

Note

Boot Loader Selection

If you update from a previous version of SUSE LINUX in which LILO was the boot manager, the new system continues to use LILO. If you install SUSE LINUX from scratch, the system uses GRUB unless the root partition is installed on a RAID system of the following types:

- CPU-controlled RAID controllers, such as many Promise and Highpoint controllers
- software RAID
- LVM

For information about the installation of LILO, search for the keyword "LILO" in the Support Database (`http://portal.suse.de/sdb/en/index.html`).

Note

8.4 Booting with GRUB

GRUB (the Grand Unified Boot loader) consists of two stages. The first stage is only 512 bytes long. It is written to the MBR or to the boot sector of a disk partition or floppy disk. The second, larger stage is loaded after that and holds the program code. The only purpose of the first stage is to load the second one.

The second stage contains code for reading file systems. Currently supported are Ext2, Ext3, ReiserFS, JFS, XFS, Minix, and the DOS FAT file system used by Windows. GRUB has the ability to access file systems even before booting is finished, as long as they are on devices handled by the BIOS (floppies or hard disks).

Note

GRUB and JFS

Though technically possible, a combination of GRUB with JFS is not recommended.

Note

One major advantage of GRUB is that all boot parameters can easily be changed *before* booting. If, for example, the menu file contains an error, it can be fixed. Boot parameters can be entered interactively at a prompt. GRUB offers the possibility to find the location of the kernel and initrd before booting. With this, you can even boot operating systems for which no entry exists in the boot menu.

8.4.1 The GRUB Boot Menu

GRUB displays a graphical splash screen or a text mode interface with a boot menu. The contents of this screen are controlled by the configuration file `/boot/grub/menu.lst`. This file contains all the information about the partitions or operating systems that can be selected from the boot menu.

This menu file is loaded by GRUB directly from the file system on each boot, so there is no need to update GRUB when the file has been modified. To reconfigure the boot loader, simply edit the file via YaST or with your favorite editor. Temporary changes can be made in the interactive edit mode.

The menu file contains commands GRUB should execute. Its syntax is quite simple. Each line consists of a command, optionally followed by arguments that must be separated by spaces, as is the case with shell commands. For historical reasons, there are some commands that allow an = before their first argument. Lines beginning with a hash (#) are comments.

To identify the menu item in the menu overview, specify a name or 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 redirection to boot loaders of other operating systems. The command is `chainloader` and the argument is usually the boot block in another partition, written in GRUB block notation, for example:

```
chainloader (hd0,3)+1
```

The device naming scheme used by GRUB is explained in Section 8.4.1 on the facing page. The above example specifies the first block of the fourth partition on the first hard disk.

The command for specifying a kernel image is `kernel`. The first argument is the path to the kernel image on a partition. The remainder are parameters that are passed to the kernel when booting.

If the kernel does not have the needed built-in drivers for accessing the root partition, `initrd` must be specified. This is a separate GRUB command whose only argument is the path to the initrd file. As the loading address of the initrd is written to the loaded kernel image, the command `initrd` must follow immediately after the `kernel` command.

The `root` command simplifies specification of kernel and initrd files. The only argument for the command `root` is a device or partition (in GRUB notation). This device is used for all kernel, initrd, or other file paths for which no device is specified. This applies up to the next `root` command. The command is not used in the default `menu.lst` file created during the installation. It merely facilitates manual editing.

The `boot` command is implied and thus automatically executed at the end of each menu entry, so it does not need to be written into the menu file. If entering GRUB commands interactively at the prompt, remember to enter the `boot` command at the end. The command itself has no arguments. It merely boots the loaded kernel image or chain loader.

Once you have written all your menu entries, specify which entry to use as the `default`. Otherwise, the first one (number `0`) is booted by default. You can also specify a time-out in seconds after which this should occur. The `timeout` and `default` usually precede the menu entries. A sample configuration file is described in Section 8.4.1 on the next page.

Naming Conventions for Hard Disks and Partitions

GRUB names hard disks and partitions according to conventions that differ from the Linux device names, such as `/dev/hda1`. The first hard disk is always referred to as `/dev/hd0`. The floppy drive is called `/dev/fd0`. The four primary partitions allowed per disk are numbered from `0` to `3`. Logical partitions are counted beginning with `4`.

```
(hd0,0)   first primary partition on 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 ...
```

GRUB does not distinguish between IDE, SCSI, or RAID devices. All hard disks detected by the BIOS or other disk controllers are counted according to the boot sequence set in the BIOS itself.

The fact that BIOS device names do not correspond to Linux devices is an issue resolved with algorithms that establish a mapping. GRUB stores the result in a file (`device.map`), which can be edited. For more information about `device.map`, refer to Section 8.4.2 on page 208.

For GRUB, a file name must be specified as a device name written in parentheses followed by the full path to the file and the file name. The path must always start with a slash. For example, on a system with a single IDE disk and Linux on the first partition, the bootable kernel might be specified with:

```
(hd0,0)/boot/vmlinuz
```

A Sample Menu File

The following example shows how the GRUB menu file works. This imaginary machine has a Linux boot partition on `/dev/hda5`, a root partition on `/dev/hda7`, and a Windows installation on `/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 part defines the splash screen configuration:

gfxmenu (hd0,4)/message The background image is located on `/dev/hda5` and has the name `message`.

color The color scheme: white as normal foreground, blue as normal background, black for the foreground of selected items, and light gray as the selection background. These colors do not affect the graphical splash screen as defined under `gfxmenu`, but the standard GRUB interface. On a SUSE LINUX system, this interface can be accessed from the splash screen by pressing Esc.

default 0 By default, the first menu entry `title linux` should be booted.

timeout 8 After 8 seconds without user input, GRUB will automatically boot the default entry.

The second, larger part defines the different operating systems to boot:

- The first entry (`title linux`) is responsible for booting SUSE LINUX. The kernel (`vmlinuz`) is located on the first hard disk on the first logical partition (which is the boot partition in this case). The appended arguments are kernel parameters, such as the root partition and the video mode. The root partition is specified according to the Linux convention (`/dev/hda7`), as this information is interpreted by the Linux kernel, not by GRUB. The `initrd` image is located on the same logical partition of the first hard disk.
- The second entry is responsible for booting Windows, which is installed on 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 the floppy drive without changing any BIOS settings.
- The `failsafe` entry boots a Linux kernel with a number of kernel parameters that enable booting Linux even if the hardware is causing problems.

Changing the Hard Disk Sequence

Some operating systems, such as Windows, can only start from the first hard disk. If you have such an operating system installed on a different hard disk, you can implement a logical change for the respective menu entry. However, this only works if the operating system accesses the hard disks by way of the BIOS when booting.

```
...
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 sequence of the hard disks is changed with `map`. This change does *not* affect the logic within the GRUB menu file. You still need to specify the second hard disk for `chainloader`.

Editing Menu Entries During the Boot Procedure

From the graphical boot menu of GRUB, use the cursor keys to select the operating system to boot. If you select a Linux system, you can add boot parameters. After pressing Esc and exiting the splash screen, n press E to edit individual menu entries directly. Changes made in this way only apply to the current boot procedure and will not be adopted permanently.

Note

Keyboard Layout during the Boot Procedure

The US keyboard layout is the only one available at boot time.

Note

After enabling the editing mode, use the arrow keys to navigate to the entry to change. To make the selected item editable, press E again. Adjust the entry as desired. Leave the editing mode with Enter and go back to the menu, where the changed entry can be booted by pressing E. In the lower part of the screen, GRUB displays further options.

8.4.2 The File device.map

The file `device.map` maps GRUB device names to Linux device names. This is only relevant when running the GRUB shell as a Linux program (command `grub`). For this purpose, the program reads the file `device.map`. See Section 8.4.4 on the next page for more information.

GRUB does not have access to the boot sequence information in the BIOS. If your system contains both IDE and SCSI hard disks, GRUB must try to determine the boot sequence by means of a special procedure. It saves the results of this check to the file `/boot/grub/device.map`. For a system that boots IDE devices before SCSI devices, the file `device.map` could look as follows:

```
(fd0) /dev/fd0
(hd0) /dev/hda
(hd1) /dev/hdb
(hd2) /dev/sda
(hd3) /dev/sdb
```

As 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 shell to modify it if necessary (see Section 8.4.4). Once you have successfully booted your Linux system, edit the file `device.map` permanently with the YaST boot loader module or an editor of your choice.

Any manual change to the `device.map` file requires that you update your GRUB installation. Use the following command:

```
grub --batch --device-map=/boot/grub/device.map \
< /etc/grub.conf
```

8.4.3 The File /etc/grub.conf

GRUB stores another important part of its configuration in the file `grub.conf`. This file defines the parameters and options needed by the `grub` command to install the boot loader correctly:

```
root (hd0,4)
install /grub/stage1 d (hd0) /grub/stage2 0x8000 (hd0,4)/grub/menu.lst
quit
```

The individual entries have the following meaning:

root (hd0,4) This command tells GRUB that all subsequent commands should be applied to the first logical partition on the first hard disk, where the boot files are located.

install `parameter` The command `grub` should be run with the parameter `install`. `stage1` of the boot loader should be installed in the MBR of the first hard disk (`/grub/stage1 d (hd0)`). `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.

8.4.4 The GRUB Shell

GRUB actually consists of two parts: the boot loader and a normal Linux program (`/usr/sbin/grub`). This program is referred to as the *GRUB shell*. The functionality to install the boot loader on a hard disk or floppy disk is integrated into the GRUB shell through the internal commands `install` and `setup` — these commands can be executed using the GRUB shell on a running Linux system. However, these commands are also available while the system is booting with GRUB — before Linux is even running. This makes the repair of a defective system much easier.

8.4.5 Setting a Boot Password

Because GRUB is able to access file systems upon booting, it could also be used to read files that would not be accessible under normal circumstances — on a running system, users would need `root` permissions to read them. To put a stop to this, set a boot password. Such a password can be used to prevent unauthorized access to file systems at boot time and to prevent users from booting certain installed systems.

To create a boot password, log in as `root` and proceed as follows:

1. At the root prompt, enter `grub`.
2. In the GRUB shell, encrypt the password:

   ```
   grub> md5crypt
   Password: ****
   Encrypted: $1$lS2dv/$JOYcdxIn7CJk9xShzzJVw/
   ```

3. 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/
   ```

   From now on, executing GRUB commands from the boot prompt is impossible without knowing the password. Permission to do so is only granted after pressing Ⓟ and entering the password. However, users can still boot all operating systems without any restriction.

4. To keep users from booting certain operating systems, add the entry `lock` for every section in `menu.lst` to prevent from being booted without entering a password. Example:

   ```
   title linux
     kernel (hd0,4)/vmlinuz root=/dev/hda7 vga=791
     initrd (hd0,4)/initrd
     lock
   ```

   After rebooting, trying to boot this entry from the menu would result in the following error message:

   ```
   Error 32: Must be authenticated
   ```

Return to the menu by pressing Enter. From the menu, pressing P prompts for the password. The selected system (Linux in this case) should boot after typing the password and pressing Enter.

Note

Boot Password and Splash Screen

Setting a boot password for GRUB disables the graphical splash screen as displayed by default.

Note

8.4.6 Boot Problems with GRUB

The geometry of attached hard disks is checked by GRUB only upon booting. In some cases, the BIOS returns inconsistent values, and GRUB reports *GRUB Geom Error* (see `http://portal.suse.com/sdb/en/2003/03/fhassel_geom-error.html`). In this case, use LILO or update the BIOS. Details on the installation, configuration, and maintenance of LILO is available in the Support Database article: `http://portal.suse.de/sdb/en/2004/01/lilo_overview.html`>.

8.4.7 For More Information

Extensive information about GRUB is available at `http://www.gnu.org/software/grub/`. If you have texinfo installed on your machine, view the GRUB info pages in a shell by entering `info grub`. You can also search for the keyword "GRUB" in the Support Database at `http://portal.suse.de/sdb/en/index.html` to get information about special issues.

8.5 Booting with LILO

The Linux boot loader LILO is suitable for installation in the MBR. LILO has access to two real-mode hard disks and is able to find all the data it needs from the *raw* hard drives without any partitioning data. Therefore, operating systems can also be booted from the second hard disk. Unlike with the DOS boot process, the entries in the partition table are ignored when using LILO.

The main difference from the standard DOS boot process is the possibility to load diverse installed operating systems when booting. After loading the `MBR` into memory, LILO is started, allowing the user to select from the list of preinstalled systems. At system start-up, it can load boot sectors from partitions to boot an operating system from the respective partition or load the Linux kernel and boot Linux. It also provides the important possibility of passing a command to the kernel. For security reasons, some or all LILO services can be protected with a password.

8.5.1 Basics

The LILO boot mechanism consists of the following components:

- The *LILO boot sector* with the initial part (first stage) of the LILO code that activates the actual LILO when the system is booted.
- The LILO machine code, located in `/boot/boot-menu.b`.
- A *map* file (`/boot/map`), where LILO enters the location of Linux kernels and other data during its installation.
- Optional: the *message file* `/boot/message`, which displays the graphical boot menu from which the operating system can be selected
- The different Linux kernels and boot sectors LILO should offer

Caution

Map file deletion through write access

Any write access (even through file movements) to any of these files corrupts the map file – unless LILO is *updated* (see 8.5.4 on page 217). This is especially important when changing kernels.

Caution

The following locations are suitable for storing the LILO *boot sector*:

On a *floppy disk* This is the simplest, but also the slowest method for booting with LILO. Choose this alternative if you do not want to change the existing boot sector.

In the *boot sector* of a primary Linux partition on the first hard disk This leaves the MBR untouched. Before it can be booted, the partition must be marked active. Start fdisk as `root` with the command `fdisk -s ⟨partition⟩`. The program will ask for a command. You can obtain a list of the available commands by entering `m`. The `a` command can be used to mark a partition as active.

In the *Master Boot Record* This variation offers the highest flexibility. It is the only possible alternative if all the Linux partitions reside on the second hard disk, and if there is no extended partition on the first drive. Every setting of the MBR must be edited with extreme care because errors may have severe consequences.

In a boot sector booted by *another boot manager* Try this if you are using another boot manager and want to continue using it. Depending on its flexibility and power, there are several variations. A common case: you have a primary Linux partition on the second hard disk from which you want to boot Linux. If your boot manager is able to boot this partition through its boot sector, you may install LILO into this boot sector and then tell your boot manager that the partition is active.

8.5.2 Configuring LILO

LILO is a flexible boot manager that offers many ways of adapting a configuration to your needs. The most important options and meanings are described below. For more detail, look at [4].

The configuration of LILO is stored in the file `/etc/lilo.conf`. You should always make a backup of the last working `lilo.conf` file before changing it. On the other hand, any changes in this file take effect only when reinstalling LILO i. e. after running the `lilo` command against the changed `/etc/lilo.conf` file. For details on this, refer to Section 8.5.4 on page 217.

8.5.3 Structure of lilo.conf

`/etc/lilo.conf` starts with a global section, followed by one or more system sections for each operating system LILO should start. Each system section starts with a line beginning with either `image` or `other`.

The order of entries in `/etc/lilo.conf` does matter, in the sense that the first one in the list is booted automatically if there is no user input at the boot screen (and unless the `default` option is used). This may happen after a certain interval which can be set with the `delay` and `timeout` options as explained below.

A sample configuration for a computer with both Windows and Linux is shown in File 8.1. The bootable systems include a newly installed Linux kernel (`/boot/vmlinuz`) and the original kernel which is used as a fallback (`/boot/vmlinuz.shipped`). There is also an entry to boot Windows on `/dev/hda1`, and an additional one to start the program MemTest86.

Example 8.1: *Sample Configuration of /etc/lilo.conf*

```
### LILO global section
boot    = /dev/hda           # LILO installation target: MBR
backup  = /boot/MBR.hda.990428  # backup file for the old MBR
                                # 1999-04-28
vga     = normal             # normal text mode (80x25 chars)
read-only
menu-scheme = Wg:kw:Wg:Wg
lba32                        # Use BIOS to ignore
                             # 1024 cylinder limit
prompt
password = q99iwr4           # LILO password (example)
timeout = 80                 # Wait at prompt for 8 s before
                             # default is booted
message = /boot/message      # LILO's greeting

### LILO Linux section (default)
  image  = /boot/vmlinuz     # Default
  label  = linux
  root   = /dev/hda7         # Root partition for the kernel
  initrd = /boot/initrd

### LILO Linux section (fallback)
  image  = /boot/vmlinuz.shipped
  label  = Failsafe
  root   = /dev/hda7
  initrd = /boot/initrd.suse
  optional

### LILO other system section (Windows)
  other  = /dev/hda1         # Windows partition
  label  = windows

### LILO memory test section (memtest)
```

```
  image  = /boot/memtest.bin
  label  = memtest86
```

Anything between a # and the end of a line is regarded as a comment. Spaces and comments are ignored by LILO and can be used to improve readability. The entries in the above sample file include mandatory options which are explained in the list below, and others which are described in Section 8.5.3 on page 213.

- **Global section** (Parameter part)
  - ▷ `boot=bootdevice`
    The device on whose first sector LILO should be installed. `bootdevice` may be a floppy disk drive (`/dev/fd0`), a partition (e.g., `/dev/hdb3`), or an entire disk (e.g., `/dev/hda`). In the last case, LILO would be installed in the MBR. If this option is missing, LILO is installed on the current root partition by default.
  - ▷ `lba32`
    With this option, ignore the 1024-cylinder limit of LILO if your BIOS supports this.
  - ▷ `prompt`
    Forces the LILO prompt to be displayed. The default is not to display any prompt (see Section 8.5.3 on page 213, option `delay`).

    This is recommended if LILO needs to manage more than one system. It should be used together with the `timeout` option, to guarantee that the default system is automatically booted if nothing is entered at the prompt.
  - ▷ `timeout=deciseconds` Sets a time-out for selecting an operating system to boot. The default system is booted after the time-out if there is no user input. The `deciseconds` value specifies the time-out in tenths of a second. Pressing Shift or the arrow keys disables the time-out, causing LILO to wait for further user input. The default time-out is set to `80` (8 seconds).
- **Linux section**
  - ▷ `image=kernelimage`
    This specifies the name of the kernel image to boot, including its directory location. With a new system, this is most likely `/boot/vmlinuz`.

  - ▷ `label=name`

    A name for the system in question (e.g., `Linux`) which may be freely chosen but must be unique as far as the contents of `/etc/lilo.conf` are concerned. Its maximum length is 15 characters, and it may only consist of letters, numbers, and underscores, i. e. no blanks or special characters. For more about the specific characters that are allowed, see [4], Section 3.2.1. The default for this option is the file name of the corresponding kernel image (e.g., `/boot/vmlinuz`).

    The same name is presented in the boot menu as one of the selectable items. If there are several systems installed, you may want to provide a more detailed description of the bootable systems by creating a message file, (see Section 8.5.3 on page 213, option `message`).

  - ▷ `root=rootdevice`

    This is used by LILO to tell the kernel about the name of the root partition (e.g., `/dev/hda2`) of your Linux system. You should use this option to be on the safe side: if it is omitted, the kernel just assumes that the root partition is identical with its own root device (as derived from `kernelimage`).

  - ▷ `append=parameter`

    In order to pass on additional boot parameters to the kernel, you can add the `append` option to an existing `lilo.conf` file, followed by a = and your parameters. Individual parameters must be seperated by spaces, and the parameter string as a whole must be enclosed in quotation marks. After saving the file, you still need to execute the `lilo` command as `root`, so that LILO reinstalls the boot loader and takes the changes into account during the next boot.

- **Linux part** (Linux — Safe Settings)

  Even if you installed a customized kernel, you are still able to boot the SuSE standard kernel.

  - ▷ `optional`

    If you decide to delete `/boot/vmlinuz.shipped` (*not recommended*), this section will be skipped without an error message during LILO installation.

- **Other systems**

- ▷ `other=partition`

  `other` tells LILO to start the partitions of other systems (e.g., `/dev/hda1`).
- ▷ `label=name`

  Select a name for the system. This is recommended, because the default — the raw device name — is not very informative.

- **Memory Test**

  Entry for the memory test program memtest86.

This section merely covers the basic entries required in `/etc/lilo.conf`. Other useful settings can be found in the man page `man lilo.conf`.

8.5.4 Installing and Uninstalling LILO

Caution

Booting of other operating systems

Before you install LILO, make sure that any other existing operating systems can be booted from floppy disk (not possible for Windows XP/2000/NT). In particular, make sure that fdisk is available. As far as SUSE LINUX is concerned, you can use the installation CD or DVD as a fallback boot medium.

Caution

Updating After Changing the Configuration

If any of the LILO components have changed, or if you have modified your configuration in `/etc/lilo.conf`, you must update the LILO boot loader. This is easily done by launching the map installer as `root` with the command `/sbin/lilo`

LILO creates a backup of the target boot sector, writes its first stage into the boot sector, then generates a new map file (also see 8.5 on page 211). LILO issues a report on each installed system In the case of the sample configuration described above, it should look like this:

Example 8.2: *Output After Launching LILO*

```
Added linux * Added suse Added windows Added memtest86
```

When the boot loader update is completed, you can reboot the machine by issuing this command as `root`: `shutdown -r now`

While rebooting, the BIOS first performs its system test. Immediately afterwards, you should see LILO and its command prompt, where you can enter parameters and select a boot image. You can also hit Tab to see a list of the systems installed.

8.6 Configuring the Boot Loader with YaST

This YaST module simplifies the configuration of the boot loader. However, you should not experiment with this module unless you understand the concepts behind it. Read the corresponding parts of the Installation and Administration before changing the boot loader configuration. The following discussion mainly covers the default boot loader GRUB.

Note

Do not change the boot method of a running system unless you really know what you are doing.

Note

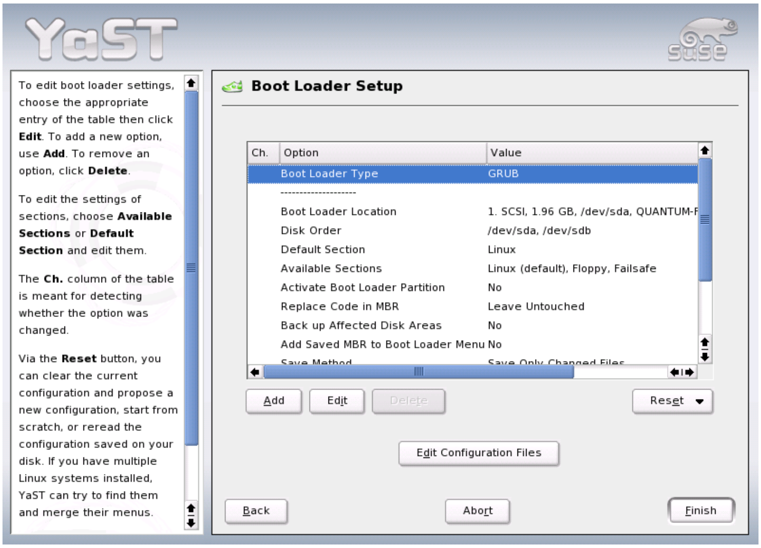
In the YaST Control Center, select 'System' → 'Boot Loader Configuration'. The current boot loader configuration of your system will be displayed, enabling you to make any needed changes (see Figure 8.1 on the facing page).

8.6.1 The Main Window

The table listing the configuration data consists of three columns. Under 'Changed' (to the left), flags mark the changed options listed in the center column. To add an option, click 'Add'. To change the value of an existing option, select it with a mouse click and click 'Edit'. If you do not want to use an existing option at all, select it and click 'Delete'.

'Reset' to the right under the configuration window offers the following options:

Propose New Configuration Generates a new configuration suggestion. Older Linux versions or other operating systems found on other partitions will be included in the boot menu, enabling you to boot Linux or its old boot loader. The latter takes you to a second boot menu.



To edit boot loader settings, choose the appropriate entry of the table then click **Edit**. To add a new option, use **Add**. To remove an option, click **Delete**.

To edit the settings of sections, choose **Available Sections** or **Default Section** and edit them.

The **Ch.** column of the table is meant for detecting whether the option was changed.

Via the **Reset** button, you can clear the current configuration and propose a new configuration, start from scratch, or reread the configuration saved on your disk. If you have multiple Linux systems installed, YaST can try to find them and merge their menus.

Boot Loader Setup

Ch.	Option	Value
	Boot Loader Type	GRUB

	Boot Loader Location	1. SCSI, 1.96 GB, /dev/sda, QUANTUM-F
	Disk Order	/dev/sda, /dev/sdb
	Default Section	Linux
	Available Sections	Linux (default), Floppy, Failsafe
	Activate Boot Loader Partition	No
	Replace Code in MBR	Leave Untouched
	Back up Affected Disk Areas	No
	Add Saved MBR to Boot Loader Menu	No
	Save Method	Save Only Changed Files

Figure 8.1: Configuring the Boot Loader with YaST

Start from Scratch Enables you to create the entire configuration from scratch. No suggestions will be generated.

Reread Configuration from Disk If you already performed some changes and are not satisfied with the result, you can reload your current configuration with this option.

Propose and Merge with Existing GRUB Menus If another operating system and an older Linux version are installed in other partitions, the menu will be generated from an entry for the new SUSE LINUX, an entry for the other system, and all entries of the old boot loader menu. This procedure might take some time. This is not possible if LILO is used.

Restore MBR from Hard Disk The MBR saved on the hard disk will be written back.

Use 'Edit Configuration Files' under to edit the relevant configuration files in an editor. To edit a file, load it by means of the selection field. Click 'OK' to save your changes. To exit the boot loader configuration, click 'Cancel'. Click 'Back' to return to the main window.

Caution

Remember that the sequence of the options or commands is very important in GRUB. If the specified sequence is not followed, the machine may not boot.

Caution

8.6.2 Boot Loader Configuration Options

For less experienced users, the configuration with YaST is much easier than editing the files directly. Select an option and click 'Edit' to open a dialog in which to change the settings according to your needs. Click 'OK' to confirm the changes and return to the main menu, where you can edit other options. The available options depend on the boot loader used. The following list introduces some options of the boot loader GRUB:

Boot Loader Type Use this option to switch between GRUB and LILO. You will be taken to another dialog in which to specify the way in which this change should be performed. For instance, convert the current GRUB configuration into a similar LILO configuration. However, some settings may be lost if no equivalent options are available. You can also create a new configuration from scratch or generate and edit a suggestion for a configuration.

If you start the boot loader configuration in the running system, you can load the configuration from the hard disk. If you decide to return to the original boot loader, you can load its configuration by means of the last option. However, this possibility only exists as long as you do not close the boot loader module.

Boot Loader Location Use this dialog to define where to install the boot loader: in the master boot record (MBR), in the boot sector of the boot partition (if available), in the the boot sector of the root partition, or on a floppy disk. Use 'Others' to specify a different location. Refer to the information about GRUB in the Installation and Administration.

Disk Order If your computer has more than one hard disk, specify the boot sequence of the disks as defined in the BIOS setup of the machine.

Default Section With this option, set which kernel or operating system to boot by default, if no other entry is selected in the boot menu. The default system is booted after the time-out. Click this option and 'Edit' to see a list of all boot menu entries. Then select one entry from the list and click 'Set as Default'. Click 'Edit' to modify an entry. The dialog that opens allows changing the order of entries in the boot menu, adding, modifying, and deleting entries, and setting a default entry.

Available Sections The existing entries of the boot menu are listed under this option in the main window. If you select this option then click 'Edit', a dialog opens that is identical to the 'Default Entry' dialog.

Make Boot Loader Partition Active
Use this option to activate the partition whose boot sector holds the boot loader, independently from the partition on which the directory with the helper files of the boot loader are stored (`/boot` or the root directory `/`).

Replace Code in MBR Specify whether to overwrite the MBR, which may be necessary if you have changed the location of the boot loader.

Backing up Files and Parts of Hard Disks
Backs up the changed hard disk areas.

Add Saved MBR to Boot Loader Menu
Adds the saved MBR to the boot loader menu.

You can use 'Time-out' to define how many seconds the boot loader should wait for keyboard input before the default system is booted. A number of other options can be specified with 'Add'. However, the use of these options requires a deeper understanding and is not covered here. Refer to the relevant chapter in the Installation and Administration and the manual pages of GRUB and LILO (`man grub`, `man lilo`, and `man lilo.conf`). Additionally, a detailed online manual for GRUB is available at `http://www.gnu.org/software/grub/`.

8.7 Uninstalling the Linux Boot Loader

There are two ways to uninstall the Linux boot loader:

- Restore the backup of the original MBR by means of the YaST boot loader module. YaST creates this backup automatically. Refer to the installation section of the *User Guide* for details of the YaST boot loader module.
- Install a different boot loader or restore the DOS or Windows MBR.

Caution

Invalid backups of boot sectors

A boot sector backup is no longer valid if the partition in question has a new file system. The partition table of an MBR backup becomes invalid if the hard disk has been repartitioned since the backup was created. Obsolete *backups* are time bombs. It is best to delete them from `/boot/backup.mbr` promptly.

Caution

8.7.1 Restoring the MBR (DOS, Win9x, or ME)

It is very simple to restore a DOS or Windows MBR. Just enter the MS-DOS command (available since DOS version 5.0) `fdisk /MBR`. These commands only write the first 446 bytes (the boot code) into the MBR and leave the partition table untouched, unless the MBR as a whole (see Section 8.1.1 on page 200) is treated as invalid due to an incorrect magic number. In this case, the partition table is set to zero. After restoring the MBR, mark the desired start partition as bootable (using fdisk again). This is required for the MBR routines of DOS and Windows.

8.7.2 Restoring the MBR of Windows XP

Boot from the Windows XP CD and press Ⓡ during the setup to start the recovery console. Select your Windows XP installation from the list and enter the administrator password. At the input prompt, enter the command `FIXMBR` and confirm with `y` when asked to do so. Then reboot the computer with `exit`.

8.7.3 Restoring the MBR of Windows 2000

Boot from the Windows 2000 CD and press Ⓡ then Ⓒ in the next menu to start the recovery console. Select your Windows 2000 installation from the list and enter the administrator password. At the input prompt, enter the command `FIXMBR` and confirm with `y` when asked to do so. Then reboot the computer with `exit`.

8.8 Creating Boot CDs

This concerns problems arising when attempting to boot a system with the LILO boot manager configured with YaST. The creation of a system boot disk fails with more recent SUSE LINUX versions because the space available on a floppy disk is no longer sufficient for the start-up files.

8.8.1 Procedure

It is possible to create a bootable CD-ROM containing the Linux start-up files if your system has an installed CD writer. This solution is only a workaround. It should normally be possible to configure LILO properly. Refer to the documentation about this subject in `/usr/share/doc/packages/lilo/README`, or read the man pages `man lilo.conf` and `man lilo`.

8.8.2 Boot CD with ISOLINUX

It is easiest to create a bootable CD with the ISOLINUX boot manager. The SuSE installation CDs are also made bootable with isolinux.

- Boot the installed system first using the following alternate procedure:
  - ▷ Boot from the installation CD or DVD as for installation.
  - ▷ Choose the preselected option 'Installation' during the boot sequence.
  - ▷ Choose the language and keyboard map next.
  - ▷ In the following menu, choose 'Boot installed system'.
  - ▷ The root partition is automatically detected and the system is booted from it.
- Install `syslinux` with YaST.
- Open a root shell. The following commands create a temporary directory and copy the files required for the booting of the Linux system (the isolinux boot loader as well as the kernel and the initrd) into it:

```
mkdir /tmp/CDroot
cp /usr/share/syslinux/isolinux.bin /tmp/CDroot/
cp /boot/vmlinuz /tmp/CDroot/linux
cp /boot/initrd /tmp/CDroot
```

- Create the boot loader configuration file `/tmp/CDroot/isolinux.cfg` with your preferred editor. Enter the following content:

  ```
  DEFAULT linux
  LABEL linux
    KERNEL linux
    APPEND initrd=initrd root=/dev/hdXY [boot parameter]
  ```

  Enter your root partition for the parameter `root=/dev/hdXY`. It is listed in the file `/etc/fstab`. Enter additional options for the setting `[boot parameter]`, which should be used during booting. The configuration files could, for example, look like this:

  ```
  DEFAULT linux
  LABEL linux
    KERNEL linux
    APPEND initrd=initrd root=/dev/hda7 hdd=ide-scsi
  ```

- The following command (entered at a command prompt) then creates an ISO-9660 file system for the CD.

  ```
  mkisofs -o /tmp/bootcd.iso -b isolinux.bin -c boot.cat
    -no-emul-boot -boot-load-size 4
    -boot-info-table /tmp/CDroot
  ```

  The complete command must be entered as one line.

- The file `/tmp/bootcd.iso` can be written to CD after that with either graphical CD writing applications, like K3b simply at a command prompt: `cdrecord -v speed=2 dev=0,0,0 /tmp/bootcd.iso -eject`

  The parameter `dev=0,0,0` must be changed according to the SCSI ID of the writer. This can be determined with the command `cdrecord -scanbus`. Also, refer to the man page `cdrecord`.

- Test the boot CD. Reboot the computer to verify whether the Linux system is started correctly from the CD.

8.9 The Boot Loader ZIPL

Once the system is installed on your DASD, you need to write initial information for the IPL from disk, such as the location of the kernel image and a parameter line. This is done by means of the tool ZIPL, which retrieves this information either from the command line or a configuration file.

8.9.1 For use with Kernel Version 2.4.x

The syntax of ZIPL is as follows:

```
zipl [options] [configuration]

 Options:
 -h or --help     prints this information
 -c <CONFIG-FILE> or --config=<CONFIG-FILE>
      <CONFIG-FILE> specifies the config file to be used.
      This option overrides the environment variable
      ZIPLCONF.
```

The following options override settings in the config file Note: ⟨*ARG*⟩ means required argument, `[ARG]` means optional argument.

`-t <DIRECTORY>` or `--target=<DIRECTORY>`
⟨*DIRECTORY*⟩ specifies the target directory where zipl installs some files needed for the ipl process

`-i <IMAGE[,ADDRESS]>` or `--image=<IMAGE[,ADDRESS]>`
⟨*IMAGE*⟩ specifies the file name of the bootable image `[ADDRESS]` specifies the address where the image will be loaded in the memory

`<RAMDISK[,ADDRESS]>` or `--ramdisk=<RAMDISK[,ADDRESS]>`
⟨*RAMDISK*⟩ specifies the file name of the ramdisk to be loaded `[ADDRESS]` specifies the address where the ramdisk will be loaded in the memory

`<PARMFILE[,ADDRESS]>` or `--parmfile=<PARMFILE[,ADDRESS]>`
⟨*PARMFILE*⟩ specifies the file name of the parm file to be loaded `[ADDRESS]` specifies the address where the parm file will be loaded in the memory

`-d <PARTITION>` or `--dumpto <PARTITION>`
⟨*PARTITION*⟩ specifies the device node of the partition on with the dump will be created. Example:`/dev/dasdb1` or `/devfs/dasd/0192/part1`

The command ZIPL reads the configuration file in `/etc/zipl.conf` and uses the parameters listed in the file.

8.9.2 The ZIPL Configuration File

The configuration file for the ZIPL boot loader resides in the directory `/etc/zipl.conf`.

Example 8.3 shows a `zipl.conf` file. It is divided into several sections. You can define more than one way for IPLing your Linux System.

Example 8.3: `/etc/zipl.conf`

```
[defaultboot]
default=ipl

[ipl]
target=/boot/zipl
image=/boot/zilo-kernel/image
#ramdisk=/boot/initrd
parameters="dasd=0150 root=/dev/dasda2 noinitrd"

[dumptape]
target=/boot
dumpto=/boot/zipl
```

The section `[defaultboot]` defines the section that will be called if you call ZIPL without any parameters.

The parameter line `parameters=...` defines the commands that are given to the kernel during start-up. Here you can specify which DASDs should be used and which one the root file system is on.

To add specific DASDs in the parameter line:

```
parameters="dasd=0150,0151,0152 root=/dev/dasda2 noinitrd"
```

If you want to add a DASD range:

```
parameters="dasd=0150-0155 root=/dev/dasda2 noinitrd"
```

Note

DASDs and the Command Line

You can add or delete DASDs or DASD ranges from the parameter line. However, do not remove the DASD containing the root file system. Otherwise, the system will not be able to boot.

Note

The Linux Kernel

The kernel manages the hardware of every Linux system and makes it available to the various processes. Although the information provided in this chapter will not make you a kernel hacker, you will learn how to perform a kernel update and how to compile and install a custom kernel. If you follow the instructions in this chapter, the previous kernel will remain functional and can be booted if necessary.

9.1 Kernel Update 228
9.2 Kernel Sources 229
9.3 Kernel Configuration 229
9.4 Kernel Modules 230
9.5 Settings in the Kernel Configuration 232
9.6 Compiling the Kernel 233
9.7 Installing the Kernel 234
9.8 Cleaning Your Hard Disk after Compilation 235

The kernel that is installed in the `/boot/` directory is configured for a wide range of hardware. Normally, there is no need to compile a custom kernel, unless you want to test experimental features and drivers.

Several `Makefiles` are provided with the kernel to automate the process. Select the hardware settings and other kernel features. As you need to know your computer system pretty well to make the right selections, modifying an existing and working configuration file is recommended for your first attempt.

9.1 Kernel Update

To install an official SUSE update kernel, download the update RPM from the SUSE FTP server or a mirror like `ftp://ftp.gwdg.de/pub/linux/suse/`. To determine the version of your current kernel, look at the version string with `cat /proc/version`. Alternatively, check to which package the kernel (`/boot/vmlinuz`) belongs with `rpm -qf /boot/vmlinuz`.

Before installing this package, make a backup copy of the original kernel and the associated `initrd`. As `root`, enter the following two commands:

```
cp /boot/vmlinuz /boot/vmlinuz.old
cp /boot/initrd /boot/initrd.old
```

Then install the new kernel with the command `rpm -Uvh <packagename>`. Replace `packagename` with the name of the kernel RPM to install.

Since SUSE LINUX 7.3, reiserfs is the standard file system. It requires the use of an “initial ramdisk”. Therefore, use the command `mk_initrd` to write the new initial ramdisk. In current SUSE LINUX versions this is done automatically when installing the new kernel.

To be able to boot the old kernel, configure the boot loader accordingly (for more information, refer to Chapter 8 on page 199). Finally, reboot to load the new kernel.

To reinstall the original kernel from the SUSE LINUX CDs, the procedure is almost the same, except you copy the kernel RPM from the directory `boot/` on CD 1 or the DVD. Now, install as described above. If you receive an error message saying that a newer kernel rpm is already installed, add the option `--force` to the above rpm command.

9.2 Kernel Sources

To build a kernel, the package `kernel-source` must be installed. Additional packages, like the C compiler (package `gcc`), the GNU binutils (package `binutils`), and the include files for the C compiler (package `glibc-devel`), are selected for installation automatically by YaST.

After installation, the kernel sources are located in `/usr/src/linux-<kernel-version>/`. If you plan to experiment with different kernels, unpack them in different subdirectories and create a symbolic link to the current kernel source. As there are software packages that rely on the sources being in `/usr/src/linux/`, maintain this directory as a symbolic link to your current kernel source. YaST does this automatically.

9.3 Kernel Configuration

The configuration of the current kernel is stored in the file `/proc/config.gz`. To modify this configuration, go to the directory `/usr/src/linux/` as `root` and execute the following commands:

```
zcat /proc/config.gz > .config
make oldconfig
```

The command `make oldconfig` uses the file `/usr/src/linux/.config` as a template for the current kernel configuration. Any new options for your current kernel sources will be queried. If the file `.config` does not exist, the default configuration included in the kernel sources will be used.

9.3.1 Configuration on the Command Line

To configure the kernel, change to `/usr/src/linux` and enter the command `make config`. Choose the features you want supported by the kernel. Usually, There are two or three options: y, n, or m. m means that this device will not be compiled directly into the kernel, but loaded as a module. Drivers needed for booting the system must be integrated into the kernel with y. Press Enter to confirm the default settings read from the file `.config`. Press any other key to view a brief help text about the respective option.

9.3.2 Configuration in Text Mode

"menuconfig" is a more comfortable way to configure the kernel. If necessary, install `ncurses-devel` with YaST. Start the kernel configuration with the command `make menuconfig`.

For minor changes in the configuration, you do not have to go through all the questions. Instead, use the menu to access certain sections directly. The default settings are loaded from the file `.config`. To load a different configuration, select 'Load an Alternate Configuration File' and enter the file name.

9.3.3 Configuration in the X Window System

If you installed and configured the X Window System (package `xf86`) and Tcl/Tk (`tcl` and `tk`), you can use the command `make xconfig` to access a graphical user interface for the configuration. If you are not logged in to the X Window System as `root`, enter the command `xhost +` to give `root` access to the display. The default settings will be loaded from the file `.config`. As the configuration with `make xconfig` is not as well maintained as the other configuration possibilities, run the command `make oldconfig` after using this configuration method.

9.4 Kernel Modules

There is a wide variety of PC hardware components. To use this hardware properly, you need a "driver" with which the operating system (in Linux, the "kernel"), can access this hardware. There are basically two ways of integrating drivers into your system:

- The drivers can be compiled directly into the kernel. Such a kernel ("in one piece") is referred to as a *monolithic* kernel. Some drivers are only available in this form.
- Drivers can be loaded into the kernel on demand. In this case, the kernel is referred to as a *modularized* kernel. This has the advantage that only those drivers really needed are loaded and the kernel thus contains nothing unnecessary.

Which drivers to compile into the kernel and which to load as run-time modules is defined in the kernel configuration. Basically, components not required for booting the system should be built as modules. This makes sure the kernel does not become too big to be loaded by the BIOS or a boot loader. Drivers for `ext2`, the SCSI drivers on a SCSI-based system, and similar drivers should be compiled into the kernel. In contrast, items, such as `isofs`, `msdos`, or `sound`, which are not needed for starting your computer system, should definitely be built as modules.

Kernel modules are located in `/lib/modules/<version>/`. `Version` stands for the current kernel version.

9.4.1 Hardware Detection with the Help of hwinfo

hwinfo can detect the hardware of your system and select the drivers needed to run this hardware. Get a small introduction to this command with `hwinfo --help`. If you, for example, need information about your SCSI devices, use the command `hwinfo --scsi`.

All this information is also available in YaST in the hardware information module.

9.4.2 Handling Modules

The following commands are available:

`insmod` `insmod` loads the requested module after searching for it in a subdirectory of `/lib/modules/<version>/`. It is better, however, to use `modprobe` rather than `insmod`.

`rmmod` Unloads the requested module. This is only possible if this module is no longer needed. For example, the `isofs` module cannot be unloaded while a CD is still mounted.

`depmod` Creates the file `modules.dep` in `/lib/modules/<version>/` that defines the dependencies of all the modules. This is necessary to ensure that all dependent modules are loaded with the selected ones. This file will be built after the system is started if it does not exist.

`modprobe` Loads or unloads a given module while taking into account dependencies of this module. This command is extremely powerful and can be used for a lot of things (e.g., probing all modules of a given type until one is successfully loaded). In contrast to insmod, modprobe checks `/etc/modprobe.conf` and therefore is the preferred method of loading modules. For detailed information about this topic, refer to the corresponding man page.

`lsmod` Shows which modules are currently loaded as well as how many other modules are using them. Modules started by the kernel daemon are tagged with `autoclean`. This label denotes that these modules will automatically be removed once they reach their idle time limit.

modinfo Shows module information.

9.4.3 /etc/modprobe.conf

The loading of modules is affected by the files `/etc/modprobe.conf` and `/etc/modprobe.conf.local` and the directory `/etc/modprobe.d`. See `man modprobe.conf`. Parameters for modules that access hardware directly must be entered in this file. Such modules may need system-specific options (e.g., CD-ROM driver or network driver). The parameters used here are described in the kernel sources. Install the package `kernel-source` and read the documentation in the directory `/usr/src/linux/Documentation/`.

9.4.4 Kmod — the Kernel Module Loader

The kernel module loader is the most elegant way to use modules. Kmod performs background monitoring and makes sure the required modules are loaded by modprobe as soon as the respective functionality is needed in the kernel.

To use Kmod, activate the option 'Kernel module loader' (`CONFIG_KMOD`) in the kernel configuration. Kmod is not designed to unload modules automatically; in view of today's RAM capacities, the potential memory savings would be marginal. For reasons of performance, monolithic kernels may be more suitable for servers that are used for special tasks and need only a few drivers.

9.5 Settings in the Kernel Configuration

All the kernel's configuration options cannot be covered here in detail. Make use of the numerous help texts available on kernel configuration. The latest kernel documentation is always in `/usr/src/linux/Documentation/`.

9.6 Compiling the Kernel

Compiling a “bzImage” is recommanded. As a rule, this avoids the problem of the kernel getting too large, as can easily happen if you select too many features and create a “zImage”. You will then get error messages like "kernel too big" or "System is too big".

After customizing the kernel configuration, start compilation by entering (remember to change into the directory `/usr/src/linux/`, first):

```
make clean
make bzImage
```

```
make clean
make vmlinux
```

► **ipseries**
If you need a kernel for an RS/6000 that can be booted from a floppy disk, enter `make zImage`. ◄

These two commands can be entered as one command line:

```
make clean bzImage
```

```
make clean vmlinux
```

► **x86, x86_86, ia64, ipseries, zseries**
After a successful compilation, find the compressed kernel in `/usr/src/linux/arch/<arch>/boot/`. The kernel image — the file that contains the kernel — is called `vmlinux.gz`. ◄

► **ipseries**
After a successful compilation, find the kernel in `/usr/src/linux/`. The kernel image — the file that contains the kernel — is called `vmlinux`; the floppy disk kernel for RS/6000 however is stored in `/usr/src/linux/arch/<arch>/chrpboot`. ◄

If you cannot find this file, an error probably occurred during the kernel compilation. In the Bash shell, enter the following command to launch the kernel compilation again and write the output to a file `kernel.out`:

```
make bzImage 2>&1 | tee kernel.out
```

```
make clean vmlinux 2>&1 | tee kernel.out
```

If you have configured parts of your kernel to load as modules, launch the module compilation. Do this with `make modules`.

9.7 Installing the Kernel

After the kernel is compiled, it must be installed so it can be booted.

► x86
If you use LILO, LILO must be updated as well. To prevent unpleasant surprises, it is recommended to keep the old kernel (e.g., as `/boot/vmlinuz.old`), so you can still boot it if the new kernel does not function as expected: ◄

```
cp /boot/vmlinuz /boot/vmlinuz.old
cp arch/i386/boot/bzImage /boot/vmlinuz
lilo
```

► x86
The Makefile target `make bzlilo` performs all three of these steps. ◄

Note

If you use GRUB as the boot loader, it does *not* need to be reinstalled. Simply carry out the first two steps to copy the kernel to the right location in the system.

Note

BootX: The above-mentioned file `vmlinux` must be copied to 'Linux Kernels' in the system directory. Next, when BootX is started, the new kernel must be selected before clicking 'Save to prefs'.

RS/6000 (quik): Copy `vmlinux` to `boot/` and adjust `/etc/quik.conf`.

RS/6000 (disk): Copy `zImage` from `/usr/src/linux/arch/<arch>/chrpboot` to DOS floppy disk.

Now the compiled modules need to be installed. Enter `make modules_install` to copy them to the correct target directories in `/lib/modules/<version>/`. If the kernel version is the same, the old modules will be overwritten. However, the original modules can be reinstalled together with the kernel from the CDs.

Note

To avoid unexpected effects, make sure that modules whose functionalities may now have been directly compiled into the kernel are removed from `/lib/modules/<version>/`. This is one of the reasons why inexperienced users are *strongly* discouraged from compiling the kernel.

Note

► **x86**
To enable GRUB or LILO to boot the old kernel (now `/boot/vmlinuz.old`), add an image entry with the label `Linux.old` in your `/boot/grub/menu.lst` or `/etc/lilo.conf` This procedure is described in detail in Chapter 8 on page 199. If you are using LILO as the boot loader, LILO must be reinstalled after modifications to `/etc/lilo.conf` with the command `lilo`. GRUB does not need to be reinstalled. ◄ ► **x86**
The file `/boot/System.map` contains kernel symbols required by the modules to ensure successful launching of kernel functions. This file depends on the current kernel. Therefore, once you have compiled and installed the kernel, copy `/usr/src/linux/System.map` to the directory (`/boot/`). This file is regenerated each time the kernel is recompiled. If you create your kernel using `make bzlilo` or `make zlilo`, this is done for you automatically. If you get an error message like "System.map does not match current kernel", `System.map` probably has not been copied. ◄

9.8 Cleaning Your Hard Disk after Compilation

If you are low on hard disk space, delete the object files generated during kernel compilation using `make clean` in the `/usr/src/linux/` directory. If you have plenty of disk space and plan to reconfigure the kernel on a regular basis, you might want to skip this. Recompiling the kernel is considerably faster then, because only the parts affected by changes will actually be recompiled.

10

Special Features of SUSE LINUX

This chapter provides information about the *Filesystem Hierarchy Standard* (FHS) and *Linux Standard Base* (LSB). Various software packages and special features, such as booting with *initrd*, `linuxrc`, and the rescue system, are described in detail.

10.1 Linux Standards 238
10.2 Hints on Special Software Packages 239
10.3 Booting with the Initial Ramdisk 245
10.4 The SUSE Rescue System 249
10.5 Virtual Consoles 254
10.6 Keyboard Mapping 254
10.7 Local Adjustments — I18N and L10N 255

10.1 Linux Standards

10.1.1 Linux Standard Base (LSB)

SUSE actively supports the efforts of the *Linux Standard Base* project. Up-to-date information about the project can be found at `http://www.linuxbase.org`. The currently valid LSB specification is at version 1.3.x. Apart from the File System Hierarchy Standard (FHS), which now forms part of it, the specification defines things like the package format and details of the system initialization (see Chapter 11 on page 259).

The LSB specification currently only comprises the x86 architecture.

10.1.2 File System Hierarchy Standard (FHS)

In accordance with the LSB specification, SUSE LINUX is also compliant with the *File System Hierarchy Standard* or FHS (package `fhs`). Also see `http://www.pathname.com/fhs/`. For this reason, in some cases it was necessary to move files or directories to their *correct* places in the file system, as specified by the FHS.

For example, one aim of the FHS is to define a structure with the help of which `/usr` can be mounted *read-only*.

10.1.3 teTeX — TeX in SUSE LINUX

TeX is a comprehensive typesetting system that runs on various platforms. It can be expanded with macro packages, like LaTeX, and consists of numerous files that have to be organized according to the *TeX Directory Structure* (TDS) (see `ftp://ftp.dante.de/tex-archive/tds/`). teTeX is a compilation of current TeX software. On a SUSE LINUX system, teTeX is installed in a way that ensures compliance with the requirements of both the TDS and the FHS.

10.1.4 Example Environment for FTP Server

To make it easier to set up an FTP server, the `ftpdir` package includes an example environment. This is installed in `/srv/ftp`.

10.1.5 Example Environment for HTTP Server

Apache is the standard web server in SUSE LINUX. Together with the installation of Apache, some example documents are made available in `/srv/www`. To set up your own web server, include your own `DocumentRoot` in `/etc/httpd/httpd.conf` and store your files (documents, picture files) accordingly.

10.2 Hints on Special Software Packages

10.2.1 Package bash and /etc/profile

1. `/etc/profile`
2. `~/.profile`
3. `/etc/bash.bashrc`
4. `~/.bashrc`

Users can make personal entries in `~/.profile` or in `~/.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` respectively into the home directory of the user. It is recommended to copy the settings from `/etc/skel` following 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
```

The personal adjustments then need to be copied back from the files `*.old`.

10.2.2 cron Package

The `cron` tables are now located in `/var/cron/tabs`. `/etc/crontab` serves as a system-wide cron table. Enter the name of the user who should run the command directly after the time table (see File 10.1, here `root` is entered). Package-specific tables, located in `/etc/cron.d`, have the same format. See `man cron`.

Example 10.1: *Example of an Entry in /etc/crontab*

```
1-59/5 * * * * root test -x /usr/sbin/atrun && /usr/sbin/atrun
```

`/etc/crontab` cannot be processed with `crontab -e`. It 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 instructions are controlled by `/usr/lib/cron/run-crons`. `/usr/lib/cron/run-crons` is run every fifteen minutes from the main table (`/etc/crontab`). This guarantees that processes that may have been neglected can be run at the proper time.

The daily system maintenance jobs have been distributed to various scripts for reasons of clarity. Apart from `aaa_base`, `/etc/cron.daily` contains, for instance, the components `backup-rpmdb`, `clean-tmp`, or `clean-vi`.

10.2.3 Log Files: the Package logrotate

There are a number of system services (*daemons*), which, 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.

Changing to logrotate

The old settings listed below are adopted when updating from a version older than SUSE LINUX 8.0:

- Entries from `/etc/logfile` not associated with a particular package are moved to `/etc/logrotate.d/aaa_base`.
- The variable `MAX_DAYS_FOR_LOG_FILES` from the former `rc.config` file is mapped as `dateext` and `maxage` in the configuration file. Refer to `man logrotate`.

Configuration

Configure logrotate with the file `/etc/logrotate.conf`. In particular, the `include` specification primarily configures the additional files to read. SUSE LINUX ensures that individual packages install files in `/etc/logrotate.d` (e.g., `syslog` or `yast`).

Example 10.2: *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 it is called daily by `/etc/cron.daily/logrotate`.

Note

The `create` option reads all settings made by the administrator in `/etc/permissions*`. Ensure that no conflicts arise from any personal modifications.

Note

10.2.4 Man 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 or 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 either via Emacs, by entering `emacs -f info`, or directly in a console with `info`. You can also use tkinfo, xinfo, or the SUSE help system to view info pages in a convenient way.

10.2.5 The Command ulimit

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 10.1.

Table 10.1: *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

System-wide settings can be made in `/etc/profile`. There, creating core files must be enabled, needed by programmers for *debugging*. A normal user cannot increase the values specified in `/etc/profile` by the system administrator, but he can make special entries in his own`~/.bashrc`.

Example 10.3: *ulimit: Settings in /.bashrc*

```
# Limits of physical memory:
ulimit -m 98304

# Limits of virtual memory:
ulimit -v 98304
```

Details of memory must be specified in KB. For more detailed information, see `man bash`.

Note

Not all shells support `ulimit` directives. PAM (for instance, `pam_limits`) offers comprehensive adjustment possibilities if you depend on encompassing settings for these restrictions.

Note

10.2.6 The free Command

The `free` command is somewhat misleading if your goal is to find out how much RAM is currently being used. The relevant information can be found in `/proc/meminfo`. These days, users, who have 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 will be either 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`).

Furthermore, 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`.

10.2.7 The File /etc/resolv.conf

Domain name resolution is handled through the file `/etc/resolv.conf`. Refer to Section 21.7 on page 455 about this.

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.

10.2.8 Settings for GNU Emacs

GNU Emacs is a complex work environment. More information is available at `http://www.gnu.org/software/emacs/`. The following sections cover the configuration files processed when GNU Emacs is started.

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`. If a user wants to customize the program, the `.gnu-emacs` should be copied to the home directory and the desired settings should be specified there:

```
cp /etc/skel/.gnu-emacs ~/.gnu-emacs
```

`.gnu-emacs` defines the file `~/.gnu-emacs-custom` as `custom-file`. If users make settings with the `customize` options, these are saved to `~/.gnu-emacs-custom`.

With SUSE LINUX, 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 system-wide 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 split in several packages:

- The base package `emacs`.
- Usually, `emacs-x11` should be installed. This package contains the program *with* X11 support.
- `emacs-nox` contains the program *without* X11 support.
- `emacs-info`: online documentation in info format.
- `emacs-el` contains the uncompiled library files in Emacs Lisp. These are not required at run-time.
- Numerous add-on packages that can be installed if needed: `emacs-auctex` (for LaTeX), `psgml` (for SGML and XML), `gnuserv` (for client and server operation), and others.

10.3 Booting with the Initial Ramdisk

As soon as the Linux kernel has been booted and the root file system (`/`) mounted, programs can be run and further kernel modules can be integrated to provide additional functions. To mount the root file system, certain conditions must be met. The kernel needs the corresponding drivers to access the device on which the root file system is located (especially SCSI drivers). The kernel must also contain the code needed to read the file system (`ext2`, `reiserfs`, `romfs`, etc.). It is also conceivable that the root file system is already encrypted. In this case, a password is needed to mount the file system.

For the problem of SCSI drivers, a number of different solutions are possible. The kernel could contain all imaginable drivers, but this might be a problem because different drivers could conflict with each other. Also, the kernel would become very large because of this. Another possibility is to provide different kernels, each one containing just one or a few SCSI drivers. This method has the problem that a large number of different kernels are required, a problem then increased by the differently optimized kernels (Athlon optimization, SMP). The idea of loading the SCSI driver as a module leads to the general problem resolved by the concept of an *initial ramdisk*: running user space programs even before the root file system is mounted.

10.3.1 Concept of the Initial Ramdisk

The *initial ramdisk* (also called *initdisk* or *initrd*) solves precisely the problems described above. The Linux kernel provides an option of having a small file system loaded to a RAM disk and running programs there before the actual root file system is mounted. The loading of `initrd` is handled by the boot loader (GRUB, LILO, etc.). Boot loaders only need BIOS routines to load data from the boot medium. If the boot loader is able to load the kernel, it can also load the initial ramdisk. Special drivers are not required.

10.3.2 The Order of the Booting Process with initrd

The boot loader loads the kernel and the `initrd` to memory and starts the kernel. The boot loader informs the kernel that an `initrd` exists and where it is located in memory. If the `initrd` was compressed (which is typically the case), the kernel decompresses the `initrd` and mounts it as a temporary root file system. A program called linuxrc is then started. This program can now do all the things necessary to mount the proper root file system.

As soon as linuxrc finishes, the temporary `initrd` is unmounted and the boot process continues as normal with the mount of the proper root file system. Mounting the `initrd` and running linuxrc can be seen as a short interlude during a normal boot process.

The kernel tries to remount `initrd` to the `/initrd` immediately after the actual root partition is booted. If this fails because the mount point `/initrd` does not exist, for example, the kernel will attempt to unmount `initrd`. If this does not work, the system is fully functional, but the memory taken up by `initrd` cannot be unlocked, so will no longer be available.

linuxrc

The only requirements for the program linuxrc in the `initrd` are: it must have the special name `linuxrc`, it must be located in the root directory of the `initrd`, and it needs to be executable by the kernel. This means that linuxrc may be dynamically linked. In this case, the shared libraries in `/lib` must be completely available in `initrd`. linuxrc can also be a shell script. For this to work, a shell must exist in `/bin`. In short, `initrd` must contain a minimal Linux system that allows the program linuxrc to be run. When SUSE LINUX is installed, a statically-linked linuxrc is used to keep `initrd` as small as possible. linuxrc is run with `root` permissions.

The Real Root File System

As soon as linuxrc terminates, `initrd` is unmounted and discarded, the boot process carries on as normal, and the kernel mounts the real file system. What is mounted as the root file system can be influenced by linuxrc. It just needs to mount the `/proc` file system and write the value of the real root file system in numerical form to `/proc/sys/kernel/real-root-dev`.

10.3.3 Boot Loaders

Most boot loaders, including GRUB, LILO, and syslinux, can handle `initrd`. Give individual boot loaders instructions for accessing `initrd` as follows:

GRUB Enter the following line in `/boot/grub/menu.lst`:

```
initrd (hd0,0)/initrd
```

As the loading address of the initrd is written to the loaded kernel image, the `initrd` command must follow the `kernel` command.

LILO Enter the following line in `/etc/lilo.conf`:

```
initrd=/boot/initrd
```

syslinux Enter the following line in `syslinux.cfg`:

```
append initrd=initrd
```

Further parameters can be appended to the line.

10.3.4 Using initrd in SUSE

Installing the System

The `initrd` has already been used for some time for the installation: the user can load modules and make the entries necessary for installation. linuxrc then starts YaST, which carries out the installation. When YaST has finished, it tells linuxrc where the root file system of the newly installed system is located. linuxrc writes this value to `/proc` and reboots the system. Then YaST starts again and installs the remaining packages in the system being installed.

Booting the Installed System

In the past, YaST offered more than forty kernels for installing in the system. The main difference between the kernels was that each of them contained a specific SCSI driver. This was necessary to be able to mount the root file system after booting. Further drivers could then be loaded afterwards as modules. As optimized kernels are now available, this concept is no longer feasible — by now, over one hundred kernel images would be needed.

This is why an `initrd` is used now even to start the system normally. The way it is used is similar to the method for installation. The linuxrc used here, however, is simply a shell script with the task of loading a given module. Typically, this is just one single module — the very SCSI driver needed to access the root file system.

Creating an initrd

An `initrd` is created by means of the script `mkinitrd` (previously `mk_initrd`). In SUSE LINUX, 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 installation linuxrc saves which modules were loaded). The modules are loaded in exactly the order in which they appear in `INITRD_MODULES`. This is especially important if several SCSI drivers are used, because otherwise the names of the hard disks would change. Strictly speaking, it would be sufficient just to load those drivers needed to access the root file system. However, all SCSI drivers needed for installation are loaded by means of `initrd` because later loading could be problematic.

Note

As the `initrd` is loaded by the boot loader in the same way as the kernel itself (in its `map` file, LILO records the location of the files), the boot loader LILO must be updated every time the `initrd` is modified. This is not necessary for GRUB.

Note

10.3.5 Possible Difficulties — Custom Kernels

A custom kernel can often lead to the following problem: out of habit, the SCSI driver is hard-linked to the kernel, but the existing `initrd` remains unchanged. When you boot, the following occurs: The kernel already contains the SCSI driver, so the hardware is detected. `initrd`, however, now tries to load the driver as a module. With some SCSI drivers, especially with `aic7xxx`, this leads to the system locking. Strictly speaking, this is a kernel error. An already existing driver should not be allowed to be loaded again as a module. The problem is already known from another context, however (serial drivers).

There are several solutions to the problem. Configure the driver as a module (then it will be correctly loaded in the `initrd`. Alternatively, remove the entry for `initrd` from the file `/etc/grub/menu.lst` or `/etc/lilo.conf`, depending on your boot loader. An equivalent to the latter solution is to remove the variable `INITRD_MODULES` then run `mkinitrd`, which then realizes that no `initrd` is needed.

10.3.6 Prospects

It is quite possible in the future that an `initrd` will be used for many more and much more sophisticated things than loading modules needed to access `/`.

- Root file system on software RAID (linuxrc sets up the `md` devices)
- Root file system on LVM
- Root file system is encrypted (linuxrc queries the password)
- Root file system on a SCSI hard disk on a PCMCIA adapter

For more information, see `/usr/src/linux/Documentation/ramdisk.txt`, `/usr/src/linux/Documentation/initrd.txt`, and the man page for initrd.

10.4 The SUSE Rescue System

SUSE LINUX contains a rescue system for accessing your Linux partitions *from the outside* in the event of an emergency. The rescue system can be loaded from CD, the network, or the SUSE FTP server. Furthermore, there is a bootable SUSE LINUX CD (the *LiveEval CD*) that can be used as a rescue system. The rescue system includes several help programs with which you can remedy large problems with inaccessible hard disks, misconfigured configuration files, or other similar problems.

Another component of the rescue system is Parted, which is used for resizing partitions. This program can be launched from within the rescue system, if you do not want to use the resizer integrated in YaST. Information about Parted can be found at `http://www.gnu.org/software/parted/`.

10.4.1 Starting the Rescue System

The rescue system is launched from CD (or DVD). The CD-ROM or DVD drive must be bootable. If necessary, change the boot sequence in the BIOS setup.

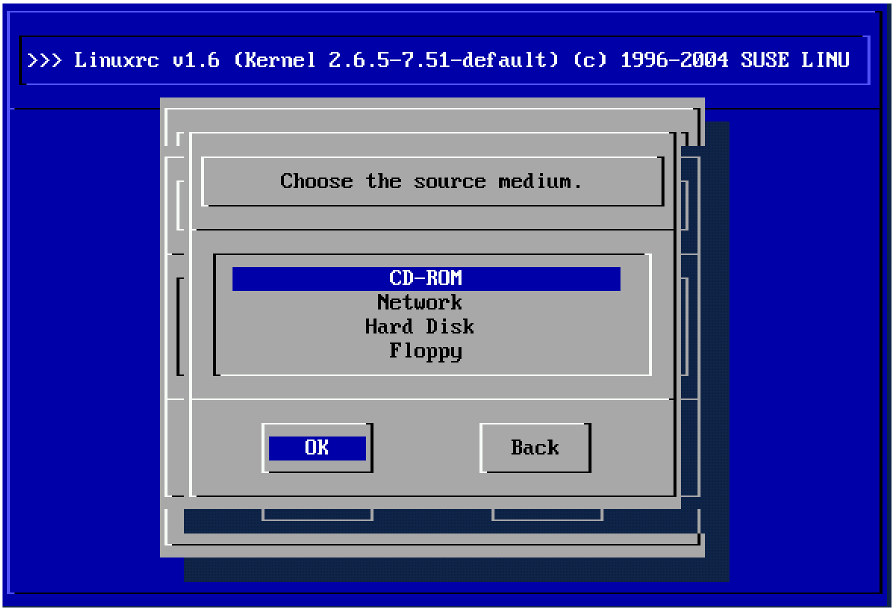
Proceed as follows to start the rescue system:

1. Insert the first SUSE LINUX CD or DVD in the respective drive and turn on your system.
2. Allow the system to boot or select 'Manual Installation' to enter special boot parameters under 'boot options'.
3. In linuxrc, select the correct language and keyboard layout.
4. Load the kernel modules required for your system. Load *all* modules you need for the rescue system. Due to limited space, the rescue system itself contains only very few modules.
5. Select 'Start Installation or System' in the main menu.
6. Select 'Start Rescue System' (see Figure 3.7 on page 112) and specify the desired source medium (Figure 10.1 on the facing page).

   'CD-ROM': Uses the rescue system on the CD-ROM.

   'Network': Starts the rescue system over a network connection. The kernel module for the network card must be loaded first (see Section 3.5.2 on page 120). A submenu offers protocols such as NFS, FTP, and SMB (see Figure 10.2 on page 252).

   'Hard Disk': If you previously copied a rescue system to a hard disk to which you have access, its location can be specified here. Subsequently, this rescue system will be loaded.



```
>>> Linuxrc v1.6 (Kernel 2.6.5-7.51-default) (c) 1996-2004 SUSE LINU
```

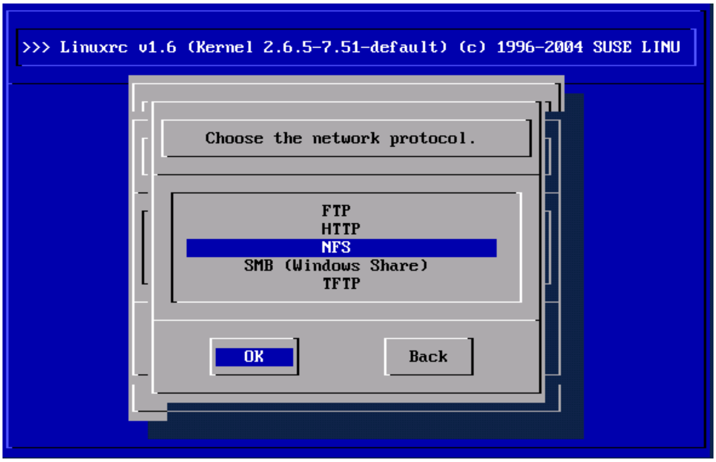
Choose the source medium.

Figure 10.1: Source Medium for the Rescue System

Regardless of the medium chosen, the rescue system will be decompressed, loaded onto a RAM floppy disk as a new root file system, mounted, and started. Now it is ready for use.

10.4.2 Working with the Rescue System

Under Alt + F1 to Alt + F3, the rescue system provides at least three virtual consoles. You can log in as `root` without a password. Press Alt + F10 to enter the system console displaying the kernel and syslog messages.


>>> Linuxrc v1.6 (Kernel 2.6.5-7.51-default) (c) 1996-2004 SUSE LINU

Choose the network protocol.

Figure 10.2: *Network Protocols*

A shell and many other useful utilities, such as the mount program, can be found in the `/bin` directory. The `sbin` directory contains important file and network utilities for reviewing and repairing the file system (e.g., e2fsck). This directory also contains the most important binaries for system maintenance, such as fdisk, mkfs, mkswap, mount, mount, init, and shutdown, as well as ifconfig, route, and netstat for maintaining the network. The directory `/usr/bin` contains the vi editor, grep, find, less, and telnet.

Accessing Your Normal System

To mount your SUSE LINUX system using the rescue system, use the mount point `/mnt`. You can also use or create another directory. The following example demonstrates the procedure for a system with the `/etc/fstab` details shown in File 10.4.

Example 10.4: *Example /etc/fstab*

```
/dev/sdb5    swap   swap    defaults   0   0
/dev/sdb3    /      ext2    defaults   1   1
/dev/sdb6    /usr   ext2    defaults   1   2
```

Caution

Pay attention to the order of steps outlined in the following section for mounting the various devices.

Caution

To access your entire system, mount it step by step in the `/mnt` directory using the following commands:

```
mount /dev/sdb3 /mnt
mount /dev/sdb6 /mnt/usr
```

Now, you can access your entire system and, for example, correct mistakes in configuration files, such as `/etc/fstab`, `/etc/passwd`, and `/etc/inittab`. The configuration files are now located in the `/mnt/etc` directory instead of in `/etc`.

Before you recover lost partitions with the fdisk program by simply setting them up again, you should *first* make a printout of `/etc/fstab` and the output of `fdisk -l`.

Repairing File Systems

Damaged file systems are tricky problems for the rescue system. 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 `kernel panic`. In this case, the only way is to repair the system from the *outside* using a rescue system.

The SUSE LINUX rescue system contains the utilities e2fsck and dumpe2fs (for diagnosis). These should remedy most problems. In an emergency, man pages often are not available. For this reason, they are included in this manual in Appendix B on page 747.

If mounting a file system fails due to an *invalid* superblock, the e2fsck program would probably fail, too. If this were the case, your superblock may be corrupted, too. There are copies of the superblock located every 8192 blocks (8193, 16385, etc.). If your superblock is corrupted, try one of the copies instead. This is accomplished by entering the command `e2fsck -f -b 8193 /dev/damaged_partition`. The `-f` option forces the file system check and overrides e2fsck's error so that, since the superblock copy is intact, everything is fine.

10.5 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 X11. More or fewer consoles can be assigned by modifying the file `/etc/inittab`.

To switch to a console from X11 without leaving X11, use Ctrl-Alt-F1 to Ctrl-Alt-F6. The key Alt-F7 then returns to X11.

10.6 Keyboard Mapping

To standardize the keyboard mapping of programs, changes were made to the following files:

```
/etc/inputrc
/usr/X11R6/lib/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/X11R6/lib/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.). Other non-SUSE applications should be adjusted to these defaults.

Under X, the compose key (*multikey*) can be accessed using the key combination Ctrl + Shift (right). Also see the corresponding entry in `/usr/X11R6/lib/X11/Xmodmap`.

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`.

10.7 Local Adjustments — I18N and L10N

SUSE LINUX 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 manual page `man locale`).

1. `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 govern the above categories. The files concerned are listed below. The current setting can be shown with the command `locale`.
2. `RC_LC_ALL`: This variable (if set) overwrites the values of the variables mentioned above.
3. `RC_LANG`: If none of the above variables are set, this is the fallback. By default, SUSE LINUX only sets `RC_LANG`. This makes it easier for users to enter their own values.
4. `ROOT_USES_LANG`: A `yes` or `no` variable. If it is set to `no`, `root` always works in the POSIX environment.

The other variables can be set via the YaST sysconfig editor. 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>]]
```

10.7.1 Some Examples

You should always set the language and country codes together. Language settings follow the standard ISO 639 (`http://www.evertype.com/standards/iso639/iso639-en.html` and `http://www.loc.gov/standards/iso639-2/`). Country codes are listed in ISO 3166, see (`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`. 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 English is selected upon 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 variable to English language, country to United States, and the character set to `ISO-8859-1`. This character set does not support the Euro sign, but it will be useful sometimes for programs that have not been updated to support `UTF-8`. The string defining the charset (`ISO-8859-1` in our case) will then be evaluated by programs like Emacs.

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 system-wide.

Users can override the system defaults by editing their `~/.bashrc` accordingly. For instance, if you do not want to use the system-wide `en_US` for program messages, you could include `LC_MESSAGES=es_ES` so messages are displayed in Spanish instead.

10.7.2 Settings for Language Support

Files in the category *Messages* are, as a rule, only stored in the corresponding language directory (for instance `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 will fall 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.

Possible Problems

The thousands comma is not recognized. `LANG` is probably set to `en`, but the description the glibc uses is located in `/usr/share/locale/en_US/LC_NUMERIC`. `LC_NUMERIC`, for example, must be set to `en_US`.

For More Information

- *The GNU C Library Reference Manual*, Chap. "Locales and Internationalization"; 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 `file:/usr/share/doc/howto/en/html/Unicode-HOWTO.html`.

11

The SUSE LINUX Boot Concept

Booting and initializing a UNIX system can challenge even an experienced system administrator. This chapter gives a short overview of the SUSE LINUX boot concept. The new implementation is compatible with the *System Initialization* section of the LSB specification (Version 1.3.x). Refer to Section 10.1.1 on page 238 for more information about LSB.

11.1 The init Program 260
11.2 Runlevels 260
11.3 Changing Runlevels 262
11.4 Init Scripts 263
11.5 The YaST Runlevel Editor 266
11.6 SuSEconfig and /etc/sysconfig 268
11.7 The YaST sysconfig Editor 270

The kernel takes control of the system's hardware as soon as the simple message "Uncompressing Linux..." is printed on screen (or, in the case of the IBM S/390 and zSeries, after IPLing). The kernel checks and sets the console (the BIOS registers of graphics cards and the screen output format), reads BIOS settings, and initializes basic hardware interfaces. Next, the drivers, which are part of the kernel, probe existing hardware and initialize it accordingly. After checking the partitions and mounting the root file system, the kernel starts init, which *boots* (Unix jargon) the main system with all its programs and configurations. The kernel controls the entire system, managing hardware access and allocating CPU time and memory to programs.

11.1 The init Program

The program init is the process responsible for initializing the system itself in the required way. All other processes are considered child processes of init. init takes a special role. It 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. Here, the *runlevels* are defined (see Section 11.2). It 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 all reside in the directory `/etc/init.d/`.

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 it is to maintain all other processes and to adjust CPU time and hardware access according to requests from other programs.

11.2 Runlevels

In Linux, *runlevels* define how the system is started. After booting, the system starts as defined in `/etc/inittab` in the line `initdefault`. Usually this is `3` or `5` (see Table 11.1 on the next page). As an alternative, the runlevel can be specified at boot time (at the boot prompt, for instance). Any parameters that are not directly evaluated by the kernel itself are passed to init.

To change runlevels while the system is running, enter `init` and the corresponding number as an argument. Only the system administrator is allowed to do this. `init 1` (or `shutdown now`) causes the system to change to *single user mode*, which is used for system maintenance and administration. After finishing his work, the administrator can switch back to the normal runlevel by entering `init 3`, which starts all the essential programs and allows regular users to log in and to work with the system. `init 0` (or `shutdown -h now`) causes the system to halt. `init 6` (or `shutdown -r now`) causes it to shut down with a subsequent reboot.

Note

Runlevel 2 with a /usr/ Partition Mounted via NFS

You should not use runlevel 2 if your system mounts the `/usr/` partition via NFS. The `/usr/` directory holds important programs essential for the proper functioning of the system. Because the NFS service is not made available by runlevel 2 (local multiuser mode without remote network), the system would be seriously restricted in many aspects.

Note

Table 11.1: *Available Runlevels*

Runlevel	Description
0	System halt
S	Single user mode; from the boot prompt, only with US keyboard
1	Single user mode
2	Local multiuser mode without remote network (e.g., NFS)
3	Full multiuser mode with network
4	Not used
5	Full multiuser mode with network and X display manager — KDM (default), GDM, or XDM
6	System reboot

Runlevel `5` is the default runlevel in all SUSE LINUX standard installations. Users are prompted for login directly under a graphical interface.

However, if the default runlevel is `3` and you want to change it to `5`, you first need to configure the X Window System in the required way (see

Chapter 12 on page 273). After doing so, check whether the system works in the desired way by entering `init 5`. If everything turns out as expected, you can use YaST to set the default runlevel to 5.

Caution

Modifying /etc/inittab

If `/etc/inittab` is damaged, the system might not boot properly. Therefore, be extremely careful while editing `/etc/inittab` and always keep a backup of an intact version. To repair damage, try entering `init=/bin/sh` after the kernel name at the boot prompt to boot directly into a shell. After that, replace `/etc/inittab` with your backup version using `cp`.

Caution

11.3 Changing Runlevels

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`) tells init to change to a different runlevel by entering `init 5`.
2. init consults its configuration file (`/etc/inittab`) and determines it should start `/etc/init.d/rc` with the new runlevel as a parameter.
3. Now rc calls all the stop scripts of the current runlevel, but only for those where there is no start script in the new runlevel. In this example, these are all the scripts that reside in `/etc/init.d/rc3.d/` (old runlevel was 3) and start with a K. The number following K specifies the order to start, as there are some dependencies to consider.
4. The last things to start are the start scripts of the new runlevel. These are, in this example, in `/etc/init.d/rc5.d/` and begin with an S. The same procedure regarding the order in which they are started is applied here.

When changing into the same runlevel as the current runlevel, init only checks `/etc/inittab` for changes and starts the appropriate steps (e.g., for starting a getty on another interface).

11.4 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 S/390 and zSeries systems, this is the case only during the boot process or if an immediate system shutdown is initiated (power failure or via “signal quiesce”).
- 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 for changing the runlevel are also found there, but 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 (e.g., if they are used in several runlevels). Because every script can be executed as both a start and a stop script, these scripts must understand the parameters `start` and `stop`. The scripts also understand the `restart`, `reload`, `force-reload`, and `status` options. These different options are explained in Table 11.2.

***Table 11.2:** 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 current status of service.

Links in each runlevel-specific subdirectory make it possible to associate scripts with different runlevels. When installing or uninstalling packages, such 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 manual page of insserv for details.

Below is a short introduction to the boot and stop scripts launched first (or last, respectively) 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 `pts/` file systems are mounted and the 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* by rc before any other one. It is stopped after the actions triggered by the above scripts (running a number of subscripts, for example) are completed. The blogd daemon 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. Further information about blogd can be obtained with `man blogd`.

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 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.

11.4.1 Adding init Scripts

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/`, and `insserv`. Additionally consult the man pages of `startproc` and `killproc`.

Caution

Creating Your Own init Scripts

Faulty init scripts may freeze your machine. Edit such scripts with great care and, if possible, subject them to heavy testing in the multiuser environment. Some useful information about init scripts can be found in Section 11.2 on page 260.

Caution

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 file names, 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 should be edited. See Example 11.1.

Example 11.1: *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, respectively, 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. Under `Default-Start:` and `Default-Stop:`, specify the runlevels in which the service should automatically be started or stopped. Finally, under `Description:`, provide a short description of the service in question.

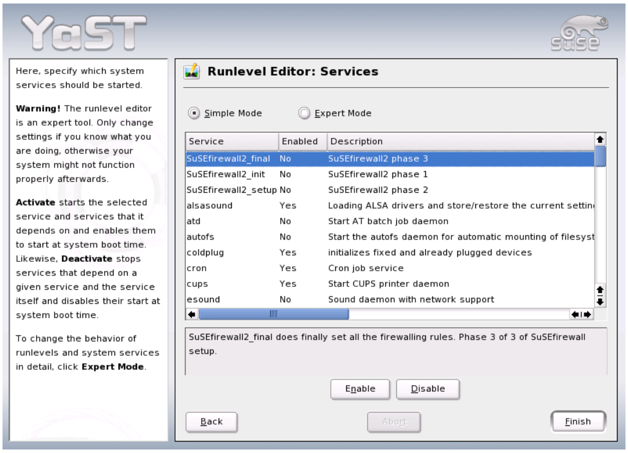
To create the links from `/etc/init.d/` to the corresponding runlevel directories (`/etc/init.d/rc?.d/`), enter the command `insserv <new-script-name>`. The insserv program evaluates the `INIT INFO` header to create the necessary links for start and stop scripts in the runlevel directories (`/etc/init.d/rc?.d/`). 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 11.5.

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 will be started automatically.

11.5 The YaST Runlevel Editor

After starting this YaST module, it displays an overview listing all the available services and the current status of each service — whether they are enabled. Decide whether to use the module in 'Simple Mode' or in 'Expert Mode'. The default 'Simple Mode' should be sufficient for most purposes. The left column shows the name of the service, the center column indicates its current status, and the right column gives a short description. For the selected service, a more detailed description is provided in the lower part of the window. To enable a service, select it in the table then select 'Enable'. The same steps apply to disable a service.

For detailed control over the runlevels in which a service is started or stopped or to change the default runlevel, first select 'Expert Mode'. In this mode, the dialog displays the current default runlevel or "initdefault" (the runlevel into which the system boots by default) at the top. Normally, the default runlevel of a SUSE LINUX system is runlevel 5 (full multiuser mode with network and X). A suitable alternative might be runlevel 3 (full multiuser mode with network).



Here, specify which system services should be started.

Warning! The runlevel editor is an expert tool. Only change settings if you know what you are doing, otherwise your system might not function properly afterwards.

Activate starts the selected service and services that it depends on and enables them to start at system boot time. Likewise, **Deactivate** stops services that depend on a given service and the service itself and disables their start at system boot time.

To change the behavior of runlevels and system services in detail, click **Expert Mode**.

Service	Enabled	Description
SuSEfirewall2_final	No	SuSEfirewall2 phase 3
SuSEfirewall2_init	No	SuSEfirewall2 phase 1
SuSEfirewall2_setup	No	SuSEfirewall2 phase 2
alsasound	Yes	Loading ALSA drivers and store/restore the current settin
atd	No	Start AT batch job daemon
autofs	No	Start the autofs daemon for automatic mounting of filesyst
coldplug	Yes	initializes fixed and already plugged devices
cron	Yes	Cron job service
cups	Yes	Start CUPS printer daemon
esound	No	Sound daemon with network support

SuSEfirewall2_final does finally set all the firewalling rules. Phase 3 of 3 of SuSEfirewall setup.

Figure 11.1: *YaST: Runlevel Editor*

This YaST dialog allows the selection of one of the runlevels (as listed in Table 11.1 on page 261) 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, tells 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 where the selected service or daemon should be running. Runlevel 4 is initially undefined to allow creation of a custom runlevel. Finally, a brief description of the currently selected service or daemon is provided just below the table overview.

With ‘Start, Stop, or Refresh’, decide whether a service should be activated. ‘Refresh status’ can be used to check the current status, if this has not been done automatically. ‘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.

Caution

Changing Runlevel Settings

Faulty runlevel settings may render a system unusable. Before applying your changes, make absolutely sure you know about their consequences.

Caution

11.6 SuSEconfig and /etc/sysconfig

The main configuration of SUSE LINUX can be made with the configuration files in `/etc/sysconfig/`. In the past, SUSE LINUX relied on `/etc/rc.config` for system configuration, but it became obsolete in previous versions. `/etc/rc.config` is not created at installation time, as all system configuration is controlled by `/etc/sysconfig/`. However, if `/etc/rc.config` exists at the time of a system update, it remains intact.

The individual files in `/etc/sysconfig/` are only read by the scripts to which they are relevant. This ensures that network settings, for instance, need to be parsed only by network-related scripts. Many other system configuration files are generated according to the settings in `/etc/sysconfig/`. This task is performed by SuSEconfig. For example, if you change the network configuration, SuSEconfig is likely to make changes to the file `/etc/host.conf` as well, as this is one of the files relevant for the network configuration.

If you change anything in these files manually, run `SuSEconfig` afterwards to make sure all the necessary changes are made in all the relevant places. If you change the configuration using the YaST sysconfig editor, all changes are applied automatically, because YaST automatically starts SuSEconfig to update the configuration files as needed.

This concept enables you to make basic changes to your configuration without needing to reboot the system. Because some changes are rather complex, some programs must be restarted for the changes to take effect. For instance, changes to the network configuration may require a restart of the network programs concerned. This can be achieved by entering the commands `rcnetwork stop` and `rcnetwork start`.

The recommended way to change the system configuration includes the following steps:

1. Bring the system into single user mode (runlevel 1) with `init 1`.
2. Change the configuration files as needed. This can be done using an editor of your choice or with the sysconfig editor of YaST (refer to Section 11.7 on the following page).

Caution

Manual Changes to the System Configuration

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.

Caution

3. Execute `SuSEconfig` to make sure that the changes take effect. If you have changed the configuration files with YaST, this is done automatically.

4. Bring your system back to the previous runlevel with a command like `init 3` (replace `3` with the previous runlevel).

This procedure is mainly relevant when changing system-wide settings (such as the network configuration). Small changes should not require going into single user mode, but you could still do so to make absolutely sure all the programs concerned are correctly restarted.

Note

To disable the automatic configuration of 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.

Note

11.7 The YaST sysconfig Editor

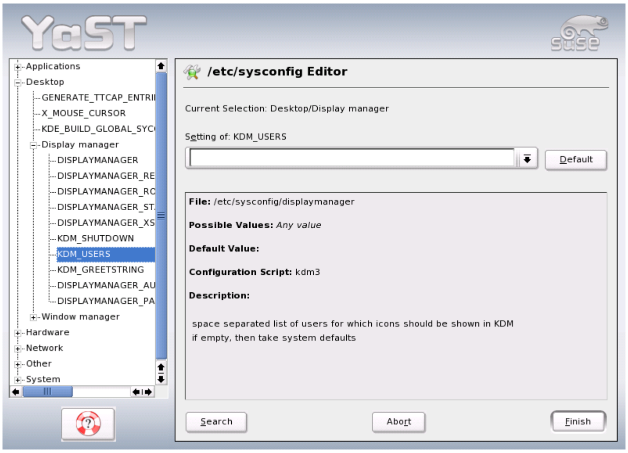
The files in which the most important SUSE LINUX settings are stored are located in the `/etc/sysconfig/` directory. The sysconfig editor presents the options in an easy-to-read manner. The values can be modified and subsequently added to the individual configuration files in this directory. In general, it is not necessary to edit them manually, however, because these files are automatically adjusted when installing a package or configuring a service.

Caution

Modifying /etc/sysconfig/* Files

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.

Caution


YaST
Applications
Desktop
GENERATE_TTCAP_ENTRII
X_MOUSE_CURSOR
KDE_BUILD_GLOBAL_SYC
Display manager
DISPLAYMANAGER
DISPLAYMANAGER_RE
DISPLAYMANAGER_RC
DISPLAYMANAGER_ST
DISPLAYMANAGER_XS
KDM_SHUTDOWN
KDM_USERS
KDM_GREETSTRING
DISPLAYMANAGER_AU
DISPLAYMANAGER_PA
Window manager
Hardware
Network
Other
System
Default
Search
Abort
Finish

/etc/sysconfig Editor

Current Selection: Desktop/Display manager

Setting of: KDM_USERS

File: /etc/sysconfig/displaymanager

Possible Values: *Any value*

Default Value:

Configuration Script: kdm3

Description:

space separated list of users for which icons should be shown in KDM
if empty, then take system defaults

Figure 11.2: *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 asks you to confirm your changes and informs you which scripts will be executed after leaving the dialog by selecting 'Finish'. Also select the services and scripts to skip for now, so they are started later.

The X Window System

The X Window System (X11) is the de facto standard for graphical user interfaces in UNIX. Moreover, X11 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 presents optimization possibilities for your X Window System environment, background information about the use of fonts in SUSE LINUX, and the configuration of OpenGL and 3D. The YaST modules for configuring the mouse and keyboard are covered in the chapter on installation in this manual (section 2.4.4 on page 70).

► **s390, zseries**
This chapter cannot be applied to SUSE LINUX Enterprise Server installations on S/390 and zSeries since the displays of these systems are not supported by XFree. ◄

12.1 Optimizing the Installation of the X Window System 274
12.2 Installing and Configuring Fonts 279
12.3 OpenGL — 3D Configuration 285

12.1 Optimizing the Installation of the X Window System

To use the available hardware (mouse, graphics card, monitor, keyboard) in the best way possible, the configuration can be optimized manually. Some aspects of this optimization are explained below. For detailed information about configuring the X Window System, review the various files in the directory `/usr/share/doc/packages/xf86` and `man XF86Config`.

Caution

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

Caution

The following paragraphs describe the structure of the configuration file `/etc/X11/XF86Config`. Each *section* starts with the keyword `Section <designation>` and ends with `EndSection`. Below is a rough outline of the most important sections.

The programs SaX2 and xf86config create the file `XF86Config`, by default in `/etc/X11`. This is the primary configuration file for the X Window System. Find all the settings here concerning your graphics card, mouse, and monitor.

`XF86Config` consists of several sections, each one dealing with a certain aspect of the configuration. A section always has the same form:

```
Section designation
  entry 1
  entry 2
  entry n
EndSection
```

The following types of sections exist:

Table 12.1: Sections in /etc/X11/XF86Config

Type	Meaning
`Files`	This section describes the paths used for fonts and the RGB color table.
`ServerFlags`	General switches are set here.
`InputDevice`	Input devices, like keyboards and special input devices (touchpads, joysticks, etc.), are configured in this section. Important parameters in this section: `Driver` and the options defining the `Protocol` and `Device`.
`Monitor`	Describes the monitor used. The individual elements of this section are the name, which is referred to later in the `Screen` definition, the `bandwidth`, 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 is to prevent too high frequencies from being sent to the monitor by accident.
`Modes`	The modeline parameters are stored here 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. An exact explanation of the individual parameters would be too much for this book. Find details of the meaning of individual number values in the HOWTO file `/usr/share/doc/howto/en/XFree86-Video-Timings-HOWTO.gz`.
`Device`	This section defines a specific graphics card. It is referenced by its descriptive name.
`Screen`	This section puts together a `Monitor` and a `Device` to form all the necessary settings for XFree. In the `Display` subsection, specify the size of the virtual screen (`Virtual`), the `ViewPort`, and the `Modes`) used with this screen.

`ServerLayout`	This section defines the layout of a single or multihead configuration. This section binds the input devices `InputDevice` and the display devices `Screen`.

`Monitor`, `Device`, and `Screen` are explained in more detail below. Further information about the other sections can be found in the manual pages of `XFree86` and `XF86Config`.

There can be several different `Monitor` and `Device` sections in `XF86Config`. Even multiple `Screen` sections are possible. The following `ServerLayout` section determines which one is used.

12.1.1 Screen Section

First, take a closer look at the screen section, which combines a monitor with a device section and determines the resolution and color depth to use. A screen section might resemble the example in File 12.1.

Example 12.1: *Screen Section of the File /etc/X11/XF86Config*

```
Section "Screen"
  DefaultDepth  16
  SubSection "Display"
    Depth       16
    Modes       "1152x864" "1024x768" "800x600"
    Virtual     1152x864
  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]"
  Monitor       "Monitor[0]"
EndSection
```

The line `Identifier` (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.

Use the `DefaultDepth` setting to select the color depth the server should use unless it is started with a specific color depth. There is a `Display` subsection for each color depth. The keyword `Depth` assigns the color depth valid for this subsection. Possible values for `Depth` are 8, 15, 16, and 24. Not all X server modules support all these values.

After the color depth, a list of resolutions is set in the `Modes` section. This list is checked by the X server from left to right. For each resolution, a suitable `Modeline` is searched in the `Modes` section. The `Modeline` depends on the capability of both the monitor and the graphics card. Thus, the `Monitor` settings will 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 left. This enables you to vary the resolution while X is running.

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. 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 the card has 16 MB video RAM, for example, the virtual screen can be 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 this memory on the card is also used for several font and graphics caches.

12.1.2 Device Section

A device section describes a specific graphics card. You can have as many device entries in `XF86Config` as you like, as long as their names are differentiated, using the keyword `Identifier`. As a rule — 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:

```
Section "Device"
  BoardName     "MGA2064W"
  BusID         "0:19:0"
  Driver        "mga"
  Identifier    "Device[0]"
  VendorName    "Matrox"
  Option        "sw_cursor"
EndSection
```

If you use SaX2 for configuring, the device section should look something like the above diagram. Both the `Driver` and `BusID` are dependent on the hardware installed in your computer and are detected by SaX2 automatically. The `BusID` defines 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.

Via the `Driver` parameter, specify the driver to use for this 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 directory `/usr/X11R6/lib/modules/drivers`. For this purpose, simply `_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/X11R6/lib/X11/doc`. Generally valid options can also be found in the manual pages (`man XF86Config` and `man XFree86`).

12.1.3 Monitor and Modes Section

Like the `Device` sections, the `Monitor` and `Modes` sections describe one monitor each. The configuration file `/etc/X11/XF86Config` can contain as many `Monitor` sections as desired. The server layout section specifies which `Monitor` section is relevant.

Monitor definitions should only be set by experienced users. The modelines constitute 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.

Caution

Unless you have an in-depth knowledge of monitor and graphics card functions, nothing should be changed in the modelines, as this could cause severe damage to your monitor.

Caution

People daring enough to develop their own monitor descriptions should be very familiar with the documentation in `/usr/X11/lib/X11/doc`. The section covering the video modes deserves a special mention. It describes in detail how the hardware functions and how to create modelines.

The manual specification of the modelines is hardly ever needed nowadays. 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, you can also use one of the VESA modes included in the X server. This will function with practically all graphics card and monitor combinations.

12.2 Installing and Configuring Fonts

The installation of additional fonts in SUSE LINUX is very easy. Simply copy the fonts to any directory that is located in the X11 font path (see Section 12.2.1 on page 283). To enable use of the fonts with the new `xft` font rendering system, the installation directory should be a subdirectory of the directories configured in `/etc/fonts/fonts.conf` (see Section 12.2.1 on the following page).

The font files can be copied manually (as `root`) to a suitable directory, such as `/usr/X11R6/lib/X11/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 configuration of the fonts. To see what exactly this script does, refer to the manual page of the script (`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 in any directory. Only CID-keyed fonts require a slightly different procedure. For this, see Section 12.2.1 on page 284.

12.2.1 Font Systems

XFree 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.

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. Moreover, 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, the two desktop environments KDE and GNOME, Mozilla, and many other applications already use Xft by default. Thus, 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 (`:outline=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:outline=true" family style weight file
```

The output of this command could appear as follows:

```
/usr/X11R6/lib/X11/fonts/truetype/FreeSansBold.ttf: FreeSans:style=Bold:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeMonoBoldOblique.ttf: FreeMono:style=BoldOblique:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeSerif.ttf: FreeSerif:style=Medium:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSerifBoldItalic.ttf: FreeSerif:style=BoldItalic:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeSansOblique.ttf: FreeSans:style=Oblique:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSerifItalic.ttf: FreeSerif:style=Italic:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeMonoOblique.ttf: FreeMono:style=Oblique:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeMono.ttf: FreeMono:style=Medium:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSans.ttf: FreeSans:style=Medium:weight=80
/usr/X11R6/lib/X11/fonts/truetype/FreeSerifBold.ttf: FreeSerif:style=Bold:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeSansBoldOblique.ttf: FreeSans:style=BoldOblique:weight=200
/usr/X11R6/lib/X11/fonts/truetype/FreeMonoBold.ttf: FreeMono:style=Bold:weight=200
```

Important parameters that can be queried with `fc-list`:

Table 12.2: Parameters of fc-list

Parameter	Meaning and Possible Values
`family`	Name of the font family, e.g., `FreeSans`.
`foundry`	The manufacturer of the font, e.g., `urw`.
`style`	The font style, e.g., `Medium`, `Regular`, `Bold`, `Italic`, `Heavy`.
`lang`	The language that the font supports, e.g., `de` for German, `ja` for Japanese, `zh-TW` for traditional Chinese, `zh-CN` for simplified Chinese.
`weight`	The font weight, e.g., `80` for regular, `200` for bold.
`slant`	The slant, usually `0` for none, `100` for italic.
`file`	The name of the file containing the font.
`outline`	`true` for outline fonts, `false` for other fonts.
`scalable`	`true` for scalable fonts, `false` for other fonts.
`bitmap`	`true` for bitmap fonts, `false` for other fonts.
`pixelsize`	Font size in pixels. In connection with fc-list, this option only makes sense for bitmap fonts.

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, and CID-keyed fonts. Unicode fonts have also been supported for quite some time. In 1987, the X11 core font system was originally developed for X11R1 for the purpose of processing monochrome bitmap fonts. All extensions mentioned above were added later.

Scalable fonts are only supported without antialiasing and subpixel rendering and the loading of large scalable fonts with glyphs for many language may take a long time. The use of Unicode fonts may also be slow and requires more memory.

The X11 core font system has a few inherent weaknesses. It is outdated and can no longer be extended in a meaningful fashion. Although it must be retained for reasons of backward compatibility, the more modern Xft/fontconfig system should be used if at all possible.

Remember that only directories meeting the following requirements are taken into account by the X server:

- Directories entered as `FontPath` in the `Files` section in the file `/etc/X11/XF86Config`.
- Directories that have a valid `font.dir` file (generated by SuSEconfig).
- Directories that are not disabled with the command `xset -fp` when the X server is active.
- Directories that are not enabled with the command `xset +fp` when the X server is active.

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`.

As the command `xset` needs access to the running X server, this will only work 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 `sux` and the root password. `sux` 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 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 contain at least the glyphs needed for European languages (formerly encoded as `iso-8859-*`).

CID-Keyed Fonts

In contrast to the other font types, you cannot simply install CID-keyed fonts in just any directory. CID-keyed fonts must be installed in `/usr/share/ghostscript/Resource/CIDFont/`. This is not relevant for Xft/fontconfig, but it is necessary for Ghostscript and the X11 core font system.

Note

See `http://www.xfree86.org/current/fonts.html` for more information about fonts under X11.

Note

12.3 OpenGL — 3D Configuration

In Linux, Direct3D is only available on x86 and compatible systems as part of the Windows emulator WINE, which in turn makes use of the OpenGL interface for the implementation.

12.3.1 Hardware Support

SUSE LINUX includes several OpenGL drivers for 3D hardware support. Table 12.3 provides an overview.

Table 12.3: *Supported 3D Hardware*

OpenGL Driver	Supported Hardware
nVidia	nVidia Chips: all except Riva 128(ZX)
DRI	3Dfx Voodoo Banshee, 3Dfx Voodoo-3/4/5, Intel i810/i815/i830M, Intel 845G/852GM/855GM/865G, Matrox G200/G400/G450/G550, ATI Rage 128(Pro)/Radeon

If you are installing with YaST for the first time, 3D acceleration can be activated during installation, provided YaST detects 3D support. For nVidia graphics chips, the nVidia driver must be installed first. To do this, select the nVidia driver patch in YOU (YaST Online Update). Due to license restrictions, the nVidia driver is not included in the distribution.

If an update is carried out instead of a new installation or a 3Dfx add-on graphics adapter (Voodoo Graphics or Voodoo-2) needs to be set up, the procedure for configuring 3D hardware support is different. This depends on which OpenGL driver is used. Further details are provided in the following section.

12.3.2 OpenGL Drivers

nVidia and DRI

These OpenGL drivers can configured easily with SaX2. For nVidia adapters, the nVidia driver must be installed first (see above). Enter the command `3Ddiag` to check if the configuration for nVidia or DRI is correct.

For security reasons, only users belonging to the group `video` are permitted to access the 3D hardware. Therefore, make sure that all local users are members of this group. Otherwise, the slow *Software Rendering Fallback* of the OpenGL driver will be used for OpenGL applications. Use the command `id` to check whether the current user belongs to the group `video`. If this is not the case, use YaST to add the user to the group.

12.3.3 The Diagnosis Tool 3Ddiag

The diagnosis tool 3Ddiag allows verification of the 3D configuration in SUSE LINUX. This is a command line tool that must be started in a terminal. Enter `3Ddiag -h` to list possible options for 3Ddiag.

To verify the XFree configuration, the tool checks if the packages needed for 3D support are installed and if the correct OpenGL library and GLX extension are used. Follow the instructions of 3Ddiag if you receive "failed" messages. If everything is correct, you will only see "done" messages on the screen.

12.3.4 OpenGL Test Utilities

For testing OpenGL, the program `glxgears` and games like `tuxracer` and `armagetron` (packages have the same names) can be useful. If 3D support has been activated, it should be possible to play these smoothly on a fairly new computer. Without 3D support, these games would run very slowly (slideshow effect). Use the `glxinfo` command to verify that 3D is active, in which case the output contains a line stating `direct rendering: Yes`.

12.3.5 Troubleshooting

If the OpenGL 3D test results are negative (the games cannot be smoothly played), use 3Ddiag to make sure no errors exist in the configuration ("failed" messages). If correcting these does not help or if failed messages have not appeared, take a look at the XFree86 log files.

Often, you will find the line `DRI is disabled` in the XFree86 4.x file `/var/log/XFree86.0.log`. The exact cause can only be discovered by closely examining the log file — a task requiring some experience.

In such cases, no configuration error exists, as this would have already been detected by 3Ddiag. Consequently, at this point, the only choice is to use the software rendering fallback of the DRI driver, which does not provide 3D hardware support. You should also go without 3D support if you get OpenGL representation errors or instability. Use SaX2 to disable 3D support completely.

12.3.6 Installation Support

Apart from the `software rendering fallback` of the DRI driver, all OpenGL drivers in Linux are in developmental phases and are therefore considered experimental. The drivers are included in the distribution because of the high demand for 3D hardware acceleration in Linux. Considering the experimental status of OpenGL drivers, SUSE cannot offer any installation support for configuring 3D hardware acceleration or provide any further assistance with related problems. The basic configuration of the graphical user interface X11 does not include 3D hardware acceleration configuration. If you experience problems with 3D hardware acceleration, it is recommended to disable 3D support completely.

12.3.7 Additional Online Documentation

- DRI: `/usr/X11R6/lib/X11/doc/README.DRI` (`XFree86-doc`)
- Mesa/Glide: `/usr/share/doc/packages/mesa3dfx/` (`mesa3dfx`)
- Mesa general: `/usr/share/doc/packages/mesa/` (`mesa`)

Printer Operation

This chapter provides information on updating from SLES 8 to SUSE LINUX Enterprise Server 9. Additionally, it provides general information on operating printers and helps to find suitable solutions for operating printers in networks.

13.1 Updating, upgrading and migrating the print system 290
13.2 Preparation and Other Considerations 293
13.3 Methods and Protocols for Connecting Printers . . . 295
13.4 Installing the Software 295
13.5 Configuring the Printer 296
13.6 Special Features in SUSE LINUX 299
13.7 Printer Hardware 304

13.1 Updating, upgrading and migrating the print system

In the previous version, SuSE Linux Enterprise Server 8, the two print systems, LPRng/lpdfilter and CUPS, were supplied as equal alternatives. In SUSE LINUX Enterprise Server 9 the focus begins to shift towards CUPS. The option for migrating an LPRng/lpdfilter configuration created with YaST to a CUPS configuration cannot be included in SUSE LINUX Enterprise Server 9. For this reason, before updating from SuSE Linux Enterprise Server 8 to SUSE LINUX Enterprise Server 9, it is necessary to decide whether a migration from LPRng/lpdfilter to CUPS is to be performed under SuSE Linux Enterprise Server 8.

CUPS is the print system of the future. Even if the `lprng` and `lpdfilter` packages continue to be supplied, the changeover to CUPS is recommended in SUSE LINUX Enterprise Server 9 because of the following advantages:

- individual setting of the print parameters on user level
- optimum support for PostScript printers
- “browsing” in the network
- web frontend (on user level and in relation to administration)

After SUSE LINUX Enterprise Server 9, the configuration of LPRng/lpdfilter will no longer be supported by YaST, but must be performed manually from this point onwards. For this reason, a check should be made under SUSE LINUX Enterprise Server 9 to see whether CUPS meets requirements and whether a changeover is possible.

13.1.1 Updating CUPS

When updating CUPS, a distinction should be made between the following pair of cases:

Updating CUPS The software packages are updated, however the existing configuration files will be accepted without change. After the update, the queues and the cupsd will continue to behave as before; this also means that many new features in SUSE LINUX Enterprise Server 9 will not be used at present and will have to be configured at a later stage if necessary.

Upgrading CUPS The existing software packages and the existing configuration files are replaced by the new software packages and their default configuration files. All new features are immediately available, however the queues must be created from scratch. The new features are described in detail in the following articles:

- `http://portal.suse.com/sdb/de/2004/03/jsmeix_print-einrichten-91.html`
- `http://portal.suse.com/sdb/de/2003/09/jsmeix_print-einrichten-90.html`
- `http://portal.suse.com/sdb/de/2003/03/jsmeix_print-einrichten-82.html`
- `http://portal.suse.com/sdb/de/2002/09/jsmeix_print-einrichten-81.html`

13.1.2 Migration from LPRng/lpdfilter to CUPS

Information on migrating from LPRng/lpdfilter to CUPS

CUPS and LPRng/lpdfilter are fundamentally different.

- In the case of CUPS, the configuration data for the queues is stored in `/etc/cups/printers.conf` and `/etc/cups/ppd/` and `/etc/printcap` is created by `cupsd` only for the purpose of compatibility with `printcap`-based application programs.
- CUPS uses the IPP protocol, while LPRng uses the LPD protocol
- CUPS normally needs more computing performance for more powerful filtering and the web interface

Note

LPD functionality with CUPS

CUPS supports fundamental LPD functionality, both on the *recipient side* by means of the `cups-lpd` and on the *sender side* by means of the `lpd` backend. However, CUPS does not support full LPRng functionality. It is also possible to use filter scripts of LPD-based print systems as “System V style interface script”.

Note

The parallel operation of a test system makes it possible to migrate to CUPS in a secure way; the existing LPD print server remains active. SUSE LINUX Enterprise Server 9 is installed with CUPS on an additional system.

The following procedure only works well for network printers; this is because printers that are connected directly to the LPD print server must be connected directly to the CUPS test system for testing purposes. The queues for the network printers are set up on the CUPS test system. It should be noted that many network printers (or their network interfaces) will be overloaded if they receive data from several computers at the same time. For this reason, printouts for the network printers to be tested should be paused on the LPD print server while testing with the CUPS test system.

If the queues on the CUPS test system are created with YaST or the Web frontend of CUPS or another graphical tool, then the settings made are to be logged precisely to enable them to be set up on the productive system at a later point. However, if the queues on the CUPS test system are only created with `lpadmin` commands, it is enough to record the `lpadmin` commands in a script and to allow this script to run on the productive system.

Migration of a YaST configuration to CUPS

The procedure is as follows:

- For queues that have only been configured with YaST:
    1. Migration from LPRng/lpdfilter to CUPS under SuSE Linux Enterprise Server 8.
    2. Update from SuSE Linux Enterprise Server 8 to SUSE LINUX Enterprise Server 9.
- There is not automatic migration for queues that have been configured not just with YaST.
- A change of installed packages from LPRng/lpdfilter to CUPS is available, however an existing configuration is not migrated.

In the case of SuSE Linux Enterprise Server 8, the two print systems, CUPS and LPRng/lpdfilter are always configured simultaneously with the YaST printer configuration. The printer configuration stores all configuration data and thereby creates the configuration for the current print system - or for the new print system if the print system has been changed. The YaST printer configuration differentiates strictly between queues that it has created itself and those created in another way. The latter are not changed and cannot be changed, but just overwritten with a new configuration.

In the case of SUSE LINUX Enterprise Server 9 YaST and other configuration tool (e. g. the CUPS Web Frontend) are synchronized. There is no more “private” YaST configuration data. The migration of an LPRng/lpdfilter configuration created with YaST to a CUPS configuration must therefore be omitted in the case of SUSE LINUX Enterprise Server 9 because “private” YaST configuration data is required for the migration of a YaST configuration. The automatic migration of any LPRng/lpdfilter configuration to a CUPS configuration cannot be achieved with reasonable effort.

13.2 Preparation and Other Considerations

CUPS is the standard print system in SUSE LINUX. CUPS is highly user-oriented. In many cases, it is compatible with LPRng or can be adapted with relatively little effort. LPRng is only included in SUSE LINUX Enterprise Server for reasons of compatibility (see 13.1 on page 290).

Printers can be distinguished in terms of the interfaces (USB, network) and the printer languages. When buying a printer, make sure the printer has an interface that is supported by the hardware and a suitable printer language.

Printers can be roughly categorized into 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.

As 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
Though these printer languages are quite old, they are still undergoing expansion in order to address new features in the printer. 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, as the Open Source developers may still be working on these features. Except for the `hpijs` drivers developed by HP, there are currently (2004) 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 (usually GDI printers)

Usually only one or several Windows drivers are available for proprietary printers. These printers do not support any of the common printer languages, and the printer languages they use are subject to change when a new edition of a model is released.

Meanwhile, the Open Source community has abandoned the policy of supporting such printers via reverse engineering, as the success is very short-lived compared to the effort required. Inexpensive Lexmark printers, which are now also offered under the Dell brand, are a typical example for this kind of printers. These printers are frequently included as give-aways in PC bundles. A set of new cartridges often costs more than the printer itself.

Most of these printers are in the low price range. They are usually not suitable for Linux.

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

- `http://cdb.suse.de/` or `http://hardwaredb.suse.de/` — the SUSE LINUX printer database
- `http://www.linuxprinting.org/` — the printer database on linuxprinting.org
- `http://www.cs.wisc.edu/~ghost/` — the Ghostscript web page
- `file:/usr/share/doc/packages/ghostscript/catalog.devices` — included drivers

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

13.3 Methods and Protocols for Connecting Printers

There are various possibilities for connecting a printer to the system. The user can connect a printer to the USB port or a parallel port of the local host. 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. The respective back-end must also be selected for the communication.

Caution

Cable Connection to the Machine

When connecting the printer to the machine, do not forget that only USB devices can be plugged in or unplugged during operation. The system should be shut down before changing other kinds of connections.

Caution

The cabling of network printers should be installed as described by the printer manufacturer.

13.4 Installing the Software

PPD (PostScript Printer Description) is the computer language that describes the properties (e.g., resolution) and options (e.g., duplex unit) of PostScript printers. These descriptions are necessary to make use of the 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, a lot of PPD files are preinstalled. In this way, even printers that do not have built-in PostScript support can be used.

If a PostScript printer is configured, the best approach is to get a suitable PPD file. If this is too difficult or if no such file exists, the system can also be used with one of the included generic PPD files. Normally, PPD files are available on the driver CDs for Windows or MacOS. If the syntax is correct, these files can also be used in Linux. Some printer manufacturers also offer PPD files on the Internet.

New PPD files can be stored in the directory `/usr/share/cups/model/`. However, the preferred approach is to add them to the print system with YaST (see 2.4.2 on page 67). Subsequently, the PPD file can be selected during the installation.

Be careful if a printer manufacturer wants you to install entire software packages. Firstly, this kind of installation would result in the loss of the support provided by SUSE LINUX, and secondly, print commands may work in a different way than formerly 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.

13.5 Configuring the Printer

After connecting the printer to the computer and installing the software, the printer must be installed in the system. If possible, this should be done with the tools delivered with SUSE LINUX, not with any other tools. As SUSE LINUX puts great emphasis on security, third-party tools often have difficulties with the security restrictions and end up causing more complications than benefit.

13.5.1 Local Printers

If you have a local printer that is not too old, it will be detected as not yet configured when you log in, and a YaST module will be started for configuring it (see 2.4.2 on page 66).

To configure the printer with command-line tools (see below), you need a device URI such as `parallel:/dev/lp0` (printer connected to the first parallel port) or `usb:/dev/usb/lp1` (first detected printer connected to the USB port).

13.5.2 Network Printers

A network printer can support various protocols, some of them even concurrently. Though 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. Thinking that delivering drivers for the current monopolistic operating system is sufficient, the manufacturers believe that they can integrate these extensions without causing any difficulties. Unfortunately, this causes problems in Linux. 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 in order to achieve a functional configuration.

CUPS supports the `socket`, `LPD`, `IPP`, and `smb` protocols. Here is some detailed information on these protocols:

socket "socket" refers to a connection in which the data are 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`. Example for a device URI: `socket://`⟨*host-printer*⟩`:9100/`

LPD (Line Printer Daemon) The proven LPD protocol is described in RFC 1179. Under this protocol, some job-related data such as the printer queue are sent before the actual print data are 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 printer queue. If necessary, the printer manual may indicate which name to use. LPT, LPT1, LP1 or similar names are often used. Of course, 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`. Example for a device URI: `lpd://`⟨*host-printer*⟩`/LPT1`

IPP (Internet Printing Protocol) IPP is a relatively new (1999) protocol that is based on the HTTP protocol. Under the IPP, much more job-related data are transmitted than in the other protocols.

CUPS uses IPP for the internal data transmission. This is the preferred protocol for a forwarding queue between two CUPS servers. The name of the printer queue is necessary in order to configure IPP correctly. The port number for IPP is `631`. Example for a device URI: `ipp://`⟨*`host-printer`*⟩`/ps` oder: `ipp://`⟨*`host-cupsserver`*⟩`/printers/ps`

SMB (Windows share) CUPS also supports printing on printers connected to Windows shares. The protocol used for this purpose is SMB. SMB makes use of the port numbers `137`, `138`, and `139`. Example for a device URI:

```
smb://user:password@workgroup/server/printer
smb://user:password@host/printer
smb://server/printer
```

The protocol supported by the printer must be determined prior to the configuration. If the manufacturer does not provide the needed information, the command `nmap` (`nmap` package) can be used to guess the protocol. `nmap` checks a host for open ports. Example:

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

13.5.3 Configuration Tasks

Configuring Network Printers

Network printers should be configured with YaST. YaST facilitates the configuration and is best equipped to handle the security restrictions in CUPS (see 2.4.2 on page 66).

Configuring with Command-line Tools

Alternatively, CUPS can be configured with command-line tools. If the preparatory work has been done (i.e., if you know the PPD file and the name of the device), the following steps are necessary:

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

Do not use `-E` as the first option. For all CUPS commands, `-E` as the first argument implies the 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
```

Example for a network printer:

```
lpadmin -p ps -v socket://192.168.1.0:9100/ -P \
/usr/share/cups/model/Postscript-level1.ppd.gz -E
```

Modifying Options

YaST allows certain options to be activated by default during the installation. These options can be modified for every print job (depending on the print tool that is used) or specified later on (e.g., with YaST).

Using command-line tools, this can be done as follows:

1. First, list all options:

   ```
   lpoptions -p <queue> -l
   ```

   Example:

   ```
   Resolution/Output Resolution: 150dpi *300dpi 600dpi 1200dpi
   ```

2. The activated default option is evident from the preceding asterisk: *
3. Change the option with `lpadmin`:

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

4. Check the new setting:

   ```
   lpoptions -p <queue> -l
   ```

   ```
   Resolution/Output Resolution: 150dpi 300dpi *600dpi 1200dpi
   ```

13.6 Special Features in SUSE LINUX

13.6.1 Administration with the Web Front-end (CUPS)

In order to be able to use the administration with the web front-end (CUPS) or the printer administration tool (KDE), the user `root` must be set up as CUPS administrator with the CUPS administration group `sys` and a CUPS password. This can be done as `root` with the following command:

```
lppasswd -g sys -a root
```

If this is not done, the administration will not be possible with the web interface or with the administration tool, as the authentication will fail if no CUPS administrator has been configured. Instead of `root`, any other user can also be appointed as CUPS administrator (see the following 13.6.2).

13.6.2 Changes in the CUPS Print Service (cupsd)

- `cupsd` runs as the user `lp`.
- Generalized functionality for `BrowseAllow` and `BrowseDeny`.
- `cupsd` is activated by default.

For more information, see the Support Database article "Printer Configuration from SUSE LINUX 9.0 on" at `http://portal.suse.com/sdb/en/2003/09/jsmeix_print-einrichten-90.html`.

cupsd Runs as the User lp

Upon start-up, `cupsd` changes from the user `root` to the user `lp`.

Advantage: Much higher level of security, as the CUPS print service does not run with unrestricted permissions, but only with the permissions needed for the print service.

Disadvantage: The authentication (more precisely: the password check) cannot be performed via `/etc/shadow`, as `lp` has no access to `/etc/shadow`. Rather, the CUPS-specific authentication via `/etc/cups/passwd.md5` must be used. For this purpose, a CUPS administrator with the CUPS administration group `sys` and a CUPS password must be entered in `/etc/cups/passwd.md5`. To do this, enter the following as `root`:

```
lppasswd -g sys -a <CUPS-admin-name>
```

Other consequences:

- If `cupsd` runs as `lp`, `/etc/printcap` cannot be generated, as `lp` is not permitted to create files in `/etc/`. Therefore, `cupsd` generates `/etc/cups/printcap`. In order to ensure that applications that can only read queue names from `/etc/printcap` continue to work properly, `/etc/printcap` is a symbolic link pointing to `/etc/cups/printcap`.

- As soon as `cupsd` runs as `lp`, port `631` cannot be opened. Therefore, `cupsd` can no longer be reloaded with `rccups reload`. Use `rccups restart` instead.

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 (more precisely: 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 is Activated by Default

In a standard installation, `cupsd` is activated automatically, thus enabling comfortable access to the queues of CUPS network servers without any additional manual actions.

The two first items are vital preconditions for this feature, as otherwise the security would not be sufficient for an automatic activation of `cupsd`.

13.6.3 PPD Files in SUSE Packages

- Printer configuration with PPD files only
- CUPS PPD files in the `cupsd` package

- Foomatic (or LinuxPrinting.org) PPD files in the `cups-drivers` package
- GimpPrint PPD files in the `cups-drivers-stp` package
- PPD files from printer manufacturers in the `manufacturer-PPDs` package

For more information, refer to the Support Database articles "Printer Configuration from SuSE Linux 8.2" at `http://portal.suse.com/sdb/en/2003/03/jsmeix_print-einrichten-82.html`

and "Printer Configuration in SUSE LINUX 9.1" at `http://portal.suse.com/sdb/en/2004/03/jsmeix_print-einrichten-91.html`

Printer Configuration with PPD Files Only

The YaST printer configuration sets up the queues for CUPS using only the PPD files installed in `/usr/share/cups/model/` on the respective system.

To determine the suitable PPD files for the respective printer model, YaST compares the vendor and model determined during the 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, you will receive the PPD files matching the respective 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 generates the vendor and model database anew. For example, if you only have PostScript printers, normally you need neither the Foomatic PPD files in the `cups-drivers` package nor the GimpPrint 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) in order 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 by the following 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`

Foomatic (or LinuxPrinting.org) 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 the following conditions are met:

- A Foomatic PPD file with the entry "*NickName: ... Foomatic ... (recommended)" matches the printer model.
- The `manufacturer-PPDs` package does not contain a more suitable PPD file (see below).

GimpPrint PPD Files in the cups-drivers-stp Package

Instead of "foomatic-rip", the CUPS filter "rastertoprinter" from GimpPrint can be used for many non-PostScript printers. This filter and suitable GimpPrint PPD files are available in the `cups-drivers-stp` package. The GimpPrint 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.

General information on the configuration of PostScript printers:

- PostScript printers should be configured with the suitable PPD file of the printer manufacturer, as 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:
  1. The vendor and model determined during the hardware detection match the vendor and model in a PPD file from the `manufacturer-PPDs` package.

2. 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:

  1. 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 (e.g., 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).
  2. The Foomatic PostScript PPD file is not “recommended”. This may be due to the following reasons:
     - ▷ The printer model does not operate efficiently enough in PostScript mode (e.g., 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).
     - ▷ The printer does not support PostScript by default (e.g., because PostScript support is available as an optional module).

- If a PPD file from the `manufacturer-PPDs` package is suitable for a PostScript printer, but YaST does not use it for the above-mentioned reasons, select the respective printer model manually in YaST.

13.7 Printer Hardware

13.7.1 Printers that Do Not Support a Standard Printer Language

As printer manufacturers often fail to provide Linux drivers for non-PostScript printers, it should be possible to address these printers with one of the common printer languages (*PostScript*, *PCL*, or *ESC/P*). Normal printers understand at least one of the common printer languages. Printers that do not support any common printer language and can only be addressed with special control sequences are called *GDI printers*. These printers only work with the operating system version for which the manufacturer delivers a driver. *GDI* is a programming interface developed by Microsoft for graphics devices. The actual problem is not the programming interface, but the fact that GDI printers can *only* be addressed with the proprietary printer language of the respective printer model.

Some printers can be switched to operate either in GDI mode or one of the standard printer languages.

Some manufacturers provide proprietary drivers for their GDI printers. The disadvantage of proprietary printer drivers is that there is no guarantee that these will 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, a decent new printer may be purchased for a reasonable price. Moreover, 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.

13.7.2 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 the "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.

13.7.3 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 the interrupt `7` is still free, it can be activated with the entry shown in File 13.1.

Example 13.1: */etc/modprobe.conf: Interrupt Mode for the First Parallel Port*

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

Before activating the interrupt mode, check the file `/proc/interrupts` to see which interrupts are already in use. Only the interrupts that are 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`.

13.7.4 Troubleshooting Network Printers

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 you experience are related to the network.

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

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

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

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

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

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

If the `lpd` does not respond, the `lpd` may not be active or there may be basic network problems. If the `lpd` responds, the response should show why printing is not possible on the `queue` on `host`. Examples:

Example 13.2: *Error Message from the* `lpd`

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

If you receive a response of this kind, the problem is caused by the remote `lpd`.

Checking a remote `cupsd` By default, the CUPS network server should broadcast its queue every 30 seconds on the 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 following output should be returned after 40 seconds:

Example 13.3: *Broadcast from the CUPS Network Server*

```
ipp://<host>.<domain>:631/printers/<queue>
```

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

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

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

```
lpstat -h <host> -l -t
```

This command returns a (possibly very long) status report for all queues on ⟨*host*⟩, provided the respective `cupsd` is active and the host accepts queries.

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

This 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.

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. As this is caused by the spooler in the print server box, there is nothing you can do about it. As a workaround, you can circumvent the spooler in the print server box by addressing the printer connected to the print server box directly via TCP socket.

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 respective 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 the following:

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 following command: `nmap -p` ⟨*from_port*⟩-⟨*to_-port*⟩> ⟨*IP-address*⟩ This may take some time. For further information, refer to `man 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 in order to test if the printer can be addressed on this port.

13.7.5 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 (e.g., if the printer is not able to print the printer-specific data), the print system will 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.

13.7.6 Disabled Queues

If the data transfer to the recipient fails entirely (normally a CUPS back-end makes several attempts), the back-end will report an error to the print system (more precisely: to the `cupsd`). The back-end decides whether and how many attempts make sense until the data transfer is reported as impossible. As further attempts would be in vain, `cupsd` will disable printing for the respective queue (`disable`). After eliminating the cause of the problem, the system administrator must reenable printing with the command `/usr/bin/enable`.

13.7.7 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 the print jobs from the applications and forwards them to the `cupsd` on the server. When a `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. As a print job is usually forwarded immediately, it cannot be deleted with the job number on the client host, as the client `cupsd` regards the print job as completed as soon as it has been forwarded to the server `cupsd` (see above). To delete the print job on the server, use a command such as the following to determine the job number on the server, provided the server has not already completed the print job (i.e., sent it to the printer):

```
lpstat -h <print-server> -o
```

Using this job number, the print job on the server can be deleted:

```
cancel -h <print-server>
 <queue>-<jobnumber>
```

13.7.8 Defective Print Jobs and Data Transfer Errors

Print jobs remain in the queues and are printed anew 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 will print loads of paper with unintelligible characters, as it is unable to process the data correctly.

1. To stop printing, remove all paper from inkjet 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, as jobs are only removed after they are sent completely to the printer. Use `lpstat -o` (or `lpstat -h ⟨print-server⟩ -o`) to check which queue is currently printing. Delete the print job with `cancel ⟨queue⟩-⟨jobnumber⟩` (or `cancel -h ⟨print-server⟩ ⟨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 power the printer on.

13.7.9 Troubleshooting the CUPS Print System

1. Set `LogLevel debug` in `/etc/cups/cupsd.conf`.
2. Stop the `cupsd`.
3. Remove `/var/log/cups/error_log*` in order to avoid having to search through very large log files.
4. Start the `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.

14

The Hotplug System

The hotplug system under SUSE LINUX was developed in connection with the *Linux Hotplug project*, but it has a few distinguishing features. The main difference is that, under SUSE LINUX, the scripts `/sbin/hwup` and `/sbin/hwdown` are used instead of the event multiplexer `/etc/hotplug.d` to initialize or stop hotplug devices. Many of the hardware and software characteristics described in this chapter are not the same on IBM S/390 and zSeries machines.

14.1 Devices and Interfaces 314
14.2 Hotplug Events 314
14.3 Hotplug Agents 315
14.4 Automatic Module Loading 316
14.5 Network Devices and Interface Designations 317
14.6 Hotplug with PCI 317
14.7 Coldplug 317
14.8 Error Analysis 318

The hotplug system is not only used for devices that can be inserted and removed during operation, but also for all devices only detected after the kernel has been booted. These devices are entered in the `sysfs` file system, which is mounted under `/sys`. Until the kernel has been booted, only devices that are absolutely necessary, such as bus system, boot disks, or keyboard, are initialized.

The most important hotplug functions are configured in two files. The first of these, `/etc/sysconfig/hotplug`, contains variables that influence the behavior of `hotplug` and `coldplug`. Every variable is explained by a commentary. The second file, `/proc/sys/kernel/hotplug`, contains the name of the executable program called by the kernel.

14.1 Devices and Interfaces

As well as devices, the hotplug system also administers interfaces. A device is either linked to a bus or an interface. An interface links devices to each other or to an application.

Devices entered in the `sysfs` file are found under `/sys/devices`. Interfaces are located under `/sys/class` or `/sys/block`. All interfaces should have a link to the related device in the `sysfs` file, but there are always some drivers that do not automatically add this link.

Typical examples of devices and interfaces include the following:

PCI Network Card A device linked to the PCI bus (`/sys/devices/pci0000:00/0000:00:1e.0/0000:02:00.0`) that has a network interface used by network services or linked to a virtual network device, such as a tunnel or VLAN. (`/sys/class/net/eth0`)

PCI SCSI Controller A device (`/sys/devices/pci0000:20/0000:20:01.1`) that makes several physical interfaces available in the form of a bus (`/sys/class/scsi_host/host1`).

SCSI Hard Disk A device (`/sys/devices/pci0000:20/0000:20:01.1/host1/1:0:0:0`) with several interfaces (`/sys/block/sda*`).

14.2 Hotplug Events

Every device and every interface has an associated *hotplug event*, which is processed by the relevant hotplug agent. Hotplug events are triggered either by the kernel when a link is established to a device or by *coldplug*, which checks the main buses at regular intervals and creates an event for all devices that have not been initialized. Further information on *coldplug* is provided in Section 14.7 on page 317. The kernel also initiates a hotplug event as soon as a driver registers an interface.

A hotplug event is a call to a hotplug user-mode tool, such as `/sbin/hotplug`, which is specified in file `/proc/sys/kernel/hotplug`. `/sbin/hotplug` searches for a hotplug agent that matches the type of event. If there is no suitable agent but there is a `dev` file in the device path, the agent generic_udev.agent is called.

Note

To ensure that events of a particular kind are ignored, edit file `/etc/sysconfig/hotplug` and set the desired event to `HOTPLUG_SKIP_EVENTS`.

Note

14.3 Hotplug Agents

A number of different hotplug events have been added to kernel 2.6. Every new driver can introduce a new event. Only events already known have agents assigned to them. These agents carry out the necessary actions.

The majority of device agents load kernel modules, but occasionally they also call additional commands. For example, with some computer architectures, such as IBM S390, a particular value must be entered for every device in `procfs` or `sysfs`, for that device to be initialized. Under SUSE LINUX, this is handled by `/sbin/hwup` or `/sbin/hwdown`. These programs search for a configuration suitable for the device in `/etc/sysconfig/hardware`. If `/sbin/hwup` does not find any configuration, modules are automatically loaded. For further information about this point, see Section 14.4 on the following page. Further information about `/sbin/hwup` is contained in file `/usr/share/doc/packages/sysconfig/README` and in `man hwup`.

Interface agents perform two main tasks. First, they initialize the interface or call `udev` to create a device node. Second, network interfaces are initialized with `/sbin/ifup` and deactivated with `/sbin/ifdown`. Further details about this subject can be found in the file `/usr/share/doc/packages/sysconfig/README` and with `man ifup`. There is also a man page for `udev` (`man udev`). Another source of information is Section 15 on page 319.

14.4 Automatic Module Loading

If it has not been possible to initialize a device with `/sbin/hwup`, the agent searches through *module maps* for a suitable driver. The first place it looks is the maps contained in `/etc/hotplug/*.handmap`. If it does not find the driver there, it also searches in `/lib/modules/<kernelversion>/modules.*map`. To use a driver other than the standard driver for the kernel, enter this in `/etc/hotplug/*.handmap`, as this is the first file read.

Note the following differences between USB and PCI. The agent `usb.agent` also searches for user-mode drivers in files `/etc/hotplug/usb.usermap` and `/etc/hotplug/usb/*.usermap`. In this way it is possible to call executable programs for particular devices.

In the case of PCI devices, `pci.agent` first queries `hwinfo` about driver modules. Only if `hwinfo` does not know of any drivers does the agent look in `pci.handmap` and the kernel map. As `hwinfo` has already looked there, the inquiry must fail. `hwinfo` has an additional database available to it, which is read after the handmap and before the kernel map.

The agent `pci.agent` can be confined to driver modules in a particular subdirectory of `/lib/modules/<kernelversion>/kernel/drivers`. To specify this directory or several directories, enter the variable `HOTPLUG_PCI_DRIVERTYPE_WHITELIST` in file `/etc/sysconfig/hotplug`. Exclude directories by specifying them in `HOTPLUG_PCI_DRIVERTYPE_BLACKLIST`. The modules in these directories are never loaded. Additionally, enter modules no agent is allowed to load in the file `/etc/hotplug/blacklist`. Write each module name in a separate line.

If several suitable modules are found in a map file, only the first module is loaded. To load all the modules, set `HOTPLUG_LOAD_MULTIPLE_MODULES=yes`.

Note

Modules loaded with `hwup` are not affected by this. Automatic module loading occurs only in exceptional cases.

Note

14.5 Network Devices and Interface Designations

If a computer has more than one network device with different drivers, it is possible for the interface designations to change after the boot process has completed if another driver has been loaded more quickly. For this reason, network devices in SUSE LINUX are administered via a queue. Alter this behavior by setting `HOTPLUG_PCI_QUEUE_NIC_EVENTS=no` in `/etc/sysconfig/hotplug`.

However, the best way to ensure consistent interface designations is to specify the desired name in the configuration files of the individual interfaces. Information about how to do this is contained in file `/usr/share/doc/packages/sysconfig/README`.

14.6 Hotplug with PCI

Some computers also allow hotplug for PCI devices. To make full use of this, it is necessary for special kernel modules to be loaded. However, on non-PCI hotplug computers, these modules could cause problems. Unfortunately, hotplug PCI slots cannot be automatically detected, so you have to configure this function manually. To do this, set variable `HOTPLUG_DO_REAL_PCI_HOTPLUG` in file `/etc/sysconfig/hotplug`.

14.7 Coldplug

Coldplug is responsible for all devices connected before the hotplug system is enabled during the boot process. It also takes care of devices that are not easy to detect.

First, the script `rccoldplug` calls the command `hwup` for every static hardware configuration `/etc/sysconfig/hardware/hwcfg-static-*`. Then the scripts `/etc/hotplug/*.rc` search for devices not yet initialized and create hotplug events. For PCI devices there is both a positive and a negative list of device types which should initialize or skip `coldplug`.

Detailed comments for this are contained in the file `/etc/sysconfig/hotplug`.

The scan scripts output one character on the screen for every device that has been checked, as follows:

.	Device is already initialized and will be skipped.
*****	Hotplug event will be created for the device.
W	Device is not on the whitelist and will be skipped.
B	Device is on the blacklist and will be skipped.

14.8 Error Analysis

14.8.1 Log Files

Unless otherwise specified, `hotplug` only sends a few important messages to `syslog`. To obtain more information, set `HOTPLUG_DEBUG=yes`. If you set this variable to the value `max`, every shell command is logged for all hotplug scripts. This means that `/var/log/messages` in which `syslog` stores all the messages will be much larger. As `syslog` is not launched during the boot process until after `hotplug` and `coldplug`, it is possible, however, for the first messages not to be logged. If these messages are important to you, specify a different log file via the variable `HOTPLUG_SYSLOG`. Information about this topic is contained in `/etc/sysconfig/hotplug`.

14.8.2 Boot Problems

If a computer hangs during the boot process, disable `hotplug` or `coldplug` by entering `NOHOTPLUG=yes` or `NOCOLDPLUG=yes`, respectively, at the boot prompt. When the system is up and running, reenable hotplug by entering the command `rchotplug start`.

To find out whether a particular module loaded by `hotplug` is responsible for the problem, enter `HOTPLUG_TRACE=<N>` at the boot prompt. The names of all the modules are then output one after another on the screen until after N seconds they are actually loaded. You cannot intervene while this is going on.

14.8.3 The Event Recorder

The script `/sbin/hotplugeventrecorder` is called for each event by `/sbin/hotplug` and `sbin/hotplug-stopped`. If a directory `/events` exists, all hotplug events are stored as individual files in this directory.

15

Dynamic Device Nodes With udev

Linux Kernel 2.6 introduces a new *user space* solution for a dynamic device directory `/dev` with consistent device designations: `udev`. The previous implementation of `/dev` with `devfs` no longer works and has been replaced by `udev`.

15.1 Background Information 320
15.2 Basics On How To Create Rules 320
15.3 Automatisms With NAME And SYMLINK 321
15.4 Regular Expressions In Keys 321
15.5 Tips On The Selection Of Suitable Keys 322
15.6 Consistent Names For Mass Storage Devices 323

15.1 Background Information

Traditionally, device nodes were stored in the `/dev` directory on Linux systems. For every possible type of device there was a node, irrespective of whether it actually existed in the system. The result was that this directory took up a lot of space. The command `devfs` has brought a significant improvement, as now only devices that really exist are given a device node in `/dev`.

`udev` introduces a new way of creating device nodes. It compares the information made available by `sysfs` with data provided by the user in the form of rules. `sysfs` is a new file system in Kernel 2.6 and provides the basic information on connected devices in the system. It is mounted under `/sys`.

It is not absolutely necessary for the user to create rules. If a device is connected, then the appropriate device node is created. However, the rules introduce the possibility of changing the names for the nodes. This offers the convenience of replacing a cryptic device name with a name that is easy to remember and also of having consistent device names where two devices of the same type have been connected.

Unless otherwise specified, two printers are given the designations `/dev/lp0` and `/dev/lp1`. Of course, which device is given which device node depends on the order in which they are switched on. Another example is external mass storage devices, such as USB hard disks. The `udev` command allows exact device paths to be entered in `/etc/fstab`.

15.2 Basics On How To Create Rules

Before `udev` creates device nodes under `/dev`, it reads file `/etc/udev/udev.rules`. The first rule which fits a device is used, even if other rules would also apply. Comments are introduced with a hash sign (#). Rules take the following form:

```
key, [key,...] NAME [, SYMLINK]
```

At least one key has to be specified, as rules are assigned to devices on the basis of these keys. It is also essential to specify a name, as the device node that is created in `/dev` bears this name. The optional symlink parameter allows nodes to be created in other places. A rule for a printer could thus take the following form:

```
BUS="usb", SYSFS{serial}="12345",
NAME="lp_hp", SYMLINK="printers/hp"
```

In this example there are two keys, `BUS` and `SYSFS{serial}`. `udev` becomes the serial number with which the serial number of the device that is connected to the USB bus is compared. To assign the name `lp_hp` to the device in the `/dev` directory, all the keys must agree. In addition, a symbolic `/dev/printers/hp`, which refers to the device node, is created. During this operation, the `printers` directory is automatically created. Print jobs can then be sent to `/dev/printers/hp` or `/dev/lp_hp`.

15.3 Automatisms With NAME And SYMLINK

The parameters NAME and SYMLINK allow operators to be used. These operators refer to kernel data on the corresponding device. A simple example will illustrate the procedure:

```
BUS="usb", SYSFS{vendor}="abc",
SYSFS{model}="xyz", NAME="camera%n"
```

The operator %n in the name is replaced by the number of the camera device: for example,`camera0`, `camera1`. Another useful operator is `%k`, which is replaced by the standard device name of the kernel, for example, `hda1`. You will find a list of all the operators in the man page for `udev`.

15.4 Regular Expressions In Keys

Regular expressions, such as wildcards, can be used in the shell. Thus, for example, the character `*` can be used as a placeholder for any characters or `?` can be used for precisely one character.

```
KERNEL="ts*", NAME="input/%k"
```

This rule assigns the standard kernel name in the standard directory to a device whose designation begins with the letters "ts". Detailed information on the use of regular expressions in udev rules can be found on man page `man udev`.

15.5 Tips On The Selection Of Suitable Keys

It is essential to choose a good key for every functioning udev rule. Here are some examples of standard keys:

BUS device bus type

KERNEL device name which the kernel uses

ID device number on the bus (for example, PCI bus ID)

PLACE physical point where the device is connected (for example, on USB)

The keys `ID` and `PLACE` can be useful, but usually the keys `BUS`, `KERNEL` and `SYSFS{...}` are used. The `udev` command also provides keys which call external scripts and evaluate their results. Further details on this can be found on man page `man udev`.

The command `sysfs` stores small files with hardware information in a directory tree. Each file generally is only given one item of information, such as the device name, the vendor or the serial number. Each of these files can be used as the value of a key. If you want to use several SYSFS... keys in one rule, however, you can only use files in the same directory.

udevinfo can be a useful tool here. You just have to find one subdirectory of `/sys` which refers to the relevant device and is given a file `dev`. These directories are all located under `/sys/block` or `/sys/class`.

If a device node already exists for the device, once again `udevinfo` can reduce the amount of work you have to do. The command `udevinfo -q path -n /dev/sda` outputs `/block/sda`. This means that the directory you are looking for is `/sys/block/sda`. Now call `udevinfo` with the following command: `udevinfo -a -p /sys/block/sda`. The two commands can also be combined, for example: `udevinfo -a -p ‘udevinfo -q path -n /dev/sda‘`. The following is an extract from the output:

```
BUS="scsi"
ID="0:0:0:0"
SYSFS{detach_state}="0"
SYSFS{type}="0"
SYSFS{max_sectors}="240"
SYSFS{device_blocked}="0"
```

```
SYSFS{queue_depth}="1"
SYSFS{scsi_level}="3"
SYSFS{vendor}="        "
SYSFS{model}="USB 2.0M DSC    "
SYSFS{rev}="1.00"
SYSFS{online}="1"
```

From all the output wealth of information, you should now look for suitable keys that will not change. Remember that you cannot normally use keys from different directories.

15.6 Consistent Names For Mass Storage Devices

SUSE LINUX comes with scripts that help you to always assign the same designations to hard disks and other storage devices. `/sbin/udev.get_persistent_device_name.sh` is a wrapper script. First it calls `/sbin/udev.get_unique_hardware_path.sh`, which ascertains the hardware path for a specified device. `/sbin/udev.get_unique_drive_id.sh` also retrieves the serial number. Both outputs are then passed to `udev`, which creates the symbolic link to the device node under `/dev`. The wrapper can be used directly in the udev rules. Here is an example for SCSI, which can also be generalized to USB or IDE (make sure you write it as one line):

```
BUS="scsi",
PROGRAM="/sbin/udev.get_persistent_device_name.sh",
NAME="%k", SYMLINK="%c{1+}"
```

As soon as a driver has been loaded for a mass storage device, it registers with all the available hard disks with the kernel. Each of them will trigger a hotplug block event which calls `udev`. First `udev` reads the rules to ascertain whether a symlink needs to be created.

If the driver is loaded via `initrd`, then the hotplug events are lost. However, all the information is stored in `sysfs`. The `udevstart` utility finds all the device files under `/sys/block` and `/sys/class` and starts `udev`.

There is also a start script `boot.udev`, which recreates all the device nodes during the boot process. However, the start script has to be activated through the YaST runlevel editor or with the command `insserv boot.udev`.

Note

There are a number of tools and programs which rely on the fact that `/dev/sda` is a SCSI hard disk and `/dev/hda` is an IDE disk. If this is not the case, then these programs will not work. However, YaST relies on these tools and therefore only works with the kernel device designations.

Note

Linux on Mobile Devices

This chapter focuses on the use of Linux on mobile devices — especially on laptops. It covers the configuration of PC cards (PCMCIA), the management of multiple system profiles with SCPM, and wireless communication with IrDA and Bluetooth.

► **s390, zseries**
The hardware specifications described in this chapter do not exist on IBM S/390 and zSeries, making this chapter irrelevant for these platforms. ◄

16.1 PCMCIA 326
16.2 SCPM — System Configuration Profile Management 336
16.3 IrDA — Infrared Data Association 343
16.4 Bluetooth — Wireless Connections 346

16.1 PCMCIA

PCMCIA stands for *Personal Computer Memory Card International Association*. It is used as a collective term for all hardware and software involved.

16.1.1 The Hardware

The essential component is the PCMCIA card. There are two distinct types:

PC cards These are currently the most used cards. They use a 16-bit bus for data transmission. These cards are inexpensive and generally very well supported by Linux.

CardBus Cards These cards represent a more recent standard. CardBus cards use a 32-bit bus, which makes them faster, but also more expensive. Since the data transfer rate is frequently restricted at some other point, it is often not worth the extra cost. There are numerous drivers for these cards, some of them still being unstable. Whether these cards are well supported also depends on the available PCMCIA controller.

What card is currently inserted can be determined with `cardctl ident` once the PCMCIA service is active. A list of supported cards can be found in `/usr/share/doc/packages/pcmcia/SUPPORTED.CARDS` The most recent version of the PCMCIA HOWTO is available in the same directory.

The second essential component consists of the PCMCIA controller of the PC card or CardBus bridge. These establish the connection between the card and the PCI bus and, in older devices, the connection to the ISA bus as well. These controllers are almost always compatible with the Intel chip i82365. All common models are supported. Retrieve the controller type with `pcic_probe`. If it is a PCI device, `lspci -vt` provides additional information.

16.1.2 The Software

Differences between PCMCIA Systems

There are currently two PCMCIA systems — external PCMCIA and kernel PCMCIA. The external PCMCIA system by David Hinds is the older one. It is quite well tested and is subject to ongoing development. The sources of the modules used are not integrated in the kernel sources, which is why it is designated as *external*.

Starting with kernel 2.4, a set of alternative modules is contained in the kernel sources forming the *kernel* PCMCIA system. The basic modules were written by Linus Torvalds. Their support of more recent CardBus bridges is better than that of external PCMCIA.

Unfortunately, the two systems are not compatible. They contain different sets of card drivers. Depending on the hardware involved, only one of the systems may be suitable. The default in SUSE LINUX is the more recent kernel PCMCIA. It is possible to change the system, however. To do this, give the variable `PCMCIA_SYSTEM` in the file `/etc/sysconfig/pcmcia/` either the value `external` or `kernel`. Then PCMCIA must be restarted with `rcpcmcia restart`. If intending to do just a temporary switch between systems, use `rcpcmcia restart external` or `rcpcmcia restart kernel`. If PCMCIA is not running, use the option `start` instead of `restart` to switch the PCMCIA system temporarily. Refer to `/usr/share/doc/packages/pcmcia/README.SuSE` for detailed information.

The Base Modules

The kernel modules for both systems are located in the kernel packages. In addition, the packages `pcmcia` and `hotplug` are required. When PCMCIA is started, the modules `pcmcia_core`, `i82365` (external PCMCIA) or `yenta_socket` (kernel PCMCIA), and `ds` are loaded. In some very rare cases, the module `tcic` is required instead of `i82365` or `yenta_socket`. They initialize the existing PCMCIA controller and provide basic functionality.

The Card Manager

As it is possible to change PCMCIA cards while the system is running, a deamon monitors any activity in the PCMCIA slots. Depending on the chosen PCMCIA system and hardware, this task is performed by the card manager or the hotplug system of the kernel. With external PCMCIA, only the card manager is used. For kernel PCMCIA, the card manager only handles PC Card cards. CardBus cards are handled by hotplug. The card manager is started by the PCMCIA start script after the base modules have been loaded. Because hotplug manages subsystems other than PCMCIA, it has its own start script.

If a card is inserted, card manager or hotplug determines the type and function of the card then loads the corresponding modules. If this is successful, card manager or hotplug starts certain initialization scripts. Depending on the function of the card, they establish a network connection, mount partitions from external SCSI hard drives, or carry out other hardware-specific actions. The scripts for the card manager are located in `/etc/pcmcia/`. The scripts for hotplug can be found in `/etc/hotplug/`.

If the card is removed, card manager or hotplug terminates all card activities using the same scripts. Finally, the modules that are no longer required are unloaded.

Both the start process of PCMCIA and card events are recorded in the system log (`/var/log/messages`). It records which PCMCIA system is currently used and which daemons have been used by which scripts to set up things. Removing a PCMCIA device should work smoothly, at least in theory. This works very well for network, modem, or ISDN cards as long as there are no active network connections. It does, however, fail if mounted partitions of an external hard drive or NFS directories are used. In such cases, ensure that these units are synchronized and cleanly unmounted. This is no longer possible if the card has already been removed. In case of doubt, `cardctl eject` can be helpful to safely eject the card.

This command deactivates all cards still inserted in the laptop. To deactivate only one card, specify the slot number, for example: `cardctl eject 0`.

16.1.3 Configuration

Set whether PCMCIA or hotplug is started at boot time with the YaST runlevel editor or on the command line using `chkconfig`. In `/etc/sysconfig/pcmcia`, there are four variables:

`PCMCIA_SYSTEM` Specifies the PCMCIA system to use.

`PCMCIA_PCIC` Contains the name of the module that addresses the PCMCIA controller. Normally, the start script should detect the module automatically. If this automatic detection fails, enter the name of the desired module here. Otherwise, this variable should be left empty.

`PCMCIA_CORE_OPTS` This was originally designed to contain parameters for the `pcmcia_core` module, which are rarely used. Refer to the manual page of `pcmcia_core` for more information on these options.

`PCMCIA_PCIC_OPTS` Parameters for the module `i82365`. Refer to the manual page of `i82365`. If `yenta_socket` is used, these options are ignored, because `yenta_socket` has no options.

Card manager refers to the files `/etc/pcmcia/config` and `/etc/pcmcia/*.conf` for the assignment of drivers to PCMCIA cards. First, `config` is read then the `*.conf` files in alphabetical order. The last entry found for a card is used. Refer to the manual page of `pcmcia` for details on the syntax of these files.

Network Cards (Ethernet, Wireless LAN, and Token Ring)

These can be set up with YaST like normal network cards. Select 'PCMCIA' as the card type. All other details about setting up the network can be found in Section 21.4 on page 437. Read the notes there about hotpluggable cards.

ISDN

Even for ISDN PC cards, configuration is done to a large extent using YaST, as with other ISDN cards. It is not important which PCMCIA card offered there is chosen, but only that it is a PCMCIA card. When setting up hardware and provider, make sure the operating mode is set to `hotplug` and not to `onboot`.

ISDN modems also exist for PCMCIA cards. These are modem cards or multifunction cards with an additional ISDN connection kit. They are treated like an ordinary modem.

Modem

For modem PC cards, there are normally no PCMCIA-specific settings. As soon as a modem is inserted, it is available under `/dev/modem`.

There are also "soft modems" for PCMCIA cards. As a rule, these are not supported. If there is a driver, it must be individually integrated into the system.

SCSI and IDE

The corresponding driver module is loaded by the card manager or hotplug. When a SCSI or IDE card is inserted, the devices connected to it are available. The device names are detected dynamically. Information about existing SCSI or IDE devices can be found in `/proc/scsi/` or `/proc/ide/`.

External hard drives, CD-ROM drives, and similar devices must be switched on before the PCMCIA card is inserted into the slot. Use active termination for SCSI devices.

Caution

Removing IDE or SCSI Cards

If you intend to remove a SCSI or IDE card, properly unmount all partions on these devices. Otherwise you would not be able to access these devices after a reboot of the system.

Caution

You can also install Linux entirely on external hard drives. However, the boot process is a bit more complicated. You will always need a boot disk containing the kernel and an initial ramdisk (initrd). More information about this can be found in Section 10.3 on page 245.

The initrd contains a virtual file system that includes all required PCMCIA modules and programs. The boot disk (or rather the boot disk image) is designed in a similar fashion. Using these, you could always boot your external installation. It is, however, tiresome to load the PCMCIA support every time by hand. Advanced Linux users can create a customized boot disk for their own system. For more information on this topic refer to the PCMCIA HOWTO, section *Booting from a PCMCIA Device*.

16.1.4 Troubleshooting

Most problems arising with certain laptops or cards using PCMCIA can be solved with little trouble provided you approach the problem systematically.

Caution

Loading Kernel Modules by Hand

Kernel and external PCMCIA cannot be used at the same time, but they exist in parallel in SUSE LINUX. Keep this in mind when loading kernel modules by hand. The modules names of both PCMCIA systems are the same, but they are located in different subdirectories under `/lib/modules/<kernelversion>`. The subdirectories are called `pcmcia/` for kernel PCMCIA and `pcmcia-external/` for external PCMCIA. Thus, the subdirectory must be specified when loading modules manually:

```
modprobe -t <directory> <modulename>
```

Caution

First, find out if the problem is with the card or with the PCMCIA base system. For this reason, always start the computer first without the card inserted. Only insert the card when the base system appears to function correctly. Use `tail -f /var/log/messages` to monitor the system log while searching for the cause of the PCMCIA failure. With this approach, the problem is narrowed down to one of the two following cases.

Nonfunctional PCMCIA Base System

If the system hangs at boot time showing the message "PCMCIA: Starting services" or other strange things happen, PCMCIA can be prevented from being started at the next system boot by entering `NOPCMCIA=yes` at the boot prompt. To further isolate the error, load the three base modules of the PCMCIA system with the following commands by hand (as user `root`):

```
modprobe -t <dir> pcmcia_core
modprobe -t pcmcia-external i82365
```

for external PCMCIA or

```
modprobe -t pcmcia yenta_socket
```

for kernel PCMCIA

(in very rare cases, `modprobe -t <dir> tcic`)

and

```
modprobe -t <dir> ds
```

The critical modules are the first two.

If the error occurs while `pcmcia_core` is loaded, refer to the manual pages for `pcmcia_core` for further information. Use the options described there for a first testing with `modprobe`. As an example, switch off the APM support for the PCMCIA module. In a few cases, there could be problems with this. Use the setting `do_apm=0` to deactivate power management:

```
modprobe -t <dir> pcmciacore do_apm=0
```

If the chosen option is successful, write it to the variable `PCMCIA_CORE_OPTS` in `/etc/sysconfig/pcmcia` to use it permanently:

```
PCMCIA_CORE_OPTS="do_apm=0"
```

Checking free I/O areas may lead to problems if other hardware components are disturbed by this. Avoid this by using `probe_io=0`.

If several options should be used, separate them by spaces:

```
PCMCIA_CORE_OPTS="do_apm=0 probe_io=0"
```

If errors occur while loading the `i82365` module, refer to the manual page of `i82365`.

A problem in this context is a resource conflict — if an interrupt, I/O port, or memory area is occupied twice. Although the module `i82365` checks these resources before they are made available to a card, sometimes just this check leads to problems. Checking the interrupt 12 (PS/2 devices) on some computers leads to the mouse or keyboard hanging. In this case, the parameter `irq_list=<List of IRQs>` can help. The list should contain all IRQs to use. For example, enter the command `modprobe i82365 irq_list=5,7,9,10` or permanently add the list of IRQs to `/etc/sysconfig/pcmcia`:

```
PCMCIA_PCIC_OPTS="irq_list=5,7,9,10"
```

In addition, there are `/etc/pcmcia/config` and `/etc/pcmcia/config.opts`. These files are evaluated by card manager. The settings made in them are only relevant when loading the driver modules for the PCMCIA cards. In `/etc/pcmcia/config.opts`, IRQs, I/O ports, and memory areas can be included or excluded. The difference from the option `irqlist` is that the resources excluded in `config.opts` are not used for a PCMCIA card, but are still checked by the base module `i82365`.

Improperly Functioning or Nonfunctional PCMCIA Card

Here, there are basically three variations: the card is not detected, the driver cannot be loaded, or the interface made available by the driver is set up incorrectly. Determine whether the card is managed by the card manager or hotplug. For external PCMCIA, card manager always takes control. For kernel PCMCIA, card manager manages PC card cards and hotplug manages CardBUS cards. Here, only the card manager is discussed.

Unrecognized Card The message "Unsupported Card in Slot x" in `/var/log/messages` indicates that card manager has failed to assign a driver to the card. The card and driver assignment is done by checking the files `/etc/pcmcia/config` or `/etc/pcmcia/*.conf`. They function as the driver database. This driver database can be easily extended using existing entries as reference. Use `cardctl ident` to find out how the card identifies itself. Refer to the PCMCIA HOWTO (Section 6) and the manual page of `pcmcia` for further details on this procedure. After modifying `/etc/pcmcia/config` or `/etc/pcmcia/*.conf`, reload the driver assignment with the command `rcpcmcia reload`.

Driver Not Loaded Wrong assignments of cards and drivers in the driver database may result in a driver not being loaded. This may happen if a vendor uses a different chip in an apparently unchanged card. Alternative drivers may also offer better support for a particular card than the default assignment. In these cases, precise information on the card is required. If needed, obtain further help from the Advanced Support Service or by asking on a mailing list.

A resource conflict may be another reason for a driver not being loaded. For most PCMCIA cards, it is irrelevant with which IRQ, I/O port, or memory area they are operated, but there are exceptions. First test only one card and, if necessary, switch off other system components, such as the sound card, IrDA, modem, or printer. The allocation of system resources can be monitored with the command `lsdev` (it is quite normal that several PCI devices share the same IRQ).

One possible solution would be to use a suitable option for the module `i82365` (see `PCMCIA_PCIC_OPTS`). Many card driver modules also have options. Find these using the command `modinfo /lib/modules/<pcmciaădirectory>/<driver>.o` (the complete path is needed to locate the correct driver). Most of the modules ship with a manual page. `rpm -ql pcmcia | grep man` lists all manual pages contained in the `pcmcia` package. To test the options, the card drivers can also be unloaded manually. Again, ensure that the module is using the correct PCMCIA system.

When a solution has been found, the use of a specific resource can, be allowed or forbidden in the file `/etc/pcmcia/config.opts`. You may even specify option for card drivers. If, for example, the module `pcnet_cs` should be exclusively operated with IRQ 5, the following entry is required:

```
module pcnet_cs opts irq_list=5
```

One problem that sometimes occurs with 10/100-Mbit network cards is incorrect automatic identification of the transmission method. Use the command `ifport` or `mii_tool` to view and modify the transmission method. To have these commands run automatically, the script `/etc/pcmcia/network` must be adjusted.

Incorrectly Configured Interface In this case, it is recommended to check the configuration of the interface to eliminate configuration errors. For network cards, the verbosity of the network scripts can be increased by assigning the value `DEBUG=yes` to the variable in `/etc/sysconfig/network/config`. For other cards or if this is of no help, there is still the possibility of inserting the line `set -x` into the script run by card manager (see `/var/log/messages`). With this, each individual command of the script is recorded in the system log. If you have found the critical part in a script, the corresponding commands can be entered in a terminal and tested.

16.1.5 Installation with PCMCIA

PCMCIA is already required for installation if you want to install over a network or if the CD-ROM relies on PCMCIA. To do this, start with a boot floppy. In addition, one of the module floppy disks is required.

After booting from floppy disk (or after selecting 'Manual Installation' booting from CD), the program `linuxrc` is started. Select 'Kernel Modules (Hardware Drivers)' → 'Load PCMCIA Module'. Two entry fields appear in which to enter options for the modules `pcmcia_core` and `i82365`. Normally, these fields can be left blank. The manual pages for `pcmcia_core` and `i82365` are available as text files on the first CD in the directory `docu/`.

SUSE LINUX is then installed with the external PCMCIA system. During the installation, system messages are sent to various virtual consoles.

Switch to them using Alt + function key. Later, when a graphical interface is active, use Ctrl + Alt + function key.

During installation, several terminals are available on which commands can be run. As long as linuxrc is running, use console 9 (a very spartan shell). After YaST starts, a Bash shell and many standard system tools are available on console 2.

If the wrong driver module for a PCMCIA card is loaded during installation, the boot floppy must be modified manually. This requires a detailed knowledge of Linux, however. When the first part of the installation is finished, the system is partially or completely rebooted. In rare cases, it is possible that the system will hang when PCMCIA is started. At this point the installation has reached an advanced stage. You can then start Linux in text mode without PCMCIA using the `NOPCMCIA=yes` boot option. See also Section 16.1.4 on page 330. You may even change some system settings on the second console before the first part of the installation is completed to make sure the reboot will be successful.

16.1.6 Other Utilities

cardctl is an essential tool for obtaining information from PCMCIA and carrying out certain actions. In cardctl, find many details. Enter just `cardctl` to obtain a list of the valid commands.

The main functions can be controlled with the graphical front-end cardinfo. For this to work, the `pcmcia-cardinfo` package must be installed.

Additional helpful programs from the `pcmcia` package are `ifport`, `ifuser`, `probe`, and `rcpcmcia`. These are not always required. To find out about everything contained in the `pcmcia`, use the command `rpm -ql pcmcia`.

16.1.7 Updating the Kernel or PCMCIA Package

If you want to update the kernel, you should use the kernel packages provided by SUSE LINUX. If it is necessary to compile your own kernel, the PCMCIA modules must also be recompiled. It is important that the new kernel is already running when these modules are recompiled, because various information is extracted from it. The `pcmcia` package should already be installed, but not started. In case of doubt, run the command `rcpcmcia stop`. Install the PCMCIA source package and enter `rpm -ba /usr/src/packages/SPECS/pcmcia.spec`

The new packages will be stored in `/usr/src/packages/RPMS/`. The package `pcmcia-modules` contains the PCMCIA modules for external PCMCIA. This package must be installed with the command `rpm --force`, because the module files belong officially to the kernel package.

16.1.8 For More Information

For more information about specific laptops, visit the Linux Laptop home page at `http://linux-laptop.net`. Another good source of information is the Mobilix home page at `http://tuxmobil.org/`. The SUSE LINUX Support Database features several articles on the use of SUSE LINUX on mobile devices. Go to `http://portal.suse.de/sdb/en/index.html` and search for *laptop*.

16.2 SCPM — System Configuration Profile Management

Some situations require a modified system configuration of your computer. This would mostly be the case for mobile computers that are operated in varying locations. If a desktop system should be operated temporarily using other hardware components than usual, SCPM comes in handy. At any rate, restoring the original system configuration should be easy and the modification of the system configuration can be reproduced.

Up to the present, this problem had only been solved for PCMCIA hardware for which various configurations could be stored in distinct profiles. This approach has been further refined with the development of SCPM (*System Configuration Profile Management*), obsoleting the restriction to PCMCIA hardware. With SCPM, any desired part of the system configuration can be kept in a customized profile. This is like taking a snapshot of the system then being able to restore it at any point in time.

SCPM's main field of application is network configuration on laptops. Different network configurations often require different settings of other services, such as e-mail or proxies. Then other elements follow, like different printers at home and at the office, a separate X server configuration for the video beamer at conferences, special power-saving settings for the road, or the differing time zone in the agency abroad.

Increasing use of this tool leads to the continuous discovery of new requirements. Feel free to contact us and share your thoughts and ideas about SCPM. SCPM is based on a flexible framework in an effort to allow even server-based profile management. Send your wishes, inspirations, and error descriptions via our web front-end at `http://www.suse.de/feedback/`.

16.2.1 Basic Terminology and Concepts

The following are some terms used in SCPM documentation and in the YaST module.

- The term *system configuration* refers to the complete configuration of the computer. It covers all fundamental settings, like use of partitions, network settings, time zone selection, and keyboard mappings.
- A *profile*, also called *configuration profile*, is a state that has been preserved and can be restored at any time.
- *Active profile* refers to the profile last selected. This does not mean that the current system configuration corresponds exactly to this profile, because the configuration can be customized at any time.
- A *resource* in the SCPM context is an element that contributes to the system configuration. This can be a file or a softlink including its metadata, like user, permissions, or access time. This can also be a system service that runs in this profile, but is deactivated in another one.
- Every resource belongs to a certain *resource group*. These groups contain all resources that logically belong together — most groups would contain both a service and its configuration files. It is very easy to assemble resources managed by SCPM because this does not require any knowledge about the configuration files of the desired service. SCPM ships with a selection of preconfigured resource groups that should be sufficient for most scenarios.

16.2.2 SCPM YaST Module and Additional Documentation

A YaST module (package `yast2-profile-manager`) is a graphical front-end to SCPM that provides an alternative to the command line front-end. Because the functionality of both front-ends is substantially the same and the knowledge of the command line front-end is useful in many cases, only the latter is described here. Differences between the YaST front-end and the command line tool are mentioned wherever appropriate.

Refer to the info pages of SCPM for the most recent documentation. Read these with tools like Konqueror (with the command `konqueror info:scpm`) or emacs. On the console, use `info` or `pinfo`. Technical information is provided at `/usr/share/doc/package/scpm`. Running `scpm` without any arguments returns a command option summary.

16.2.3 Configuring SCPM

SCPM must be activated before use. By default, SCPM handles network and printer settings as well as the XFree86 configuration. If you need to manage special services or configuration files, activate appropriate resource groups. To list the predefined resource groups, use `scpm list_groups`. To see only the groups already activated, use `scpm list_groups -a`. Issue these commands as `root` on the command line. Activate or deactivate a group with `scpm activate_group NAME` or `scpm deactivate_group NAME`. Replace `NAME` with the relevant group name. All the resource groups can also be configured with the YaST profile manager.

Activate SCPM with `scpm enable`. When run for the first time, SCPM is initialized, which takes a few seconds. Deactivate SCPM with `scpm disable` at any time to prevent the unintentional switching of profiles. A subsequent reactivation simply resumes the initialization.

16.2.4 Creating and Managing Profiles

A profile named `default` already exists after SCPM has been activated. Get a list of all available profiles with `scpm list`. This only existing profile is also the active one, which can be verified with `scpm active`. The profile `default` is a basic configuration from which the other profiles are derived. For this reason, all settings that should be identical in all profiles should be made first. These modifications are then stored in the active profile with `scpm reload`. The profile `default` can be used, renamed, or deleted.

There are two possibilities to add a new profile. If the new profile (named `work` here) should be based on the profile `default`, create it with `scpm copy default work`. The command `scpm switch work` changes into the new profile, which can then be modified. Sometimes the system configuration was modified for special purposes that should be kept in a new profile. The command `scpm add work` creates a new profile by saving the current system configuration in the profile `work` and marking it as active. Running `scpm reload` then saves changes to the profile `work`.

Rename or delete profiles with the commands `scpm rename x y` and `scpm delete x`. For example, to rename `work` to `project` use `scpm rename work project`. Delete `project` with `scpm delete project`. The active profile cannot be deleted.

The YaST module only offers an 'Add' button. Pressing it opens a dialog in which to select whether an existing profile should be copied or the current system configuration should be saved. Use 'Edit' for renaming.

16.2.5 Switching Configuration Profiles

The command `scpm switch work` switches to another profile (the profile `work`, in this case). Switch to the active profile to save modified settings of the system configuration. Alternatively, use `scpm reload` to do this.

When switching profiles, SCPM first checks which resources of the active profile have been modified. It then queries whether the modification of each resource should be added to the active profile or dropped. If you prefer a separate listing of the resources (as in former versions of SCPM) use the switch command with the `-r` parameter: `scpm switch -r work`.

SCPM then compares the current system configuration with the profile to which to switch. In this phase, SCPM evaluates which system services need to be stopped or restarted due to mutual dependencies or to reflect the changes in configuration. This is like a partial system reboot that concerns only a small part of the system while the rest continues operating without change.

It is only at this point that the system services are stopped, all modified resources (e.g., configuration files) are written, and the system services are restarted.

16.2.6 Advanced Profile Settings

You can enter a description for every profile that is displayed with `scpm list`. For the active profile, set it with `scpm set description "text"`. Provide the name of the profile for inactive profiles, for instance, `scpm set description "text" work`. Sometimes it might be desirable to perform additional actions not provided by SCPM while switching profiles. Attach up to four executables for each profile. They are invoked at different stages of the switching process. These stages are referred to as:

prestop prior to stopping services when leaving the profile

poststop after stopping services when leaving the profile

prestart prior to starting services when activating the profile

poststart after starting services when activating the profiles

Switching from profile `work` to profile `home` thus proceeds as follows:

1. The prestop action of the profile `work` is executed.
2. The services are stopped.
3. The poststop action of the profile `work` is executed.
4. The system configuration is changed.
5. The prestart action of the profile `home` is executed.
6. The services are started.
7. The poststart action of the profile `home` is executed.

Attach these actions with the command `set` by entering `scpm set prestop <filename>`, `scpm set poststop <filename>`, `scpm set prestart <filename>`, or `scpm set poststart <filename>`. The call must be made to an executable — scripts must refer to the correct interpreter and must be executable at least for the superuser.

Query all additional settings entered with `set` with `get`. The command `scpm get poststart`, for instance, returns the name of the poststart call or simply nothing if nothing has been attached.

Reset such settings by overwriting with `""`. This means that the command `scpm set prestop ""` removes the attached prestop program.

All `set` and `get` commands can be applied to an arbitrary profile in the same manner as comments are added. For example, `scpm get prestop <filename work>` or `scpm get prestop work`.

Caution

These scripts or programs should not be modifiable by any user because they are executed with the rights of the superuser. It is recommended to make scripts only readable to the superuser because they can contain sensitive information. It is best to provide these programs with the permissions `-rwx---- root root` with the commands `chmod 700 variablefilename` and `chown root.root <filename>`.

Caution

16.2.7 Profile Selection at Boot

It is possible to select a profile during the boot process by providing the boot parameter `PROFILE=<name-of-the-profile>` at the boot prompt. In the boot loader configuration (`/boot/grub/menu.lst`), the option `title` reflects the name of the profile. For example:

Example 16.1: *The File /boot/grub/menu.lst*

```
gfxmenu (hd0,5)/boot/message
color white/green black/light-gray
default 0
timeout 8

title work
   kernel (hd0,5)/boot/vmlinuz root=/dev/hda6 PROFILE=work
   initrd (hd0,5)/boot/initrd

title home
   kernel (hd0,5)/boot/vmlinuz root=/dev/hda6 PROFILE=home
   initrd (hd0,5)/boot/initrd

title road
   kernel (hd0,5)/boot/vmlinuz root=/dev/hda6 PROFILE=road
   initrd (hd0,5)/boot/initrd
```

For systems that use LILO as the boot loader, refer to File 16.2 on the next page as an example. Then you can select the desired profile at the boot prompt.

Example 16.2: **File /etc/lilo.conf**

```
boot    = /dev/hda
change-rules
reset
read-only
menu-scheme = Wg:kw:Wg:Wg
prompt
timeout = 80
message = /boot/message

  image  = /boot/vmlinuz
  label  = home
  root   = /dev/hda6
  initrd = /boot/initrd
  append = "vga=0x317 hde=ide-scsi PROFILE=home"

  image  = /boot/vmlinuz
  label  = work
  root   = /dev/hda6
  initrd = /boot/initrd
  append = "vga=0x317 hde=ide-scsi PROFILE=work"

  image  = /boot/vmlinuz
  label  = road
  root   = /dev/hda6
  initrd = /boot/initrd
  append = "vga=0x317 hde=ide-scsi PROFILE=road"
```

16.2.8 Troubleshooting

In most cases, SCPM should function smoothly. There are, however, some pitfalls, which are described here.

SCPM is currently not able to survive a system update. The difficulty lies in the fact that, with a system update, the data stored in the profiles is not cleanly updated by the automatic mechanisms. SCPM then detects a system update and refuses to work. In this situation, you should get an error message from SCPM that contains "your operating system installation changed/is unknown, read man page!" In this case, reinitialize SCPM with `scpm -f enbale`. Your profiles, however, will be lost and you must reconfigure them.

It can also sometimes occur that SCPM stops working during a switch procedure. This may be caused by some outside effect, such as a user abort, a power fault, another similar problem, or even an error in SCPM itself. In this case, an error message saying SCPM is locked appears the next time you start SCPM. This is for system safety, because the data stored in its database may differ from the state of the system. To solve this issue, delete the lock file with `rm /var/lib/scpm/#LOCK` then update your database with `scpm -s reload`. After this procedure, proceed as usual.

There is no real problem with changing the resource group configuration of an already initialized SCPM. However, you must run `scpm rebuild` after adding or deleting groups. This adds new resources to all profiles and removes the deleted ones. The deleted ones are then lost to the system. If there are different configurations for the same resource in different profiles, the deletion of resources might cause serious problems. The current profile, which is not touched by SCPM, will not be affected. If you reconfigure your system with YaST, the rebuild is handled by YaST.

16.3 IrDA — Infrared Data Association

IrDA (*Infrared Data Association*) is an industry standard for wireless communication with infrared light. Many laptops sold today are equipped with an IrDA-compatible transceiver that enables communication with other devices, such as printers, modems, LANs, or other laptops. The transfer speed ranges from 2400 bps to 4 Mbps.

There are two IrDA operation modes. The standard mode, SIR, accesses the infrared port through a serial interface. This mode works on almost all systems and is sufficient for most requirements. The faster mode, FIR, requires a special driver for the IrDA chip. Not all chip types are supported in FIR mode because of a lack of appropriate drivers. Set the desired IrDA mode in the BIOS of your computer. The BIOS also shows which serial interface is used in SIR mode.

Information about IrDA can be found in the IrDA how-to by Werner Heuser at `http://tuxmobil.org/Infrared-HOWTO/Infrared-HOWTO.html`. Additionally refer to the web site of the Linux IrDA Project `http://irda.sourceforge.net/`.

16.3.1 Software

The necessary kernel modules are included in the kernel package. The package `irda` provides the necessary helper applications for supporting the infrared interface. The documentation can be found at `/usr/share/doc/packages/irda/README` after the installation of the package.

16.3.2 Configuration

The IrDA system service is not started automatically by the booting process. Use the YaST runlevel module to change the settings of the system services. Alternatively, use `chkconfig`. Every few seconds, IrDA sends out a "discovery packet" to detect other peripheral devices in its neighborhood. This consumes a considerable amount of battery power. For this reason, IrDA is disabled by default and should only be started when needed. Manually activate it with `rcirda start` or deactivate it with `rcirda stop`. All kernel modules needed are loaded automatically when the interface is activated.

The file `/etc/sysconfig/irda` contains only the one variable `IRDA_PORT`. This is where the interface used in SIR mode is set. The script `/etc/irda/drivers` of the infrared support package sets this variable.

16.3.3 Usage

Data can be sent to the device file `/dev/irlpt0` for printing. The device file `/dev/irlpt0` acts just like the normal `/dev/lp0` cabled interface, except the printing data is sent wireless with infrared light. Printers used with the infrared interface are installed just like printers connected to the parallel or serial ports. Make sure the printer is in visible range of the infrared interface and the infrared support is started.

Communication with other hosts and with mobile phones or other similar devices is conducted through the device file `/dev/ircomm0`. The Siemens S25 and Nokia 6210 mobile phones, for instance, can dial and connect to the Internet with the wvdial application using the infrared interface. Synchronizing data with a Palm Pilot is also possible, provided the device setting of the corresponding application has been set to `/dev/ircomm0`.

Only those devices that support the printer or IrCOMM protocols can be accessed without any further adjustments. Devices that support the IROBEX protocol, such as the 3Com Palm Pilot, can be accessed with special applications, like irobexpalm and irobexreceive. Refer to the IR-HOWTO on this subject. The protocols supported by the device are stated in brackets behind the name of the device in the output of irdadump. IrLAN protocol support is still a "work in progress" — it is not stable yet, but will surely also be available for Linux in the near future.

16.3.4 Troubleshooting

If devices connected to the infrared port do not respond, use the command `irdadump` (as `root`) to check if the other device is recognized by the computer. Something similar to Example 16.3 appears regularly when a Canon BJC-80 printer is in visible range of the computer:

Example 16.3: *Output of irdadump*

```
21:41:38.435239 xid:cmd 5b62bed5 > ffffffff S=6 s=0 (14)
21:41:38.525167 xid:cmd 5b62bed5 > ffffffff S=6 s=1 (14)
21:41:38.615159 xid:cmd 5b62bed5 > ffffffff S=6 s=2 (14)
21:41:38.705178 xid:cmd 5b62bed5 > ffffffff S=6 s=3 (14)
21:41:38.795198 xid:cmd 5b62bed5 > ffffffff S=6 s=4 (14)
21:41:38.885163 xid:cmd 5b62bed5 > ffffffff S=6 s=5 (14)
21:41:38.965133 xid:rsp 5b62bed5 < 6cac38dc S=6 s=5 BJC-80
                        hint=8804 [Printer IrCOMM ] (23)
21:41:38.975176 xid:cmd 5b62bed5 > ffffffff S=6 s=* earth
                        hint=0500 [ PnP Computer ] (21)
```

Check the configuration of the interface if there is no output or the other device does not reply. Verify that the correct interface is used. The infrared interface is sometimes located at `/dev/ttyS2` or at `/dev/ttyS3` and another interrupt than IRQ 3 is sometimes used. These settings can be checked and modified in the BIOS setup menu of almost every laptop.

A simple CCD video camera can also help in determining whether the infrared LED lights up at all. Most video cameras can see infrared light, whereas the human eye cannot.

16.4 Bluetooth — Wireless Connections

Bluetooth is a wireless technology for connecting various devices. Two important characteristics distinguish Bluetooth from IrDA: the individual devices do not need to "see" each other directly and several devices can be connected in a network with a maximum data rate of 720 Kbps (in the current version 1.1). Theoretically, Bluetooth is even capable of communicating through walls. However, in practice this largely depends on the properties of the walls and the device class. There are three device classes with maximum transmission ranges between ten and a hundred meters.

16.4.1 Profiles

In Bluetooth, services are defined by means of profiles, such as the file transfer profile, the basic printing profile, and the personal area network profile. To enable a device to use the services of another device, both must understand the same profile — a piece of information that is often missing on the device package and in the manual. Although some manufacturers strictly comply with the definitions of the individual profiles, others do not. Nevertheless, the communication between the devices usually works smoothly.

16.4.2 Software

To be able to use Bluetooth, you need a Bluetooth adapter (built-in or external), drivers, and a Bluetooth protocol stack. By default, the Linux kernel contains the basic drivers needed for using Bluetooth. The Bluez system is used as protocol stack. Additionally, install all packages associated with Bluetooth (`bluez-libs`, `bluez-bluefw`, `bluez-pan`, `bluez-sdp`, and `bluez-utils`), as these provide some necessary services and utilities.

16.4.3 Configuration

The configuration files described in this section can only be modified by the user `root`. Currently, there is no graphical user interface for setting the parameters. Therefore, the files must be modified with a text editor.

A PIN number provides basic protection against unwanted connections. Mobile phones usually query the PIN when establishing the first contact (or when setting up a device contact on the phone). For two devices to be able to communicate, both must identify themselves with the same PIN. On the computer, the PIN is located in the file `/etc/bluetooth/pin`. Currently, only one PIN is supported in Linux, regardless of the number of installed Bluetooth devices. Thus, multiple devices cannot be addressed with different PINs. Accordingly, you need to set the same PIN on all devices or deactivate the PIN authentication entirely.

Note

Security of Bluetooth Connections

Despite the PINs, the transmission between two devices may not be entirely secure.

Note

`/etc/bluetooth/hcid.conf`is main configuration file for Linux Bluetooth. Various settings, such as the device names and the security modes, can be modified in this file. Usually, the settings should be adequate. The file contains comments describing the options for the various settings.

`security auto;` is one of the most important settings. If necessary, a PIN is activated for the identification. If problems are encountered, the option `auto` disables the PIN. Depending on your preferences and your security needs, set this option to `none` never to use PIN numbers or to `user` to use PIN numbers.

Another important section is the one beginning with `device {`. In this section, define the name under which the host should be displayed on the other side. The device class (`Laptop`, `Server`, etc.), authentication, and encryption are defined in this section.

16.4.4 System Components and Useful Tools

The operability of Bluetooth depends on the interaction of various services. At least two background daemons are needed: hcid (*host controller interface*), which serves as an interface for the Bluetooth device and controls it, and sdpd (*service discovery protocol*), by means of which a device can find out which services the host makes available. If they are not activated automatically when the system is started, both hcid and sdpd can be activated with the command `rcbluetooth start`. This command must be executed as `root`.

The following paragraphs describe the main tools needed for working with Bluetooth. Currently, only command-line programs are available. Extensions for Konqueror (KDE desktop) and Nautilus (GNOME desktop) are under development. This functionality may have been implemented after the editorial deadline. In this case, the URL `sdp://` should display local Bluetooth devices (physically connected to the host) as well as remote Bluetooth devices (accessible by way of a wireless connection).

Note

Other functionalities of the above-mentioned Bluetooth applications can be viewed with `man <program_name>`.

Note

Some of the commands can only be executed as `root`. This includes the command `l2ping <device_address>` for testing the connection to a remote device.

hcitool

hcitool can be used to determine whether local and remote devices are detected. The command `hcitool dev` should list your devices. The output generates a line in the form `<interface_name> <device_address>` for every detected local device.

The command `hcitool name <device_address>` can be used to determine the device name of a remote device. If, for example, another computer is detected, the displayed class and device name corresponds to the information in the file `/etc/bluetooth/hcid.conf` on the remote computer. Local devices addresses generate an error output.

hciconfig

Get more information about the local device with `/sbin/hciconfig`. Search for remote devices (that are not connected physically to the host) with the command `hcitool inq`. Three values are displayed for every detected device: the device address, the clock offset, and the device class. The device address is important, as other commands use it for identifying the target device. The clock offset mainly serves technical purposes. In the class, the device type and the service type are encoded as a hexadecimal value.

sdptool

The program sdptool can be used to check which services are made available by a specific device. The command `sdptool browse <device_address>` returns all services of a device. The command `sdptool search <service_code>` can be used to search for a specific service. This command scans all accessible devices for the requested service. If one of the devices offers the service, the program prints the (full) service name returned by the device together with a brief description. A list of all possible service codes can be viewed by entering `sdptool` without any parameters.

16.4.5 Examples

The following two examples demonstrate some of the capabilities of Bluetooth.

Network Connection between Two Hosts

The first example shows the establishment of a network connection between two hosts with pand (*personal area networking*). The following commands must be executed by the user `root`. The description focuses on the Bluetooth-specific actions and does not provide a detailed explanation of the network command (ip):

Start pand with the command `pand -s` on one of the two hosts (referred to as *H1*). Determine the device address of the second host (*H2*) by running `hcitool inq` on this host. Run `pand -c <device_address>` to establish a connection. If you query the available network interfaces with `ip link show`, an entry such as the following should be displayed (the local device address should be displayed instead of `00:12:34:56:89:90`):

```
 bnep0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000
    link/ether 00:12:34:56:89:90 brd ff:ff:ff:ff:ff:ff
```

This interface must be assigned an IP address and activated. This can be done with the following two commands. On *H1*:

```
ip addr add 192.168.1.3/24 dev bnep0
ip link set bnep0 up
```

On *H2*:

```
ip addr add 192.168.1.4/24 dev bnep0
ip link set bnep0 up
```

Now *H1* can be accessed from *H2* under the IP `192.168.1.3`. Use the command `ssh 192.168.1.4` to access *H2* from *H1* (provided *H2* runs an sshd, which is activated by default in SUSE LINUX). The command `ssh 192.168.1.4` can also be run as a "normal" user.

File Transfer from a Mobile Phone to the Host

The second example shows how to transfer a photograph created with a mobile phone with a built-in digital camera to a computer (without incurring additional costs for the transmission of a multimedia message). Although the menu structure may differ on various mobile phones, the procedure is usually quite similar. Refer to the manual of your phone, if necessary. This example describes the transfer of a photograph from a Sony Ericsson mobile phone to a laptop. The service Obex-Push must be available on the computer and the computer must grant the mobile phone access. In the first step, the service is made available on the laptop. This is done by means of the opd daemon from the package `bluez-utils`. Start the daemon with the following command:

```
opd --mode OBEX --channel 10 --daemonize --path /tmp --sdp
```

Two important parameters are used: `--sdp` registers the service with the sdpd and `--path /tmp` instructs the program where to save the received data — in this case to `/tmp/`. You can also specify any other directory to which you have write access.

Now the mobile phone must "get to know" the computer. To do this, open the 'Connect' menu on the phone and select 'Bluetooth'. If necessary, click 'Turn On' before selecting 'My devices'. Select 'New device' and let your phone search for the laptop. If a device is detected, its name appears in the display. Select the device associated with the laptop. If you encounter a PIN query, enter the PIN specified in `/etc/bluetooth/pin`. Now your phone knows the laptop and is able to exchange data with the laptop. Exit the current menu and go to the image menu. Select the image to transfer and press 'More'. In the next menu, press 'Send' to select a transmission mode. Select 'Via Bluetooth'. The laptop should be listed as a target device. Select the laptop to start the transmission. The image is then saved to the directory specified with the opd command. In the same way, you can also transfer audio tracks to the laptop.

16.4.6 Troubleshooting

If you have difficulties establishing a connection, proceed as follows:

- Check the output of `hcitool dev`. Is the local device listed? If not, the hcid may not have been started at all or the device may not be recognized as a Bluetooth device (either because the driver is not able to do this or because the device is defective). Restart the daemon with the command `rcbluetooth restart` and check `/var/log/messages` to see if any errors occurred.
- Does the computer "see" other devices when you execute `hcitool inq`? Try the command twice — the connection may have been faulty, as the frequency band for Bluetooth is also used by other devices.
- Make sure that the PIN in `/etc/bluetooth/pin` is the same as the PIN of the remote device.
- Try to establish the connection from the other device. Check if this device sees the computer.
- The first example (network connection) does not work. This may be due to various reasons. Possibly one of the two hosts does not understand the ssh protocol. Try if `ping 192.168.1.3` or `ping 192.168.1.4` works. If it does, check if sshd is active. Another problem could be that you already have other addresses that conflict with the address `192.168.1.X` used in the example. If this is the case, try other addresses, such as `10.123.1.2` and `10.123.1.3`.
- In the second example, the laptop does not appear as the target device. Does the mobile device recognize the Obex-Push service on the laptop? In 'My devices', select the respective device and view the list of 'Services'. If Obex-Push is not displayed (even after the list is updated), the problem is caused by the opd on the laptop. Is the opd active? Do you have write access to the specified directory?
- Does the second example work in reverse order? If obexftp is installed, this should work with `obexftp -b <device_address> -B 10 -p <image>` on some devices (Siemens and Sony Ericsson have been tested, other devices may or may not support this action).

16.4.7 For More Information

An extensive overview of various instructions for the use and configuration of Bluetooth is available at `http://www.holtmann.org/linux/bluetooth/`.

Useful information and instructions:

Connection to PalmOS PDA:

`http://www.cs.ucl.ac.uk/staff/s.zachariadis/btpalmlinux.html`

Official howto for the integrated Bluetooth protocol stack in the kernel:

`http://bluez.sourceforge.net/howto/index.html`

Power Management

This chapter provides an overview of the various power management technologies in Linux. The configuration of all available APM (Advanced Power Management), ACPI (Advanced Configuration and Power Interface), and CPU frequency scaling settings are described in detail.

► **s390, zseries**
The features and hardware described in this chapter do not exist on IBM S/390 and zSeries rendering this chapter irrelevant for these platforms. ◄

17.1 Power Saving Functions 354
17.2 APM 356
17.3 ACPI 358
17.4 Rest for the Hard Disk 364
17.5 powersave 365
17.6 The YaST Power Management Module 371
17.7 WOL – Wake On LAN 373

Unlike APM, which was previously used on laptops for power management only, the hardware information and configuration tool ACPI is available on all modern computers (laptops, desktops, and servers). On many types of modern hardware, the CPU frequency can be adapted to the respective situation, which helps save valuable battery time especially on mobile devices (*CPU frequency scaling*).

All power management technologies require suitable hardware and BIOS routines. Most laptops and many modern desktops and servers meet these requirements. APM had been used in many older computers. As 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.

Note

Power Management for AMD64 Processors

AMD64 processors with a 64-bit kernel only support ACPI.

Note

17.1 Power Saving Functions

Although many of these functions are of general interest, they are especially important for mobile deployment. The following paragraphs describe the functions and which system offers them.

Standby This operating mode merely turns the display off. On some computers, the processor performance is throttled. This function is not available in all APM implementations. The corresponding ACPI state is *S1*.

Suspend (to memory) This mode writes the entire system state to the RAM. Subsequently, the entire system except the RAM is put to sleep. As the computer consumes very little power in this state, the battery may last anywhere from twelve hours to several days, depending on the device. The advantage of this state is the possibility to resume work at the same point within a few seconds, without having to boot and restart applications. Most modern devices can be suspended by closing the lid and activated by opening it. The corresponding ACPI state is *S3*. Support of this state largely depends on the hardware.

Hibernation (suspend to disk) This operating mode enables the computer to hibernate, as the entire system state is written to the hard disk and the system is powered off. The reactivation from the state of hibernation takes about 30 to 90 seconds. The state prior to the suspend will be restored. Some manufacturers offer useful hybrid variants of this mode in their APM (such as RediSafe in IBM Thinkpads). The corresponding ACPI state is *S4*.

Battery monitor In addition to monitoring the battery charge level, something must be done when power reserves are low. This control function is handled by ACPI or APM.

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 The most important component for saving power is the hard disk. 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 can be deactivated via ACPI (at least theoretically) or permanently in the BIOS setup.

Processor speed control AMD PowerNow! and Intel SpeedStep are two concepts designed for reducing the power consumption of the overall system. For this purpose, the power consumption of the most power-hungry component — the processor — is reduced. A pleasant side-effect of the reduced processor speed is the reduced generation of heat. Thus, adjustable fans will also make less noise. This feature is controlled by the CPU frequency scaling functions of the Linux kernel. Basically, three different processor speed levels are distinguished:

performance Maximum processor performance for AC operation.

powersave Minimum processor performance for battery operation.

dynamic Dynamic adaption of the processor performance to the current processor load — this is the recommended setting for battery operation and AC operation in order to save battery power, reduce noise, and achieve optimum performance. Switching between the speed levels usually takes place seamlessly, unnoticed by the user.

See Section 17.5 on page 365 for more information about controlling the processor speed.

17.2 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 a suitable kernel and the respective packages.

By default, APM support is integrated in the kernels shipped with SUSE LINUX. However, APM is only activated if no ACPI is 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`.

Strange things may happen if the BIOS implementation does not fully comply with the standard. Some problems can be circumvented with special boot parameters (formerly kernel configuration options). All parameters are entered at the boot prompt in the form `apm=<parameter>`:

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.

17.2.1 The APM Daemon (apmd)

The `apmd` daemon (package `apmd`) monitors the battery and can trigger certain actions when a standby or a suspend event occurs. Although it is not mandatory for operation, it may be useful for some problems.

`apmd` is not started automatically when the system is booted. If you want it started automatically, edit the settings for the system services with the YaST runlevel editor. Alternatively, this can also be done with the `chkconfig` utility. The daemon can be started manually with the command `rcapmd start`.

A number of configuration variables are available in `/etc/sysconfig/powermanagement`. As the file is commented, only some information is provided here:

APMD_ADJUST_DISK_PERF Adapts the disk performance to the power supply status. This can be done with a number of additional variables beginning with `APMD_BATTERY` (for battery operation) or `APMD_AC` (for AC operation).

APMD_BATTERY/AC_DISK_TIMEOUT
Disk inactivity period after which the disk is spun down. The values are described in Section 17.4 on page 364 or in the manual page for `hdparm`, option `-S`.

APMD_BATTERY/AC_KUPDATED_INTERVAL
Interval between two cycles of the kernel update daemon.

APMD_BATTERY/AC_DATA_TIMEOUT
Maximum age of buffered data.

APMD_BATTERY/AC_FILL_LEVEL
Maximum fill level of the hard disk buffer.

APMD_PCMCIA_EJECT_ON_SUSPEND
Although PCMCIA is implemented with APM support, difficulties may sometimes be encountered. Some card drivers do not resume correctly after a suspend (`xirc2ps_cs`). Therefore, apmd can deactivate the PCMCIA system prior to the suspend and reactivate it afterwards. To do this, set this variable to `yes`.

APMD_INTERFACES_TO_STOP Set network interfaces to stop prior to a suspend and restart afterwards.

APMD_INTERFACES_TO_UNLOAD
Use this variable if you also need to unload the driver modules of these interfaces.

APMD_TURN_OFF_IDEDMA_BEFORE_SUSPEND
Sometimes, resuming after a suspend may not work if an IDE device (hard disk) is still in DMA mode.

Other options include the possibility to correct the key repeat rate or the clock after a suspend or to shut down the laptop automatically when the APM BIOS send a "battery critical" event. To execute special actions, adapt the script `/usr/sbin/apmd_proxy` (performs the tasks listed above) to your needs.

17.2.2 Further Commands

apmd contains a number of useful tools. `apm` can be used to query the current battery charge level and to set the system to standby (`apm -S`) or suspend (`apm -s`). Refer to the manual page of `apm`. The command `apmsleep` suspends the system for a specified time. To watch a log file without keeping the hard disk spinning, use `tailf` instead of `tail -f`.

There are also tools for the X Window System. apmd contains the graphical utility xapm for displaying the battery charge level. If you use the KDE desktop or at least kpanel, use kbatmon to view the battery charge level and suspend the system. xosview is another interesting alternative.

17.3 ACPI

ACPI (Advanced Configuration and Power Interface) was designed to enable the operating system to set up and control the individual hardware components. Thus, ACPI supersedes both PnP and APM. Moreover, ACPI 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. As the operating system executes commands stored in the BIOS, the functionality depends on the BIOS implementation. The tables ACPI is able to detect and load are reported in `/var/log/boot.msg`. See Section 17.3.4 on page 362 for more information about troubleshooting ACPI problems.

17.3.1 ACPI in Action

If the kernel detects an ACPI BIOS when the system is booted, ACPI is activated automatically (and APM is deactivated). The boot parameter `acpi=on` 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. If this is the case, there is a directory `/proc/acpi/`, which is described later.

Subsequently, a number of modules must be loaded. This is done by the start script of the ACPI daemon. If any of these modules causes 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.

In `/proc/acpi/`, find a number of files that provide information on the system state or can be used to change some of the states actively. However, many features do not work yet, either because they are still under development or because they have not been implemented by the manufacturer.

All files (except `dsdt` and `fadt`) can be read with `cat`. In some files, settings can be modified by entering `echo X <file>` to specify suitable values for X (the objects in `/proc` are not real files on the hard disks but interfaces to the kernel). The most important files are described below:

`/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 a daemon like `acpid` or `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 described below (see Section 17.3.3 on page 362). The charge level at which a battery event is triggered can be specified in `alarm`.

`/proc/acpi/button` This directory contains information about various switches.

`/proc/acpi/fan/FAN/state` Shows if the fan is currently active. The fan can be activated and deactivated 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 it gets too warm.

`/proc/acpi/processor/CPU*/info`
Information about the energy saving options of the processor.

`/proc/acpi/processor/CPU*/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` figure.

`/proc/acpi/processor/CPU*/performance`
This interface is no longer used.

`/proc/acpi/processor/CPU*/throttling`
Enables further linear throttling of the processor. This interface is outdated. Its function has been taken over by the settings in `/etc/sysconfig/powersave/common` (see Section 17.5.2 on page 368).

`/proc/acpi/processor/CPU*/limit`
If the performance 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. However, their function has been taken over by the settings in `/etc/sysconfig/powersave/common` (see Section 17.5.2 on page 367).

`/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, as the life span of the hardware is at stake. Therefore, some of the following descriptions 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 will always be “ok”.

`/proc/acpi/thermal_zone/*/cooling_mode`
Enables the selection of the passive (less performance, very economical) or active (full performance, uninterrupted fan noise) cooling method for full ACPI control.

`/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”).

`/proc/acpi/thermal_zone/*/polling_frequency`
If the value in `temperature` is not updated automatically when the temperature changes, the polling mode can be toggled 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.

17.3.2 The ACPI Daemon (acpid)

Like the APM daemon, the ACPI daemon processes certain events. Currently, the only supported events are the actuation of switches, such as the power button or the lid contact. All events are logged in the system log. The actions to perform in response to these events can be determined in the variables `ACPI_BUTTON_POWER` and `ACPI_BUTTON_LID` in `/etc/sysconfig/powermanagement`. For more options, modify the script `/usr/sbin/acpid_proxy` or the `acpid` configuration in `/etc/acpi/`.

Unlike `apmd`, little is preconfigured here, as ACPI in Linux is still in a very dynamic development stage. If necessary, configure `acpid` according to your needs. If you have any suggestions regarding preparatory actions, contact us through `http://www.suse.de/feedback`.

17.3.3 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`).

17.3.4 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. This will solve many problems. If the computer does not boot properly, one of the following boot parameters may be helpful:

pci=noacpi Do not use ACPI for configuring the PCI devices.

acpi=oldboot Only perform a simple resource configuration. Do not use ACPI for other purposes.

acpi=off Disable ACPI.

Caution

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.

Caution

Take a closer look at the boot messages. This can be done with the command `dmesg | grep -2i acpi` (or all messages, as 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 will be ignored. The procedure is described in Section 17.5.4 on page 369.

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 manufacturer of the device. Although manufacturers may not always be able to provide assistance for Linux, it is still important that they hear the word “Linux” as often as possible. Manufacturers will only take the issue seriously if they realize that an adequate number of their customers use Linux.

Additional documentation and help:

- `http://www.columbia.edu/~ariel/acpi/acpi_howto.txt` (slightly outdated ACPI HowTo, incomplete)
- `http://www.cpqlinux.com/acpi-howto.html` (more 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)

17.4 Rest for the Hard Disk

In Linux, a hard disk that is not used can be put to sleep. The hdparm utility modifies various hard disk settings. The option `-y` instantly switches the hard disk to the standby mode; the option `-Y` (caution) puts it to sleep. `hdparm -S <x>` causes the hard disk to be spun down after a certain period of inactivity. The placeholder ⟨*x*⟩ can be used 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.

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 five seconds and notifies the bdflush daemon when data is older than thirty seconds or the buffer reaches a fill level of thirty percent. The bdflush daemon then writes the data to the hard disk. It also writes independently from kupdated if, for instance, the buffer is full. On a stable system, these settings can be modified. However, do not forget that this may have a detrimental effect on the data integrity.

Caution

Impairment of the Data Integrity

Changes to the kernel update daemon settings affect the data integrity. Do not touch these settings if you are not sure.

Caution

The settings for the hard disk time-out, the kupdated interval, the buffer threshold, and the age limit for data can be specified in `/etc/sysconfig/powermanagement` for battery operation and for AC operation. The variables are described in Section 17.2.1 on page 357 and in the file itself. Further information is available in `/usr/share/doc/packages/powersave`.

Apart from these processes, journaling file systems, like ReiserFS and Ext3, write their meta data 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 will access the hard disk far less frequently. However, this is irrelevant if the interval for kupdated was increased.

17.5 powersave

On laptops, the `powersave` package can be used to control the power saving function during battery operation. Some of the features of this package can also be used on normal workstations and servers (suspend, standby, ACPI button functionality, putting IDE hard disks to sleep).

This package comprises all power management features of your computer. It supports hardware using ACPI, APM, IDE hard disks, and PowerNow! or SpeedStep technologies. The functionalities from the packages `apmd`, `acpid`, `ospmd`, and `cpufreqd` (now `cpuspeed`) have been consolidated in the `powersave` package. For this reason, daemons from these packages should not be run together with the powersave daemon.

Even if your system does not have all hardware elements listed above (APM and ACPI are mutually exclusive), use the powersave daemon for controlling the power saving function. The daemon will automatically detect any changes in the hardware configuration.

Note

Information about powersave

Apart from this chapter, topical information about the powersave package is also available in `/usr/share/doc/packages/powersave/README_POWERSAVE`.

Note

17.5.1 Configuration of powersave

Normally, the configuration of powersave is distributed to several files:

`/etc/powersave.conf` The powersave daemon needs this file for delegating system events to the powersave_proxy. Additionally, custom settings for the behavior of the daemon can be made in this file.

`/etc/sysconfig/powersave/common`
This file provides the general configuration of the start-up script (`rcpowersave`) and the proxy. Usually, the default settings can be adopted as they are.

`/etc/sysconfig/powersave/scheme_*`
These are the various schemes or profiles that control the adaption of the power consumption to specific scenarios, some of which are already preconfigured and ready to use without any changes. Any custom profiles can be saved here.

17.5.2 Configuration of APM and ACPI

Suspend and Standby

In the file `/etc/sysconfig/powersave/common`, specify any critical modules and services that need to be unloaded or stopped prior to a suspend or standby event. When the system operation is resumed, these modules and services will be reloaded or restarted. The default settings mainly affect USB and PCMCIA modules.

POWERSAVE_SUSPEND_RESTART_SERVICES=""
List the services to restart after a suspend.

POWERSAVE_STANDBY_RESTART_SERVICES=""
List the services to restart after a standby.

POWERSAVE_UNLOAD_MODULES_BEFORE_SUSPEND=""
List the modules to unload before a suspend.

POWERSAVE_UNLOAD_MODULES_BEFORE_STANDBY=""
List the modules to unload before a standby.

Make sure that the following standard options for the correct processing of suspend, standby, occurrence, and resume are set (normally, these are the default settings following the installation of SUSE LINUX):

```
POWERSAVE_EVENT_GLOBAL_SUSPEND="prepare_suspend"
POWERSAVE_EVENT_GLOBAL_STANDBY="prepare_standby"
POWERSAVE_EVENT_GLOBAL_RESUME_SUSPEND="restore_after_suspend"
POWERSAVE_EVENT_GLOBAL_RESUME_STANDBY="restore_after_standby"
```

In `/etc/powersave.conf` (configuration file of the powersave daemon), these events are allocated to the `powersave_proxy` script. This script is executed when these events occur (default setting following the installation):

```
global.suspend=/usr/sbin/powersave_proxy
global.standby=/usr/sbin/powersave_proxy
global.resume.suspend=/usr/sbin/powersave_proxy
global.resume.standby=/usr/sbin/powersave_proxy
```

User-Defined Battery States

In the file `/etc/powersave.conf`, define three battery charge levels (in percent) that trigger system alerts or execute specific actions when they are reached.

```
POWERSAVED_BATTERY_WARNING=20
POWERSAVED_BATTERY_LOW=10
POWERSAVED_BATTERY_CRITICAL=5
```

The actions or scripts to execute when the charge levels drops under the specified limits are defined in `/etc/powersave.conf`. The action type is configured in `/etc/sysconfig/powersave/common`:

```
POWERSAVE_EVENT_BATTERY_NORMAL="ignore"
POWERSAVE_EVENT_BATTERY_WARNING="notify"
POWERSAVE_EVENT_BATTERY_LOW="notify"
POWERSAVE_EVENT_BATTERY_CRITICAL="suspend"
```

Further options are explained in this configuration file.

Adapting the Power Consumption to Various Conditions

The system behavior can be adapted to the type of power supply. Thus, the power consumption of the system should be reduced when the system is disconnected from the AC power supply and operated with the battery. In the same way, the performance should automatically be increased as soon as the system is connected to the AC power supply. The CPU frequency, the power saving function of IDE hard disks, and some other factors can be modified.

In `/etc/powersave.conf`, the execution of the actions triggered by the disconnection from or connection to the AC power supply is delegated to `powersave_proxy`. The scenarios (called schemes or profiles) to apply can be defined in `/etc/sysconfig/powersave/common`:

```
POWERSAVE_AC_SCHEME="performance"
POWERSAVE_BATTERY_SCHEME="powersave"
```

The schemes are located in files designated as `scheme_<name of the scheme>` in `/etc/sysconfig/powersave/`. The example refers to two schemes: `scheme_performance` and `scheme_powersave`. `performance`, `powersave`, and `acoustic` are preconfigured. The YaST Power Management module can be used to edit, create, and delete schemes or change their association with specific power supply states.

17.5.3 Additional ACPI Features

If you use ACPI, you can control the response of your system to *ACPI buttons* (Power, Sleep, Lid Open, Lid Closed). In `/etc/powersave.conf`, the execution of the respective actions is delegated to the powersave_proxy. The action itself is defined in the file `/etc/sysconfig/powersave/common`. The individual options are explained in this configuration file.

POWERSAVE_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.).

POWERSAVE_EVENT_BUTTON_SLEEP="suspend"
When the sleep button is pressed, the system is set to the suspend mode.

POWERSAVE_EVENT_BUTTON_LID_OPEN="ignore"
Nothing happens when the lid is opened.

POWERSAVE_EVENT_BUTTON_LID_CLOSED="screen_saver"
When the lid is closed, the screen saver is activated.

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 `POWERSAVED_CPU_LOW_LIMIT` and the time-out in `POWERSAVED_CPU_IDLE_TIMEOUT`.

17.5.4 Troubleshooting

All error messages and alerts are logged at `/var/log/messages`. If you cannot find the needed information, use the variable `DEBUG` for powersave in the file `/etc/sysconfig/powersave/common` to increase the verbosity of the messages. Increase the value of the variable to 7 or even 15 and restart the daemon. The error messages in `/var/log/messages` will now be more detailed, enabling you to identify the error. The following items cover the most frequent problems in connection with powersave.

ACPI is activated, but the battery states and buttons do not work as configured

If you experience problems with ACPI, use the following command to search the output of `dmesg` for ACPI-specific messages: `dmesg|grep -i acpi`. 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. Request 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/tables`. 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), it must be compiled with iasl (package `pmtools`). To do this, 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 uninstall the kernel and use `mkinitrd` to create an initrd, the modified DSDT will be 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/common` by means of the variables `CPUFREQD_MODULE` and `CPUFREQD_MODULE_OPTS`.

Suspend and standby do not work

There are several kernel-related problems that prevent the use of suspend and standby on **ACPI** systems:

- Currently, systems with more than 1 GB RAM do not (yet) support suspend.
- Currently, multiprocessor systems and systems with a P4 processor (with hyperthreading) do not support suspend.

The error may also be due to a faulty DSDT implementation (BIOS). If this is the case, install a new DSDT as described under *Although I activated ACPI, the battery states and buttons do not work as configured.*.

On ACPI and APM systems:

When the system attempts to unload faulty modules, the proxy is arrested and the suspend event is not triggered. The same can happen if you do not unload modules or stop services that prevent a successful suspend. In both cases, try to identify the modules causing the problem by manipulating the following settings in `/etc/sysconfig/powersave/common`:

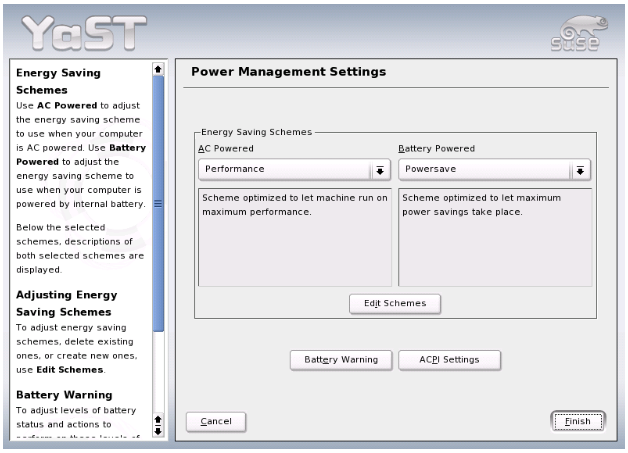
```
POWERSAVE_UNLOAD_MODULES_BEFORE_SUSPEND=""
POWERSAVE_UNLOAD_MODULES_BEFORE_STANDBY=""
POWERSAVE_SUSPEND_RESTART_SERVICES=""
POWERSAVE_STANDBY_RESTART_SERVICES=""
```

When using ACPI, the powersave daemon does not notice a certain battery limit

With ACPI, the operating system can request the BIOS to send a message when the battery charge level drops under a certain limit. The advantage of this method is that the battery state does not need to be polled constantly, which would impair the performance of the computer. However, this notification may not take place when the charge level drops under the specified limit, even though the BIOS supposedly supports this feature. If this happens on your system, set the variable `POWERSAVED_FORCE_BATTERY_POLLING` in the file `/etc/powersave.conf` to `yes` to force battery polling.

17.6 The YaST Power Management Module

The YaST Power Management module serves the configuration of all power management settings described above. When starting the module from the YaST Control Center (‘System’ → ‘Power Management’), the first dialog of the module is displayed (see Figure 17.1). In this dialog, select the schemes to use for battery operation and AC operation.



Energy Saving Schemes
Use **AC Powered** to adjust the energy saving scheme to use when your computer is AC powered. Use **Battery Powered** to adjust the energy saving scheme to use when your computer is powered by internal battery.

Below the selected schemes, descriptions of both selected schemes are displayed.

Adjusting Energy Saving Schemes
To adjust energy saving schemes, delete existing ones, or create new ones, use **Edit Schemes**.

Battery Warning
To adjust levels of battery status and actions to

Power Management Settings

Energy Saving Schemes

AC Powered: Performance

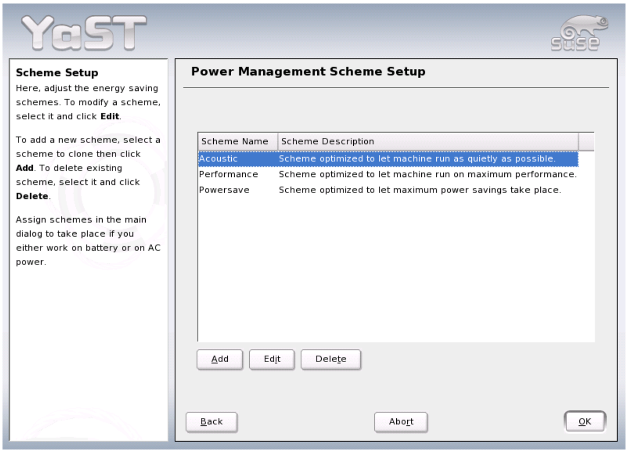
Scheme optimized to let machine run on maximum performance.

Battery Powered: Powersave

Scheme optimized to let maximum power savings take place.

Figure 17.1: YaST Power Management: Scheme Selection

In this dialog, select one of the existing schemes or click ‘Edit Schemes’ to access an overview of the existing schemes. This is shown in Figure 17.2.



Scheme Setup

Here, adjust the energy saving schemes. To modify a scheme, select it and click **Edit**.

To add a new scheme, select a scheme to clone then click **Add**. To delete existing scheme, select it and click **Delete**.

Assign schemes in the main dialog to take place if you either work on battery or on AC power.

Power Management Scheme Setup

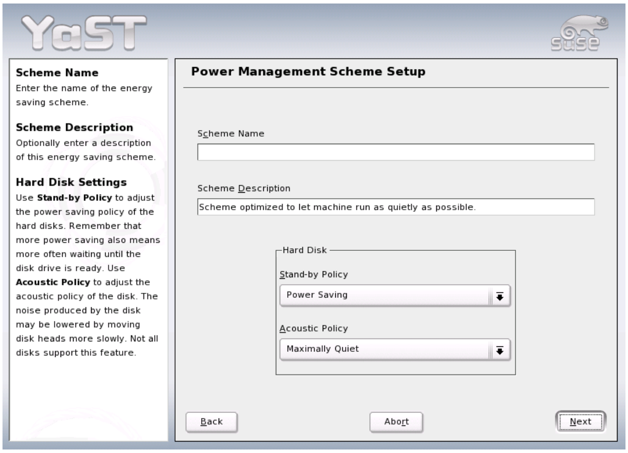
Scheme Name	Scheme Description
Acoustic	Scheme optimized to let machine run as quietly as possible.
Performance	Scheme optimized to let machine run on maximum performance.
Powersave	Scheme optimized to let maximum power savings take place.

Figure 17.2: YaST Power Management: Overview of Existing Schemes

In the schemes overview, select a scheme and click ‘Edit’ to access the editing dialog (see Figure 17.3 on the next page). To create a new scheme, click ‘Add’. The subsequent dialog is the same in both cases.

First, enter a suitable name and description for the new or edited scheme. 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. Click ‘Next’ to enter the ‘CPU’ and ‘Cooling Policy’ dialog. ‘CPU’ comprises the options ‘CPU Frequency Scaling’ and ‘Throttling’. Use these options to define if and to what extent the CPU frequency may be throttled. The ‘Cooling Policy’ determines the cooling method. Complete all settings for the scheme and click ‘OK’ to return to the start dialog (Figure 17.1 on the preceding page). In the start dialog, assign the custom scheme to one of the two operating modes. To activate your settings, exit this dialog with ‘OK’.

Apart from selecting schemes for various operating modes, global power management settings can also be made from the start dialog using the ‘Battery Warnings’ or ‘ACPI Settings’ buttons (see Figure 17.1 on the page before). Click ‘Battery Warnings’ to access the dialog for the battery charge level, shown in Figure 17.4 on page 374.



Scheme Name
Enter the name of the energy saving scheme.

Scheme Description
Optionally enter a description of this energy saving scheme.

Hard Disk Settings
Use **Stand-by Policy** to adjust the power saving policy of the hard disks. Remember that more power saving also means more often waiting until the disk drive is ready. Use **Acoustic Policy** to adjust the acoustic policy of the disk. The noise produced by the disk may be lowered by moving disk heads more slowly. Not all disks support this feature.

Scheme Name:

Scheme Description: Scheme optimized to let machine run as quietly as possible.

Stand-by Policy: Power Saving

Acoustic Policy: Maximally Quiet

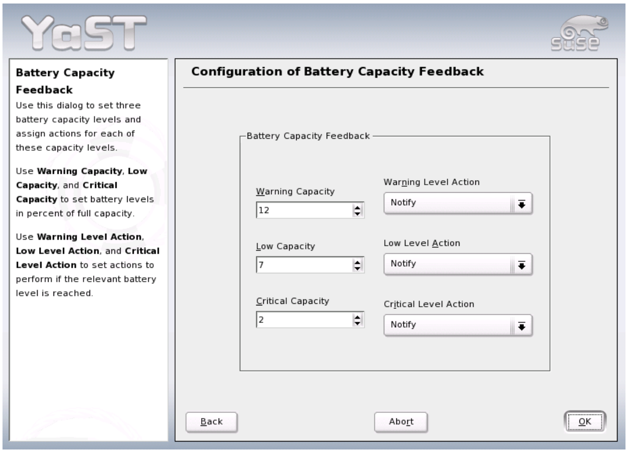
Figure 17.3: YaST Power Management: Adding a Scheme

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 suspend, as the remaining energy is not sufficient for continued system operation. Select suitable charge levels and the respective actions then click ‘OK’ to return to the start dialog. Access the dialog for configuring the ACPI buttons using the ‘ACPI Settings’ button. It is shown in Figure 17.5 on page 375.

The settings for the ACPI buttons determine how the system should respond to the actuation of 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 (Figure 17.1 on page 371). Click ‘OK’ again to exit the module and confirm your power management settings.

17.7 WOL – Wake On LAN

WOL (*Wake on LAN*) refers to the possibility of “waking up” a computer from standby mode over the network, using special packages. This “magic



Battery Capacity Feedback
Use this dialog to set three battery capacity levels and assign actions for each of these capacity levels.

Use **Warning Capacity**, **Low Capacity**, and **Critical Capacity** to set battery levels in percent of full capacity.

Use **Warning Level Action**, **Low Level Action**, and **Critical Level Action** to set actions to perform if the relevant battery level is reached.

Configuration of Battery Capacity Feedback

Battery Capacity Feedback

Warning Capacity: 12
Warning Level Action: Notify
Low Capacity: 7
Low Level Action: Notify
Critical Capacity: 2
Critical Level Action: Notify

Figure 17.4: *YaST Power Management: Battery Charge Level*

packet" is received by the network card and ensures that the motherboard activates the power supply to the power source and boots the computer.

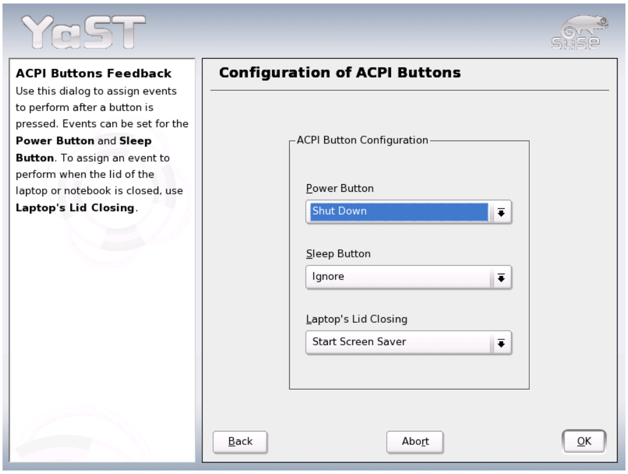
The advantage of this method is that computers do not have to be permanently switched on (which saves energy), yet they can be activated via WOL.

Note

Support for WOL

Wake on LAN only works with more recent motherboards which support this functionality in their BIOS. WOL-capable network cards mostly contain chips in the Intel i82557 (EEpro100B), i82558 (EEPro100+) or i82559 series. Further information on this is available from
`http://support.intel.com/support/network/sb/cs-008459.htm`.
Unfortunately, a lot of hardware does not have this functionality, even though it is apparently WOL-capable. Unfortunately, there is no alternative but to try out the relevant steps.

Note



ACPI Buttons Feedback

Use this dialog to assign events to perform after a button is pressed. Events can be set for the **Power Button** and **Sleep Button**. To assign an event to perform when the lid of the laptop or notebook is closed, use **Laptop's Lid Closing**.

Configuration of ACPI Buttons

ACPI Button Configuration

Power Button: Shut Down

Sleep Button: Ignore

Laptop's Lid Closing: Start Screen Saver

Figure 17.5: *YaST Power Management: ACPI Settings*

17.7.1 BIOS Configuration

Before you use WOL, you need to enable an option in BIOS which is frequently labeled 'OnBoard LAN' or 'Boot from LAN'. Depending on your particular variant of BIOS, it can be found in the 'OnBoard Device Configuration', 'Boot' or 'PowerSave' menu. In case of doubt, consult the documentation for your motherboard.

You should further check that your server has the latest BIOS and, if necessary, update it. Information on BIOS updates can be found on the homepage of the relevant motherboard vendor.

Caution

BIOS updates

A BIOS update is tricky to implement. It is therefore imperative that you stick closely to the instructions issued by your motherboard vendor, as otherwise your motherboard could become inoperable and your server could no longer start up.

Caution

Older network cards (for example, 3COM) have to be connected to the motherboard with a 3-pin cable. On newer network cards, this procedure is no longer necessary.

17.7.2 Configuration With YaST

Start up YaST as `root` user and call the module via menu selection 'Network Services' → 'WOL'. If there is a DHCP server running on your computer, the WOL module will display the existing computers on your network which you can include in your WOL list.

If a DHCP server is not running, then you will have to enter the remote computers manually. Click 'Add' and enter the host name and MAC (*Media Access Connector*) address for the network card. The MAC address is unique for every network device and can be displayed through the following call:

```
# ip link show eth0
2: eth0: <BROADCAST,MULTICAST,NOTRAILERS,UP>
mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:80:c8:94:c3:e7 brd ff:ff:ff:ff:ff:ff
```

In the 'Enter the MAC Address of the client:' field, enter the value contained in 'link/ether'. Confirm with 'Save'. The configuration with YaST is now complete.

17.7.3 Waking Up Computers

If your computers are appropriately configured, you can wake them up in the WOL YaST module by clicking 'Wake up'. The selected computer will then start up.

Another option is to enter the command `ether-wake` from the `netdiag` package. With this command, a particular computer can be woken up by entering its MAC address, as follows:

```
# ether-wake 00:80:C8:94:C3:E7
```

You can obtain help on this command through `--help` or `-u`.

17.7.4 Further Information

Further information is available in the mini HOWTO for WOL at `http://gsd.di.uminho.pt/jpo/software/wakeonlan/mini-howto/wol-mini-howto.html`.

File Systems in Linux

Linux supports a number of different file systems. This chapter presents a brief overview of the most popular Linux file systems, elaborating on their design concept, advantages, and fields of application. Some additional information about LFS (large file support) in Linux is also provided.

18.1 Glossary 378
18.2 Major File Systems in Linux 378
18.3 Some Other Supported File Systems 384
18.4 Large File Support in Linux 385
18.5 For More Information 386

18.1 Glossary

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 partly why the file systems show different performance characteristics. It is of major importance 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, date and time of creation, modification, and access, and pointers to the disk blocks where the file contents are actually stored.

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.

18.2 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 substitute for 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.

Note

Setting up File Systems

Unless stated otherwise in this chapter, all the steps required to set up or to change partitions and file systems can be performed using the YaST module.

Note

18.2.1 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. Yet, as 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.

18.2.2 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
As 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, JFS, 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, as the recreation of 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 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 (i.e., data integrity), but can slow down the system as 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 blocks are grouped as a "transaction" and 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.

Note

Converting an Ext2 File System into an Ext3 File System

Converting from Ext2 to Ext3 involves two separate steps:

Creating the Journal Log in as `root` and run `tune2fs -j`. 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 its manual page (`man 8 tune2fs`).

Specifying the File System Type in /etc/fstab To ensure that the Ext3 file system is recognized as such, edit the file `/etc/fstab`, changing the file system type specified for the corresponding partition from `ext2` to `ext3`. The change takes effect after the next reboot.

Using `ext3` for the Root Directory To boot a root file system set up as an `ext3` partition, include the modules `ext3` and `jbd` in the `initrd`. To do so, edit the file `/etc/sysconfig/kernel` to include the two modules under `INITRD_MODULES` then execute the command `mk_initrd`.

Note

18.2.3 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 SUSE LINUX version 6.4. ReiserFS was designed by Hans Reiser and the Namesys development team. ReiserFS has proven to be a powerful alternative to the old Ext2. Its key assets are better disk space utilization, better disk access performance, and faster crash recovery. However, there is a minor drawback: ReiserFS pays great care to metadata but not to the data itself. Future generations of ReiserFS will include data journaling (both metadata and actual data are written to the journal) as well as ordered writes.

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 as 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, you will often find that both file data and "stat_data" (inode) information are 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.

18.2.4 JFS

JFS, the *Journaling File System* was developed by IBM. The first beta version of the JFS Linux port reached the Linux community in the summer of 2000. Version 1.0.0 was released in 2001. JFS is tailored to suit the needs of high throughput server environments where performance is the ultimate goal. Being a full 64-bit file system, JFS supports both large files and partitions, which is another reason for its use in server environments.

A closer look at JFS shows why this file system might prove a good choice for your Linux server:

Efficient Journaling JFS follows a "metadata-only" approach like ReiserFS. Instead of an extensive check, only metadata changes generated by recent file system activity are checked, which saves a great amount of time in recovery. Concurrent operations requiring multiple concurrent log entries can be combined into one group commit, greatly reducing performance loss of the file system through multiple write operations.

Efficient Directory Organization JFS holds two different directory organizations. For small directories, it allows the directory's content to be stored directly into its inode. For larger directories, it uses B+trees, which greatly facilitate directory management.

Better Space Usage through Dynamic inode Allocation
For Ext2, you must define the inode density in advance (the space occupied by management information), which restricts the maximum number of files or directories of your file system. JFS spares these considerations — it dynamically allocates inode space and frees it when it is no longer needed.

18.2.5 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. As 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. A feature truly unique to XFS is *delayed allocation*. XFS 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 at

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 as the contents of a file are not distributed all over the file system.

18.3 Some Other Supported File Systems

Table 18.1 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 18.1: *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*: 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 file names) by creating special files.
`vfat`	*Virtual FAT*: extension of the `fat` file system (supports long file names).
`ntfs`	*Windows NT file system*, read-only.

18.4 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 18.2 offers an overview of the current limitations of Linux files and file systems for kernel 2.4.

***Table 18.2:** Maximum Sizes of File Systems (On-Disk Format)*

File System	**File Size [Byte]**	**File System Size [Byte]**
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} (16 TB)

Ext2 or Ext3 (8 kB block size) (systems with 8 kB pages, like Alpha)	2_{46} (64 TB)	2_{45} (32 TB)
ReiserFS 3.5	2_{32} (4 GB)	2_{44} (16 TB)
ReiserFS 3.6 (under Linux 2.4)	2_{60} (1 EB)	2_{44} (16 TB)
XFS	2_{63} (8 EB)	2_{63} (8 EB)
JFS (512 byte block size)	2_{63} (8 EB)	2_{49} (512 TB)
JFS (4 kB block size)	2_{63} (8 EB)	2_{52} (4 PB)
NFSv2 (client side)	2_{31} (2 GB)	2_{63} (8 EB)
NFSv3 (client side)	2_{63} (8 EB)	2_{63} (8 EB)

Note

Linux Kernel Limits

Table 18.2 on the page before 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 large. However, this limit is still out of reach for the currently available hardware.

Note

18.5 For More Information

Each of the file system projects described above maintains its own home page on which to find mailing list information, further documentation, and FAQs.

- `http://e2fsprogs.sourceforge.net/`
- `http://www.zipworld.com.au/~akpm/linux/ext3/`

- `http://www.namesys.com/`
- `http://oss.software.ibm.com/developerworks/opensource/jfs/`
- `http://oss.sgi.com/projects/xfs/`

A comprehensive multipart tutorial about Linux file systems can be found at *IBM developerWorks*: `http://www-106.ibm.com/developerworks/library/l-fs.html`. For a comparison of the different journaling file systems in Linux, look at Juan I. Santos Florido's article at *Linuxgazette*: `http://www.linuxgazette.com/issue55/florido.html`. Those interested in an in-depth analysis of LFS in Linux should try Andreas Jaeger's LFS site: `http://www.suse.de/~aj/linux_lfs.html`.

19

High availability under Linux

This chapter contains a short overview of the key concepts and tools from the area of high availability under Linux. It also offers suggested further reading for all the topics touched on.

19.1 What does high availability mean? 390
19.2 Important terms 390
19.3 A sample minimum scenario 391
19.4 Components of a high availability solution 392
19.5 The software side of high availability 394
19.6 Clustering 395
19.7 Further information 396

19.1 What does high availability mean?

High availability describes systems that can mask certain malfunctions – in particular the failure of individual computers – so that the service can be made available to the user again after only a short downtime. Hardware and software are carefully coordinated and laid out for redundancy, enabling an automatic switch to the other components in the event of a malfunction. High availability differs from "error tolerance" because the service is temporarily unavailable for the short service switchover phase, which can make itself noticeable in delays or short losses in connection.

When we refer to high availability systems, we particularly mean when the overall availability of the service is between 99.999% and 99.99999%; this corresponds to a downtime of between 5 minutes and 3 seconds over an entire year. The most important factor is not just the software and hardware side, but, primarily, well-conceived system administration with well-documented and understandable processes for minimizing faults. In every case, it involves the weighing up of risks/ costs; different requirements and solutions may be appropriate, depending on the application scenario. Your Novell partner will be happy to advise you.

19.2 Important terms

First, here are a few important terms that crop up again and again in relation to high availability:

SPoF *Single Point of Failure*: Components of a system whose failure impairs the functioning of the whole system.

Failover Another, similar system component automatically takes over the function of a failed component.

Cold Standby The alternative hardware is on "cold standby". The failover must be performed manually, so that the failure will be clearly apparent.

Warm Standby The backup system runs in the background, so that the transfer can take place automatically. The data on both systems is automatically synchronized. For the user, the failover is like a very fast automatic service reboot. However, the current transaction may be aborted because it was not possible to synchronize the data prior to failure.

Hot Standby Both systems permanently run in parallel – data on both systems is one hundred percent synchronized. Users will not be aware of any failures. This level cannot usually be reached without making a corresponding modification to the client. In order to run both systems with 100% synchrony, the connections to the client must be mirrored 100%. This normally requires clients that have connections with two or more servers at the same time and that communicate with all of them. A normal web browser cannot do this.

Load Balancing The distribution of load within a cluster of computers. *Load Balancing* is used in an LVS scenario (*Linux Virtual Server*) for example (see 19.6.2 on page 395).

STONITH *Shot the other node in the head*: Special hardware/ software that ensures that a faulty node does not write-access distributed media within a cluster, threatening data consistency in the entire cluster. This involves simply disconnecting the system from the mains power supply.

19.3 A sample minimum scenario

The procedures within a 2-node cluster when one node fails and the various types of standby systems that can take over as necessary are outlined below (see diagram 19.1 on the next page).

The two servers (Primary and Backup) are both connected to a SAN (Storage Area Network). Depending on the mode, this is only accessed by the active node. The servers communicate with each other in such a way that they regularly emit a "sign of life" (heartbeat). The communication channels (or *heartbeat links*) are also laid out in a redundant way, so that independent channels can be used by means of a variety of network cards and cable channels. If one of the links fails, then its backups continue to report correctly that the relevant server is still "alive". If there is no sign of life from the main system, the standby system is activated, so that it takes over the services of the failed partner and removes it from the network completely (see *STONITH* above).

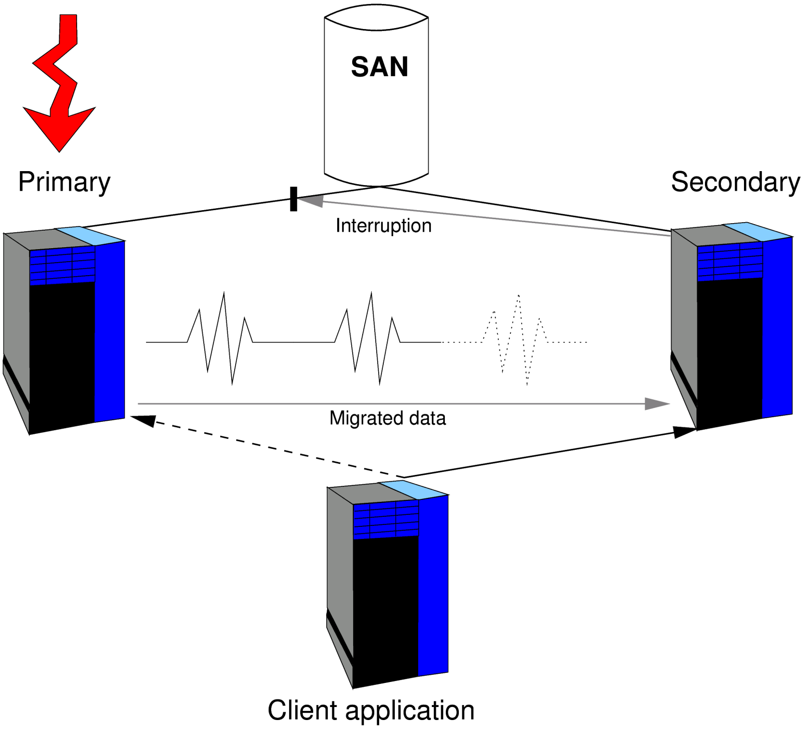


Figure 19.1: A simple high availability cluster

19.4 Components of a high availability solution

A high availability solution consists of several different components:

General infrastructure When designing a high availability solution, it should generally be remembered that even the installation of all key servers at a single location can be a potential *SPOF* if this location is hit by (environmental) disaster or power failures. The "environmental conditions" of the servers should also be taken into account – (redundant) air conditioning systems are essential.

Hardware Even the most sophisticated software cannot produce a high availability system without the greatest possible security from failure on a hardware level. The key hardware components that should be considered and laid out with the greatest possible redundancy are:

Power supply If possible, secure your servers using a UPS (uninterruptible power supply), so as to ensure that a brief power failure can be bridged and the systems can be shut down correctly in the event of a longer power failure. Of course the power supply should also be configured for redundancy.

Network interfaces Make sure that each of your systems has several network interfaces. If one interface fails, then another must automatically take over the address and task of the failed component. Redundancy expressly relates to the two interface directions. There is no harm planning an active and backup interface for both the internal and external interfaces.

Hard disks Assign several hard disks to your system and arrange the data backup (e.g. using RAID or drbd) in such a way that if one of these disks is lost, the others always contain the intact data record. It must be possible to replace a faulty disk with a new one without stopping the system.

Applications All important data and applications that form the outer face of your systems must be arranged in such a way that they will not prevent s restart. If an application does not release its lock files after a crash, this will prevent the relevant process from being restarted. This means that the application is not suitable for a high availability environment. Ideally, the “health” of certain applications, operating system processes and network connections should be monitored with a suitable monitoring tool.

Data After a system fails, all key data must be available to the failover system complete and intact. This type of high availability is achieved by distributing stored data over several systems or hard disks. For this, the contents of a disk are regularly mirrored on another disk (or several disks), which can take over with the intact data record if a failure occurs. You should use a journaling file system to ensure that a file system restarts in a consistent state after a system crash.

Network All network infrastructure should be configured for redundancy, from the router/ switch infrastructure down to the simple network cable.

19.5 The software side of high availability

The key software aspects of high availability solutions are described below:

19.5.1 heartbeat

heartbeat is a package that is used to monitor all the nodes used in the cluster. heartbeat exchanges "heartbeats" on the network interfaces of the members of the cluster in order to find out which nodes in the cluster are active. If a node fails, then it does not emit a signal. In this case, heartbeat ensures that another node will take over the relevant tasks and identity and makes the failover known within the network. This means that the cluster remains consistent. At present, the heartbeat failover function is limited to two nodes.

19.5.2 RAID

RAID (*Redundant Array of Independent Disks*) brings together several hard disk partitions to form a large *virtual* hard disk. RAID can be used to optimize the performance and data security of your system. RAID levels 1 and 5 offer protection against the failure of a disk because the data is recorded on several disks at the same time. This ensures that the complete data record is always available on another disk in the system should a disk fail. You will find more information about RAID under SUSE LINUX in 3.11 on page 139.

19.5.3 rsync

rsync can be used to enable large amounts of data to be synchronized between a server and its backup. rsync has sophisticated mechanisms for only transferring changes to files. This applies not only to text files, but also to binary files. To enable the differences between files to be identified, rsync divides the files into blocks and calculates checksums for these blocks. You will find more information about rsync in 23.6 on page 581.

19.5.4 DRBD

Distributed Replicated Block Device (drbd) enables the mirroring (RAID1) of partitions/ logical volumes (data areas) by means of a normal network on the basis of TCP/IP. Each node has a particular drbd resource "active", and all changes are mirrored as secure transactions.

drbd has additional features in comparison with RAID1 for local disks that enable the resynchronization time to be minimized after the two nodes have been disconnected briefly, as well as a robust check after various malfunctions to establish which side has the latest, consistent data.

19.6 Clustering

19.6.1 Cluster alias

The cluster alias is a technology that allows several nodes to be configured with a shared IP address, while also permitting TCP/IP connections to be established at this address. Inbound TCP/IP connections are automatically distributed.

Unlike the Linux virtual server, a dedicated load balancer is not required. However, because of the type of implementation, the cluster alias is less efficient when there is a large number of nodes: In the case of the cluster alias, all IP packages are distributed to all nodes, which then filter out the packages intended for them, while, in the case of LVS, this decision is only taken once by the load balancer. For further information on how to configure this feature see iptables in the manual pages.

19.6.2 Linux Virtual Server

Linux Virtual Server is based on a real cluster of several servers, which are connected together by means of a load balancer for distributing the payload among the various members of the cluster. From the outside, a cluster of this kind simply looks like a single virtual server. Naturally the load balancer should also be configured for redundancy and should be secured using heartbeat. The aim of an LVS configuration is to make the best possible use of the existing resources and to offer good scalability. The heartbeat-ldirectord daemon is used in these scenarios to monitor the "health" of the various real servers.

19.6.3 High availability clusters

High availability clusters are designed so that all available services can be provided at all times, despite hardware or software failures. If a node in a cluster fails, then another takes over immediately. This node (*secondary*) is a mirror image of the failed node (*primary*) and actually assumes the identity of the failed node during failover, so that the cluster environment remains externally consistent.

19.7 Further information

19.7.1 High availability (HA) in general and heartbeat

The primary source for information about high availability under Linux is the homepage of the *Linux-HA* project (`http://linux-ha.org`). This contains a wide range of tips and links to documentation, reports and scenarios.

For information in print about high availability see *Blueprints in High Availability*:

Marcus, Evan & Stern, Hal: *Blueprints in High Availability*. John Wiley & Sons Inc., 2000. - (ISBN 0-471-35601-8)

► **zseries, s390**
There is a very detailed redpaper for Linux on IBM S/390 and zSeries with a wide range of sample scenarios and configurations under: `http://publib-b.boulder.ibm.com/Redbooks.nsf/RedpaperAbstracts/redp0220.html`. ◄

A Wiki relating to HA and heartbeat: `http://wiki.trick.ca/linux-ha/PressRoom` and `http://wiki.trick.ca/linux-ha/HeartbeatTutorials`

19.7.2 DRBD

The homepage for the DRBD project: `http://www.drbd.org/`

An article in the Linux magazine:`http://www.linux-mag.com/2003-11/drbd_01.html`

19.7.3 RAID

A detailed collection of links relating to the topic of RAID: `http://linas.org/linux/raid.html`

19.7.4 Clustering

The *Linux Clustering Information Center* home page offers further information about clustering: `http://www.lcic.org/`

The home page for the *Linux Virtual Server* project: `http://www.linuxvirtualserver.org/`

You will find information about the *Oracle cluster file system* on the project home page under `http://oss.oracle.com/projects/ocfs/` and detailed documentation under: `http://oss.oracle.com/projects/ocfs/documentation/`.

PAM — Pluggable Authentication Modules

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 system-wide basis, so they can be requested by any application. This chapter describes how the modular authentication mechanism works and how it is configured.

20.1 Structure of a PAM Configuration File 400
20.2 The PAM Configuration of sshd 402
20.3 Configuration of PAM Modules 403
20.4 For More Information 406

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 or SAMBA) 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 to delegate the latter 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 that are used for authentication. In addition, there are global configuration files for most PAM modules under `/etc/security/`, which define the exact behavior of these modules (examples are `pam_env.conf`, `pam_pwcheck.conf`, `pam_unix2.conf`, `time.conf`, etc.). 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.

20.1 Structure of a PAM Configuration File

Each line in a PAM configuration file comprises 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 user name 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 to 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.

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, 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).

20.2 The PAM Configuration of sshd

To show how the theory behind PAM works, consider the PAM configuration of sshd as a practical example:

Example 20.1: *PAM Configuration for sshd*

```
#%PAM-1.0
auth required   pam_unix2.so # set_secrpc
auth required   pam_nologin.so
auth required   pam_env.so
account required        pam_unix2.so
account required        pam_nologin.so
password required       pam_pwcheck.so
password required       pam_unix2.so    use_first_pass use_authtok
session required        pam_unix2.so    none     # trace or debug
session required        pam_limits.so
# 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
```

sshd first calls the three modules of the `auth` type. The first one, `pam_unix2`, checks the user's login and password against `/etc/passwd` and `/etc/shadow`. The next module (`pam_nologin`) checks whether the file `/etc/nologin` exists. If it does, no user other than `root` may log in. The third module is `pam_env`, which loads the file `/etc/security/pam_env.conf` to set the environment variables as specified in the 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 whole stack of `auth` modules is processed before sshd gets any feedback about whether the login has succeeded or not. 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.

The next stack of modules includes all the `account` type modules, which check whether the user has general permission to use the requested service. This again involves the successful processing of the modules `pam_unix2` and `pam_nologin` (`required`). If `pam_unix2` returns the result that the user exists and if `pam_nologin` returns the result that the user may indeed log in, sshd receives a message about the success, after which the next module stack is processed.

The following two modules are of the `password` type and must also be successfully 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, which uses the CrackLib library to check whether the password is secure, warning the user if he has chosen a password which is lacking in any respect (too short, too simple). The previously used `pam_unix2` module carries over any old and new passwords from `pam_pwcheck`, so the user does not have 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.

As the final step, the modules of the `session` type 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. 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 user logs out.

20.3 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`.

20.3.1 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 from 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 20.2.

Example 20.2: *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 of pam_unix2.

20.3.2 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. It lets you 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 very common example for which the default should be overridden by `pam_env` is the `DISPLAY` variable, which is changed whenever a remote login takes place. See Example 20.3 on the next page.

Example 20.3: *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`. More information can be obtained from the comments in the file `/etc/security/pam_env.conf`.

20.3.3 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 20.4 is an example:

Example 20.4: *pam_pwcheck.conf*

```
password:    nullok blowfish use_cracklib
```

This tells `pam_pwcheck` to allow empty passwords and modification of passwords. It also tells the module to use the Blowfish algorithm for password encryption and to check passwords with CrackLib. More options for the module are mentioned in the file `/etc/security/pam_pwcheck.conf`.

20.3.4 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.

20.4 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 for SUSE LINUX and made available some information about them at: `http://www.suse.de/~kukuk/pam/`

Part III

Services

21 Linux in the Network

Linux, really a child of the Internet, offers all the necessary networking tools and features for integration into all types of network structures. An introduction into the customary Linux protocol, TCP/IP, follows. The various services and special features of this protocol are discussed. Network access using a network card can be configured with YaST. The central configuration files are discussed and some of the most essential tools described. Only the fundamental mechanisms and the relevant network configuration files are discussed in this chapter.

21.1 TCP/IP — The Protocol Used by Linux 410
21.2 IPv6 — The Next Generation Internet 418
21.3 Manual Network Configuration 428
21.4 Network Integration 437
21.5 Routing in SUSE LINUX 450
21.6 SLP services in the network 452
21.7 DNS — Domain Name System 455
21.8 LDAP — A Directory Service 483
21.9 NIS — Network Information Service 513
21.10 NFS — Shared File Systems 518
21.11 DHCP 522
21.12 Time Synchronization with xntp 535

21.1 TCP/IP — The Protocol Used by Linux

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. TCP/IP was developed based on an application used for military purposes and was defined in its present form in an RFC in 1981. RFC stands for *Request for Comments*. They are documents that describe various Internet protocols and implementation procedures for the operating system and its applications. Since then, the TCP/IP protocol has been refined, but the basic protocol has remained virtually unchanged.

Note

The RFC documents describe the setup of Internet protocols. To expand your knowledge about any of the protocols, refer to the appropriate RFC document. They are available online at `http://www.ietf.org/rfc.html`.

Note

The services listed in Table 21.1 are provided for the purpose of exchanging data between two Linux machines via TCP/IP. Networks combined by TCP/IP, comprising a world-wide network are also referred to, in their entirety, as “the Internet.”

Table 21.1: *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 already 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, a special echo mode is provided by ICMP that can be viewed using the program `ping`.
IGMP	Internet Group Management Protocol: This protocol controls the machine behavior when implementing IP multicast. The following sections do not contain more information regarding IP multicasting, because of space limitations.

Almost all hardware protocols work on a packet-oriented basis. The data to transmit is packaged in *packets*, as it cannot be sent all at once. This is why TCP/IP only works with small data packets. The maximum size of a TCP/IP packet is approximately 64 kilobytes. The packets are normally quite a bit smaller, as the network software 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.

21.1.1 Layer Model

IP (Internet Protocol) is where the insecure data transfer takes place. TCP (Transmission Control Protocol), to a certain extent, is simply the upper layer for the IP platform serving to guarantee secure data transfer. The IP layer itself is, in turn, supported by the bottom layer, the hardware-dependent protocol, such as ethernet. Professionals refer to this structure as the *layer model*. See Figure 21.1 on the following page.

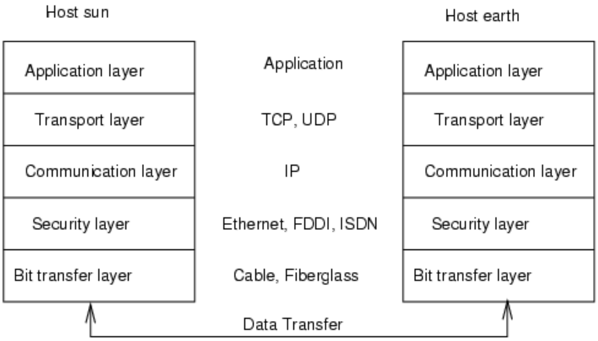

Host sun
Host earth
Application layer
Transport layer
Communication layer
Security layer
Bit transfer layer
Application
TCP, UDP
IP
Ethernet, FDDI, ISDN
Cable, Fiberglass
Application layer
Transport layer
Communication layer
Security layer
Bit transfer layer
Data Transfer

Figure 21.1: *Simplified Layer Model for TCP/IP*

The diagram provides one or two examples for each layer. As you can see, 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 bit transfer and security layers represent the physical network used (such as ethernet).

- While layer 1 deals with cable types, signal forms, signal codes, and the like, layer 2 is responsible for accessing procedures (which host may send data?) and error correction. Layer 1 is called the *physical layer*. Layer 2 is called the *data link layer*.
- Layer 3 is the *network layer* and is responsible for remote data transfer. The network layer ensures that the data arrives at the correct remote destination and can be delivered to it.
- Layer 4, the *transport layer*, is responsible for application data. It ensures that data arrives in the correct order and is not lost. While the data link layer is only there to make sure that the data as transmitted is the correct one, the transport layer protects it from being lost.
- Finally, layer 5 is the layer where data is processed by the application itself.

For every layer to serve its designated function, 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 21.2.

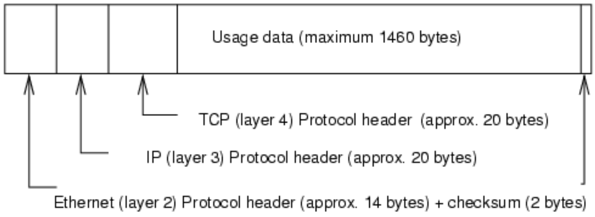

Usage data (maximum 1460 bytes)
TCP (layer 4) Protocol header (approx. 20 bytes)
IP (layer 3) Protocol header (approx. 20 bytes)
Ethernet (layer 2) Protocol header (approx. 14 bytes) + checksum (2 bytes)

Figure 21.2: *TCP/IP Ethernet Packet*

The proof sum is located at the end of the packet, not at the beginning. This simplifies things for the network hardware. The largest amount of usage data possible in one packet is 1460 bytes in an ethernet network.

When an application sends data over the network, the data passes through each layer, all implemented in the Linux kernel except layer 1 (network card). Each layer is responsible for preparing the data so it can be passed to the next layer below. 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, layer 4 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 being transmitted, as long as packets are in the correct format.

21.1.2 IP Addresses and Routing

Note

The discussion in the following sections is limited to IPv4 networks. For information about IPv6 protocol, the successor to IPv4, refer to Section 21.2 on page 418.

Note

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 Table 21.1.

Example 21.1: *How an IP Address is Written*

```
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 certainly exceptions to this rule, but these play a minimal role in the following passages.

The ethernet card itself has its own unique address, the *MAC*, or media access control address. It is 48 bits long, internationally unique, and is programmed into the hardware by the network card vendor. There is, however, an unfortunate disadvantage of vendor-assigned addresses — *MAC* addresses do not make up a hierarchical system, but are instead more or less randomly distributed. Therefore, they cannot be used for addressing remote machines. The *MAC* address still plays an important role in communication between hosts in a local network and is the main component of the protocol header of layer 2.

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 so was discontinued. Now, *classless routing* (or CIDR, Classless Inter Domain Routing) is used.

Netmasks and Routing

Netmasks were conceived for the purpose of informing the host with the IP address 192.168.0.0 of the location of the host with the IP address 192.168.0.20. To put it simply, the netmask on a host with an IP address defines what is internal and what is external. Hosts located internally (professionals say, "in the same subnetwork") respond directly. Hosts located externally ("not in the same subnetwork") only respond via a gateway or router. Because every network interface can receive its own IP address, it can get quite complicated.

Before a network packet is sent, the following runs on the computer: the IP address is linked to the netmask via a logical AND and the address of the sending host is likewise connected to the netmask via the logical AND. If there are several network interfaces available, normally all possible sender addresses are verified. The results of the AND links will be compared. If there are no discrepancies in this comparison, the destination, or receiving host, is located in the same subnetwork. Otherwise, it must be accessed via a gateway. The more "1" bits are located in the netmask, the fewer hosts can be accessed directly and the more hosts can be reached via a gateway. Several examples are illustrated in Table 21.2.

Example 21.2: *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
```

The netmasks appear, like IP addresses, in decimal form divided by periods. Because the netmask is also a 32-bit value, four number values are written next to each other. Which hosts are gateways or which address domains are accessible over which network interfaces must be entered in the user configuration.

To give another example: all machines connected with the same ethernet cable are usually located in the same subnetwork and are directly accessible. When the ethernet is divided by switches or bridges, these hosts can still be reached.

However, the economical ethernet is not suitable for covering larger distances. You must transfer the IP packets to another hardware (such as FDDI or ISDN). Devices for this transfer are called routers or gateways. A Linux machine can carry out this task. The respective option is referred to as `ip_forwarding`.

If a gateway has been configured, the IP packet is sent to the appropriate gateway. This then attempts to forward the packet in the same manner — from host to host — until it reaches the destination host or the packet's TTL (time to live) has expired.

Table 21.2: *Specific Addresses*

Address Type	Description
Base network address	This is the netmask AND any address in the network, as shown in Table 21.2 on the page before 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 strictly assigned to the "loopback device" on each host. A connection can be set up to your own machine with this address.

As IP addresses must be unique all over the world, you cannot just come up with your own random addresses. There are three address domains to use to set up a private IP-based network. With these, you cannot set up any connections to the rest of the Internet, unless you apply certain tricks, because these addresses cannot be transmitted over the Internet. These address domains are specified in RFC 1597 and listed in Table 21.3 on the facing page.

Table 21.3: 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`

21.1.3 Domain Name System

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 `laurent.suse.de`, written in the format `hostname.domain`. A full name, referred to by experts as a *fully qualified domain name*, or FQDN for short, consists of a host name and a domain name (`suse.de`). The latter also includes the *top level domain* or TLD (`de`).

TLD assignment has become, for historical reasons, quite confusing. 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, multiletter 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 host names 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, or NIC. Each root name server knows about the name servers responsible for a given top level domain. More information about top level domain NICs is available at `http://www.internic.net`.

DNS can do more than just resolve host names. 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 is described in Section 21.7 on page 455.

whois

The protocol `whois` is closely related to DNS. With this program, quickly find out who is responsible for any given domain.

21.2 IPv6 — The Next Generation Internet

Note

s390, zseries: IPv6 Support

IPv6 is not supported bz the CTC and IUCV network connections of the IBM S/390 and zSeries hardware.

Note

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 last ten 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 100 million.

As mentioned, an IP 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 the number of bits squared minus two. A subnetwork has, for example, two, six, or fourteen addresses available. To connect 128 hosts to the Internet, for instance, 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 quite a chore to set up and a burden to maintain. To set up a host in an IPv4 network, you need to find out about quite a number of address items, such as the host's own IP address, the subnetmask, the gateway address, and maybe a name server address. In fact, all these items need to be known, meaning they 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.

21.2.1 Advantages of IPv6

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 21.2.2 on the following page.

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 autoconfig 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 will be impossible to switch the entire Internet from IPv4 to IPv6 in one fell swoop. 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 on the one hand (IPv4 addresses can easily be translated into IPv6 addresses) and through the use of a number of tunnels on the other (see Section 21.2.3 on page 426). 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 instance, or all routers (the *all routers multicast group*).

21.2.2 The IPv6 Address System

As mentioned, the current IP protocol is lacking in two important aspects: on the one hand, there is an increasing shortage of IP addresses; on the other hand, configuring the network and maintaining the routing tables is becoming a more and 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 allocate 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.

Structure of an IPv6 Address

An IPv6 address is made up of eight four-digit fields, each of them representing sixteen 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 Output 21.3, where all three lines represent the same address.

***Example 21.3:** 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 Output 21.4 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 21.4: *IPv6 Address Specifying the Prefix Length*

```
fe80::10:1000:1a4/64
```

IPv6 knows about several predefined types of prefixes, some of which are shown in Table 21.4.

Table 21.4: *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 are not supposed to 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 (e.g., `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 the packet shall be delivered.

Interface ID The third part identifies the interface to which the packet shall be delivered. 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.

► **s390, zseries**
Devices without MAC addresses on IBM S/390 and zSeries are IUCV and CTC (point-to-point). ◄

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 21.2.3 on the next page) 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 are not supposed to 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 null bytes. Addresses of this type are used during autoconfiguration 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 as defined by IPv4. They contain a special prefix (`fec0::/10`), the interface ID, and a sixteen bit field specifying the subnetwork ID. Again, the rest is filled with null 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 in completely automatic fashion, 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, no matter 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 further 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.

21.2.3 Coexistance of IPv4 and IPv6

The migration of all hosts connected to the Internet from IPv4 to IPv6 will be 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 how an IPv6 enabled host is supposed to communicate with an IPv4 host and how IPv6 packets should be transported by the current networks, which are predominantly IPv4 based. The best solution offer tunneling and compatibility addresses (see Section 21.2.2 on page 422).

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 it 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.

Note

The 6bone Initiative

In the heart of the “old-time” Internet, there is already a globally distributed network of IPv6 subnets that are connected through tunnels. This is the *6bone* network (`www.6bone.net`), an IPv6 test environment that may be used by programmers and Internet providers who want to develop and offer IPv6-based services to gain the experience necessary to implement the new protocol. More information can be found on the project’s Internet site.

Note

21.2.4 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.ngnet.it/e/cosa-ipv6.php` An article series providing a well-written introduction to the basics of IPv6. A good primer on the topic.

`http://www.bieringer.de/linux/IPv6/` Here, find the Linux IPv6-HOWTO and many links related to the topic.

`http://www.6bone.net/` Visit this site if you want to join a tunneled IPv6 network.

`http://www.ipv6.org/` The starting point for everything about IPv6.

RFC 2640 The fundamental RFC about IPv6.

IPv6 Essentials A book describing all the important aspects of the topic. Silvia Hagen: *IPv6 Essentials*. O'Reilly & Associates, 2002 (ISBN 0-596-00125-8).

21.3 Manual Network Configuration

Manual configuration of the network software should always be the last alternative. Using YaST is recommended. All network interfaces are activated with the script `/sbin/ifup`. To halt the interface, use `ifdown`. To check its status, use `ifstatus`.

If you only use internal network cards, simply configure the interfaces by means of their names. With the commands `ifup eth0`, `ifstatus eth0`, and `ifdown eth0`, start, check, or stop the interface `eth0`. The respective configuration files are stored in `/etc/sysconfig/network/ifcfg-eth0`. `eth0` is the name of the interface and the name of the configuration.

Alternatively, configure the network in relation to the hardware address (MAC address) of a network card. In this case, use a hardware-based configuration file named in the format `ifcfg-<hardwareaddresswithoutcolons>`. Use lowercase characters in the hardware address, as displayed by the command `ip link` (`ifconfig` shows uppercase letters). If `ifup` finds a configuration file matching the hardware address, a possibly existing file `ifcfg-eth0` will be ignored.

Things are a little more complicated with hotplug network cards, however. If you do not use one of those cards, proceed directly to Section 21.3.1 on the next page.

Hotplug network cards do not get a static interface name assigned, therefore the configuration for one of those cards cannot be stored under the name of the interface. Instead, a name is used that contains the kind of hardware and the connection point. In the following, this name is referred to as the hardware description. `ifup` must be started with two arguments — the hardware description and the current interface name. `ifup` then determines the configuration that best fits the hardware description.

Note

Hotpluggable Network Cards on IBM S/390 and zSeries

As a general rule, network cards on IBM S/390 and zSeries are hotpluggable. Unlike PCMCIA cards on PCs, the automatic configuration via DHCP is not possible on these platforms. The cards are detected but must be configured manually. The following example deals exclusively with hardware with PCMCIA support.

Note

For example, consider a laptop with two PCMCIA slots, a PCMCIA ethernet network card, and an internal network card configured as `eth0`. If the internal card is in slot `0`, its hardware description is `eth-pcmcia-0`. The cardmgr or the hotplug network script runs the command `ifup eth-pcmcia-0 eth1`. `ifup` searches `/etc/sysconfig/network/` for the file `ifcfg-eth-pcmcia-0`. If this file does not exist, it consecutively searches for `ifcfg-eth-pcmcia`, `ifcfg-pcmcia-0`, `ifcfg-pcmcia`, `ifcfg-eth1`, and `ifcfg-eth`. The first of these files found by `ifup` is used for the configuration. To generate a network configuration valid for all PCMCIA network cards in all slots, the configuration file must be named `ifcfg-pcmcia`. This file would be used for the ethernet card in slot 0 (`eth-pcmcia-0`) as well as for a token ring card in slot 1 (`tr-pcmcia-1`).

YaST numbers the configurations for hotplug cards and writes the corresponding settings to `ifcfg-eth-pcmcia-<number>`. To use such a configuration file for all slots, `ifcfg-eth-pcmcia` is linked to this file. Keep this in mind if you configure the network sometimes with and sometimes without YaST.

21.3.1 Configuration Files

This section provides an overview of the network configuration files and explains their purpose and the format used.

/etc/syconfig/hardware/*

This directory contains a separate file for every device (network card). These files contain the configuration parameters (kernel module, start mode, script assocations, etc.).

/etc/sysconfig/network/ifcfg-*

These files contain data specific to a network interface. They may be named after the network interface (`ifcfg-eth2`), the hardware address of a network card (`ifcfg-000086386be3`), or the hardware description (`ifcfg-usb`). If network aliases are used, the respective files are named `ifcfg-eth2:1` or `ifcfg-usb:1`. The script `ifup` gets the interface name and, if necessary, the hardware description as arguments then searches for the best matching configuration file.

► **s390, zseries**
IBM S/390 and zSeries do not support USB. The names of the interface files and network aliases contain S/390-specific elements like `qeth`. ◄

The configuration files contain the IP address (BOOTPROTO="static", IPADDR="10.10.11.214") or the direction to use DHCP (BOOTPROTO="dhcp"). The IP address should already contain the netmask (IPADDR="10.10.11.214/16"). Refer to `man ifup` for the complete list of variables. In addition, all the variables in the files `dhcp`, `wireless`, and `config` can be used in the `ifcfg-*` files, if a general setting should only be used for one interface. By using the variables `POST_UP_SCRIPT` and `PRE_DOWN_SCRIPT`, individual scripts can be run after starting or before stopping the interface.

/etc/sysconfig/network/config, dhcp, wireless

The file `config` contains general settings for the behavior of `ifup`, `ifdown`, and `ifstatus`. `dhcp` contains settings for DHCP and `wireless` for wireless lan cards. The variables in all three configuration files are commented and can also be used in `ifcfg-*` files, where they are treated with higher priority.

/etc/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 (keyword `nameserver`) to access. 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`. Comments are preceded by # signs.

Example 21.5: `/etc/resolv.conf`

```
# Our domain
search example.com
#
# We use sun (192.168.0.20) as nameserver
nameserver 192.168.0.20
```

An example of `/etc/resolv.conf` is shown in File 21.5. YaST enters the specified name server here. 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 by which it has been modified, 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 where introduced. Services that may need this flexibility include isdn, pcmcia, and hotplug.

If it happens that 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` (e.g., 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 will subsequently warn the user that changes will be lost after restoring the file. Apart from this, YaST will 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 above-mentioned services are only temporary.

/etc/hosts

In this file (see File 21.6), IP addresses are assigned to host names. If no name server is implemented, all hosts to which an IP connection will be set up must be listed here. For each host, a line consisting of the IP address, the fully qualified host name, and the host name (e.g., `earth`) is entered into the file. The IP address must be at the beginning of the line, the entries divided by blanks and tabs. Comments are always preceded by the # sign.

Example 21.6: `/etc/hosts`

```
127.0.0.1 localhost
192.168.0.20 sun.example.com sun
192.168.0.0 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 File 21.7).

Example 21.7: `/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 the libc4 or the 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 21.5 on the next page shows the parameters available. An example for `/etc/host.conf` is shown in File 21.8 on the facing page.

Table 21.5: *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*: Via 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 host name after host name resolution (as long as the host name 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 21.8: `/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 `man 5 nsswitch.conf` and *The GNU C Library Reference Manual* for more details.

The order for queries is defined in the file `/etc/nsswitch.conf`. An example of `nsswitch.conf` is shown in File 21.9 on the next page. 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 Section 21.7 on page 455).

Example 21.9: `/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 21.6 on the next page. In addition, `automount`, `bootparams`, `netmasks`, and `publickey` are expected in the near future. The configuration options for NSS databases are listed in Table 21.7 on the facing page.

Table 21.6: 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 host names 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 `man 5 netgroup`.
`networks`	Network names and addresses, used by `getnetent`.
`passwd`	User passwords, used by `getpwent`; see `man 5 passwd`.
`protocols`	Network protocols, used by `getprotoent`; see `man 5 protocols`.
`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 `man 5 shadow`.

Table 21.7: Configuration Options for NSS Databases

`files`	directly access files, for example, to `/etc/aliases`
`db`	access via a database
`nis`	NIS, see also Section 21.9 on page 513
`nisplus`	
`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 `man 8 nscd` and `man 5 nscd.conf`). By default, the system entries of `passwd` and `groups` are cached by nscd. `hosts` is not cached by default, because the mechanism in nscd to cache hosts causes the local system to be 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. This waiting time can be reduced by restarting nscd with the command `rcnscd restart`.

/etc/HOSTNAME

Here is the host name 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 host name is set.

21.3.2 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* (see also Table 21.8).

Table 21.8: *Some Start-Up Scripts for Network Programs*

`/etc/init.d/network`	This script handles the configuration of the network hardware and software when the system is booted.
`/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/sendmail`	Controls the sendmail process.
`/etc/init.d/ypserv`	Starts the NIS server.
`/etc/init.d/ypbind`	Starts the NIS client.

21.4 Network Integration

Currently TCP/IP is the standard network protocol by which all modern operating systems can communicate. Nevertheless, Linux also supports other network protocols, such as the IPX protocol (formerly) used by Novell Netware or the Appletalk protocol used by Macintosh machines. This chapter merely focuses on the integration of a Linux host in a TCP/IP network. To integrate arcnet, token ring, or FDDI network cards, refer to the kernel source documentation in `/usr/src/linux/Documentation` (package `kernel-source`).

21.4.1 Requirements

The machine must have a supported network card. Normally, the network card is detected during the installation and a suitable driver is loaded. To see if your card has been integrated correctly with the appropriate driver, enter the command `ifstatus eth0`. The output should list all information about the network device `eth0` or display an error message.

► **s390, zseries**
On IBM's S/390 and zSeries platforms, other possible device names (apart from `eth0`) include `hsi0`, `ctc0` and `iuvc0`. ◄

If the kernel support for the network card is implemented as a module, which is done by default for the SUSE kernel, the name of the module must be entered as an alias in `/etc/modules.conf`. This is done automatically when the driver support for the network card is loaded in linuxrc during the first installation. This task can also be done after installation with YaST.

If you are using a hotplug network card (e.g., PCMCIA or USB), the drivers are autodetected when the card is plugged in. No configuration is necessary.

Note

s390, zseries: Hotpluggable Network Cards

On IBM's S/390 and zSeries platforms, hotpluggable network cards are supported, but not their automatic network integration via DHCP (as is the case on the PC). After detection, you still have to manually configure the interface.

Note

21.4.2 Configuring the Network Card with YaST

Note

s390, zseries: Configuring a Network Card

Basically the procedure on IBM's S/390 and zSeries platforms is not different from the one described below. What *is* different though, are the types of network devices available for selection, and also some additional options (which correspond to the network options for these platforms as mentioned in the Chapter *Types of Network Devices* of the Architecture-specific Information manual.)

Note

After starting the module, YaST displays a general network configuration dialog. The upper part shows a list with all the network cards yet to be configured. Any card properly autodetected during the boot procedure will be listed with its name. Devices that could not be detected are listed as 'Other (not detected)'. In the lower part, the dialog displays a list of the devices configured so far, with their network type and address. You can now configure a new network card, or change an existing configuration.

Manual Configuration of a Network Card

The configuration of a network card that was not autodetected includes the following items:

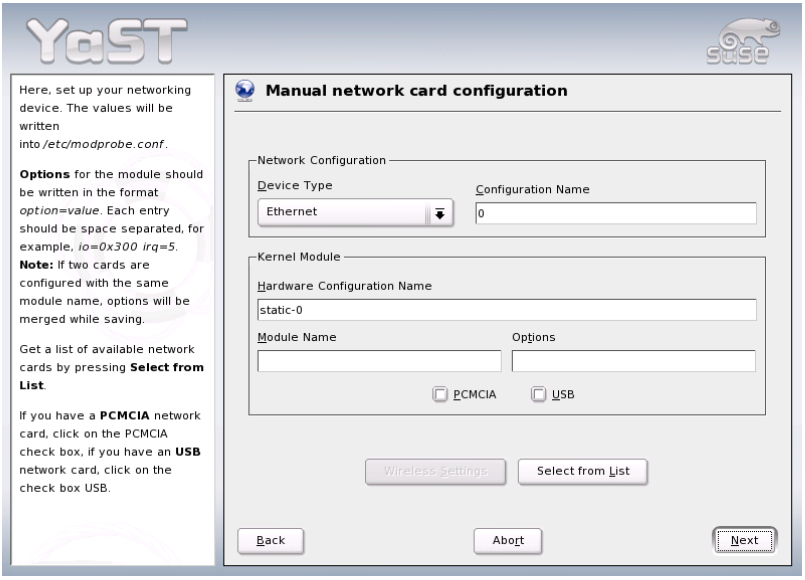
Device Type Specify the type of network device and the device number.

Wireless Support If you are within reach of a wireless network and your network card is designed for this connection type, use the 'Wireless Settings' button to open the dialog 'Wireless Network Card Configuration', where you can set the operating mode, the network name (ESSID), the network identifier (NWID), the encryption key, and a nickname. After setting these options, close the dialog with 'OK'.

► **s390, zseries**
Wireless LAN devices are no supported on IBM's S/390 and zSeries platforms. ◄

Kernel Module and Selection of Network Card
If your network card is a PCMCIA or USB device, enable the corresponding checkboxes and then leave the dialog by selecting 'Next'. Otherwise, use the button 'Select from List' and then specify your

network card. YaST will automatically load the appropriate driver for the selected card. Leave this dialog by selecting 'Next'.



Here, set up your networking device. The values will be written into */etc/modprobe.conf*.

Options for the module should be written in the format *option=value*. Each entry should be space separated, for example, *io=0x300 irq=5*. **Note:** If two cards are configured with the same module name, options will be merged while saving.

Get a list of available network cards by pressing **Select from List**.

If you have a **PCMCIA** network card, click on the PCMCIA check box, if you have an **USB** network card, click on the check box USB.

Device Type: Ethernet
Configuration Name: 0
Hardware Configuration Name: static-0
Module Name:
Options:
☐ PCMCIA ☐ USB

Figure 21.3: *Configuration of the Network Card*

Setting the Network Address

This lets you specify how the address should be assigned to your network card:

'Automatic address setup (via DHCP)'

If your network includes a DHCP server, you can rely on it to set up your network address automatically. The option should also be used if you are using a DSL line but with no static IP assigned by the ISP. If you decide to use DHCP, you can configure the details after selecting the button 'DHCP Client Options'. Specify whether the DHCP server should always honor broadcast requests, and any identifier that you want to use. By default, DHCP servers use the card's hardware address to identify an interface. But if you have a virtual host setup where different hosts communicate through the same interface, an identifier is necessary to distinguish between them.

► **s390, zseries**
On IBM's S/390 and zSeries platforms, DHCP based address configuration is only supported with network cards having a MAC address. This is only the case with OSA and OSA Express cards. ◄

'Static Address Setup' If your have a static address, enable the corresponding checkbox. Then enter the address and subnet mask for your network. The preset subnet mask should match the requirements of a typical home network.

You can now leave this dialog by selecting 'Next', or proceed to configure the host name, name server and routing details (see Section 2.6.2 on page 84 and Section 2.6.7 on page 85).

Cable Modem

Note

s390, zseries: Cable Modem

The configuration of this type of hardware is not supported on IBM's S/390 and zSeries platforms.

Note

In some countries (Austria, US), it is quite common to access the Internet through the TV cable network. The TV cable subscriber will usually get a modem which 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'. Nowadays most providers use DHCP, while a static IP address often comes as part of a special deal offered to businesses.

For further information about the configuration of cable modems, you are encouraged to read the Support Database article on the topic, which is available online at `http://sdb.suse.de/en/sdb/html/cmodem8.html`.

21.4.3 Modem

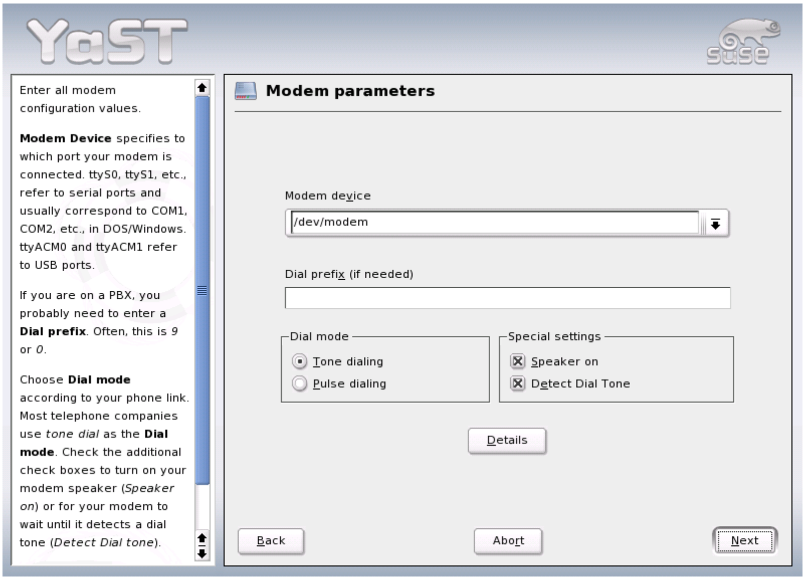
Note

s390, zseries: Modem

The configuration of this type of hardware is not supported on IBM's S/390 and zSeries platforms.

Note

In the YaST Control Center, you can access the modem configuration under the 'Network/Basics' item. If your modem was not autodetected, open the dialog for manual configuration. In the dialog that opens, enter the interface to which the modem is connected under 'Modem device:'.



Enter all modem configuration values.

Modem Device specifies to which port your modem is connected. ttyS0, ttyS1, etc., refer to serial ports and usually correspond to COM1, COM2, etc., in DOS/Windows. ttyACM0 and ttyACM1 refer to USB ports.

If you are on a PBX, you probably need to enter a **Dial prefix**. Often, this is *9* or *0*.

Choose **Dial mode** according to your phone link. Most telephone companies use *tone dial* as the **Dial mode**. Check the additional check boxes to turn on your modem speaker (*Speaker on*) or for your modem to wait until it detects a dial tone (*Detect Dial tone*).

Modem device: /dev/modem

Dial prefix (if needed):

Dial mode
● Tone dialing
○ Pulse dialing

Special settings
☒ Speaker on
☒ Detect Dial Tone

Figure 21.4: *Modem Configuration*

If you are behind a private branch exchange (PBX), you may need to enter a dial prefix (this is often a zero; consult the instructions that came with the PBX to find out). You can also select whether to use tone or pulse dialing, whether the loudspeaker should be on or off, and whether the modem should wait until it detects a dial tone. The latter option should not be enabled if the modem is connected to an exchange.

Under 'Details', you can set the baud rate and the modem initialization strings. You should only change these settings if your modem was not autodetected, or if it requires special tweaks to make the data transmission work. This is mainly the case with ISDN terminal adapters. Leave this dialog by selecting 'OK'.

In the next dialog, select the ISP (Internet service provider). To choose from a pre-defined list of IPSs operating in your country, select the 'Countries' radiobutton. Alternatively, select the 'New' button to open a dialog where you can provide the data for your own ISP. This includes a name for the dialout connection and for the ISP, as well the login and the password as provided by your ISP. Enable the checkbox 'Always Ask for Password', if you prefer to be prompted for the password each time you dial out.

The last dialog allows you to specify additional connection options:

'Dial on Demand' If you enable the dial on demand option, you need to specify at least one name server.

'Modify DNS when Connected' This checkbox is enabled by default, with the effect that the name server address is updated each time you connect to the Internet. However, if you enable 'Dial on Demand', you should disable this and also provide a fixed name server address.

'Stupid Mode' This option is enabled by default. It has the effect that input prompts sent by the ISP's dial-in server are ignored, to prevent it from interfering with the connection process.

'Activate Firewall' Selecting this option enables the SUSE firewall, which protects you from outside attacks for the time of your Internet connection.

'Idle Time-out (seconds)' This option lets you specify a period of network inactivity after which the modem will hang up automatically.

IP Details This button takes you to the address configuration dialog. If your ISP does not assign a dynamic IP address to your host, disable the checkbox 'Dynamic IP address' and then enter your host's local IP address as well as the remote IP address. If you do not know them, ask your ISP for information. Leave the 'Defaultroute' option enabled and close the dialog by selecting 'OK'.

Selecting 'Next' takes you to the original dialog, which will display a summary of the modem configuration. You can close this dialog with 'Finish'.

21.4.4 DSL

Note

s390, zseries: DSL

The configuration of this type of hardware is not supported on IBM's S/390 and zSeries platforms.

Note

To configure your DSL device, select the 'DSL' module from the YaST 'Network Devices' section. This YaST module comprises several dialogs which allow you to set the parameters of DSL links based on one of the following protocols:

- PPP over Ethernet (PPPoE) - Germany
- PPP over ATM (PPPoATM) - England
- CAPI for ADSL (Fritz cards)
- Point-to-Point Tunneling Protocol (PPTP) - Austria

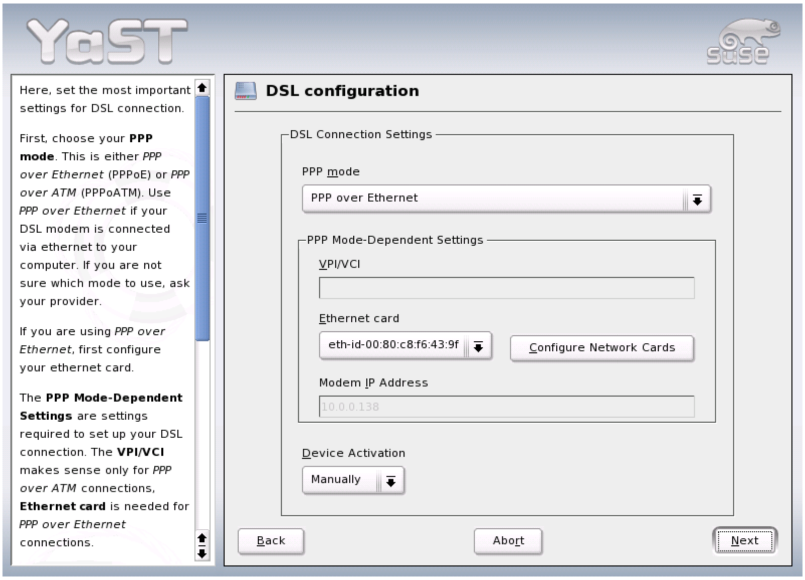
The configuration of a DSL connection based on PPPoE or PPTP requires that the corresponding network card has already been set up in the correct way. If you have not done so yet, first configure the card by selecting 'Configure Network Cards' (see Section 21.4.2 on page 438). In the case of a DSL link, 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, for instance `192.168.22.1`. In the 'Subnet mask' field, enter `255.255.255.0`. If you are configuring a stand-alone workstation, make sure to leave the 'Default Gateway' field empty.

Note

Values in the 'IP Address' and 'Subnet mask' fields are only placeholders. They are only needed to initialize the network card and do not represent the DSL link as such.

Note

To begin with the DSL configuration as such (see Figure 21.5 on the following page), first select the PPP mode and the Ethernet card to which the DSL modem is connected (in most cases this should be `eth0`). Then use



Here, set the most important settings for DSL connection.

First, choose your **PPP mode**. This is either *PPP over Ethernet* (PPPoE) or *PPP over ATM* (PPPoATM). Use *PPP over Ethernet* if your DSL modem is connected via ethernet to your computer. If you are not sure which mode to use, ask your provider.

If you are using *PPP over Ethernet*, first configure your ethernet card.

The **PPP Mode-Dependent Settings** are settings required to set up your DSL connection. The **VPI/VCI** makes sense only for *PPP over ATM* connections, **Ethernet card** is needed for *PPP over Ethernet* connections.

DSL configuration

DSL Connection Settings

PPP mode: PPP over Ethernet

PPP Mode-Dependent Settings

VPI/VCI:

Ethernet card: eth-id-00:80:c8:f6:43:9f

Modem IP Address: 10.0.0.138

Device Activation: Manually

Figure 21.5: DSL Configuration

the combobox labeled 'Device Activation' to specify whether you want the DSL link to be established during the boot process or at some later point (for instance manually). The dialog also lets you select your country and allows you to choose from a number of ISPs operating in it. The details of any subsequent dialogs of the DSL configuration will depend a lot on the options set so far, which is why they are just briefly touched upon in the paragraphs below. For details on each of the available options, read the detailed help available from the dialogs.

To be able to use 'Dial on Demand' on a stand-alone workstation, you also have to specify the name server (DNS server). Most ISPs nowadays support dynamic DNS, which is to say that the IP address of a name server is communicated by the ISP each time you connect. For a single workstation, however, you still need to provide a placeholder address in the DNS server field, for instance `192.168.22.99`. If your ISP does not support dynamic DNS, enter the IP address(es) of the name server(s) that the ISP has given to you.

Another interesting option can be set using the 'Idle Time-out (seconds)' combobox. This allows you to define a period of network inactivity after which the connection will be automatically terminated. A reasonable time-out value should be somewhere between 60 and 300 seconds.

Note

Dial on Demand

If you enable 'Dial on Demand' in addition to the option mentioned above, the connection will not be completely terminated after the time-out. Instead, the connection will remain in a standby mode, and gets re-established automatically as soon as a program requests some kind of data traffic. On the other hand, if 'Dial on Demand' is disabled, the connection is completely terminated and must by re-established manually whenever there is a need for it. It may then be useful to set the time-out to zero, which has the effect that an automatic hang-up will not take place at all.

Note

The configuration of T-DSL is very similar to the DSL setup. Just select 'T-Online' as your provider, and YaST will take you to the T-DSL configuration dialog. In this dialog you need to provide some additional information required for T-DSL, namely 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.

21.4.5 ISDN

Note

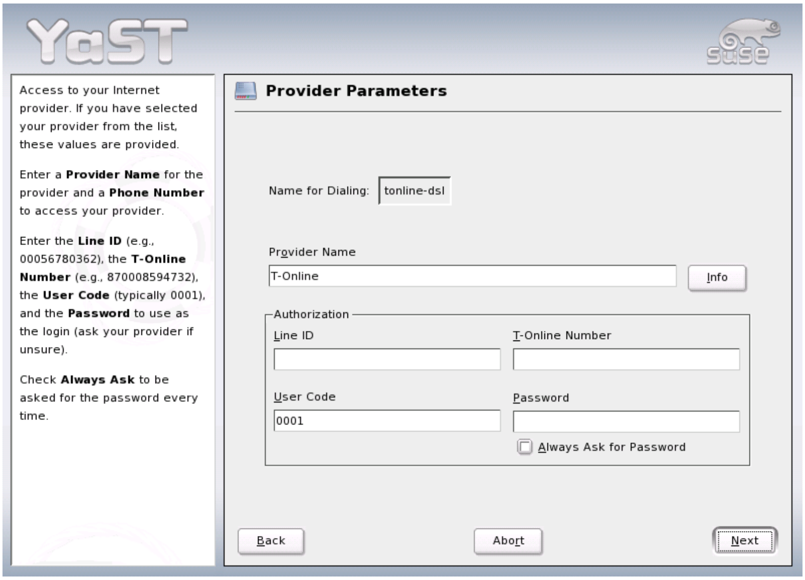
s390, zseries: ISDN

The configuration of this type of hardware is not supported on IBM's S/390 and zSeries platforms.

Note

This module allows you to configure one or several ISDN cards for your system. If YaST did not autodetect your ISDN card, you first need to manually select it. Though it is well possible to set up several ISDN interfaces, most home users will not need to do so, because several ISPs can be configured for one interface. The subsequent dialogs allow you to set the ISDN options necessary for the proper functioning of the card.

The next dialog (see Figure 21.7 on page 447) lets you select the protocol ('Selection of ISDN protocol'). The default is 'Euro-ISDN (EDSS1)' (see



Access to your Internet provider. If you have selected your provider from the list, these values are provided.

Enter a **Provider Name** for the provider and a **Phone Number** to access your provider.

Enter the **Line ID** (e.g., 00056780362), the **T-Online Number** (e.g., 870008594732), the **User Code** (typically 0001), and the **Password** to use as the login (ask your provider if unsure).

Check **Always Ask** to be asked for the password every time.

Name for Dialing: tonline-dsl

Provider Name: T-Online

Line ID:

T-Online Number:

User Code: 0001

Password:

☐ Always Ask for Password

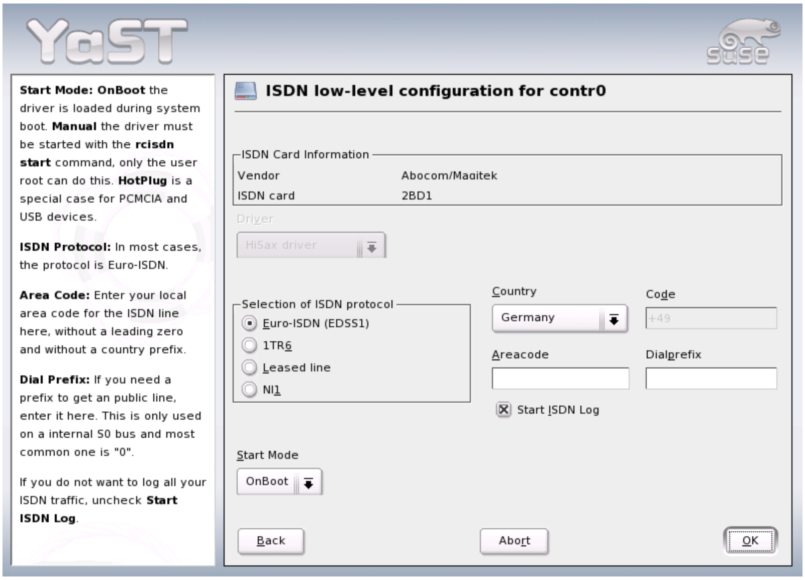
Figure 21.6: *T-DSL Configuration (Germany)*

points 1. and 2.a below), but for older or larger exchanges, you need to select '1TR6' (see point 2.b below). If you are in the US, select 'NI1'. To select your country, use the combobox provided; the corresponding country code will then be filled into the field next to it. Finally provide your 'Areacode', together with the dial prefix (if necessary).

The 'Start Mode' combobox lets you define how the ISDN interface should be started: 'OnBoot' causes the ISDN driver to be initialized each time during the boot process, while 'Manual' requires you to load the ISDN driver as `root` with the command `rcisdn start`. 'Hotplug' is for PCMCIA or USB devices, in which case the driver gets loaded after the device is plugged in. When you are finished with all these settings, select 'OK'.

The next dialog allows you to specify the interface type for your ISDN card, and to add ISPs to an existing interface. Interfaces may be either of the `SyncPPP` or the `RawIP` type, though most ISP operate in the `SyncPPP` mode, which is described below.

The number to be entered under 'My phone number' will vary depending on your particular setup:



Start Mode: OnBoot the driver is loaded during system boot. **Manual** the driver must be started with the **rcisdn start** command, only the user root can do this. **HotPlug** is a special case for PCMCIA and USB devices.

ISDN Protocol: In most cases, the protocol is Euro-ISDN.

Area Code: Enter your local area code for the ISDN line here, without a leading zero and without a country prefix.

Dial Prefix: If you need a prefix to get an public line, enter it here. This is only used on a internal S0 bus and most common one is "0".

If you do not want to log all your ISDN traffic, uncheck **Start ISDN Log**.

ISDN low-level configuration for contr0

ISDN Card Information
Vendor: Abocom/Magitek
ISDN card: 2BD1

Driver: HiSax driver

Selection of ISDN protocol
● Euro-ISDN (EDSS1)
○ 1TR6
○ Leased line
○ NI1

Country: Germany
Code: +49
Areacode:
Dialprefix:
☒ Start ISDN Log

Start Mode: OnBoot

Figure 21.7: *ISDN Configuration*

1. **ISDN card directly connected to phone outlet (NTBA)**

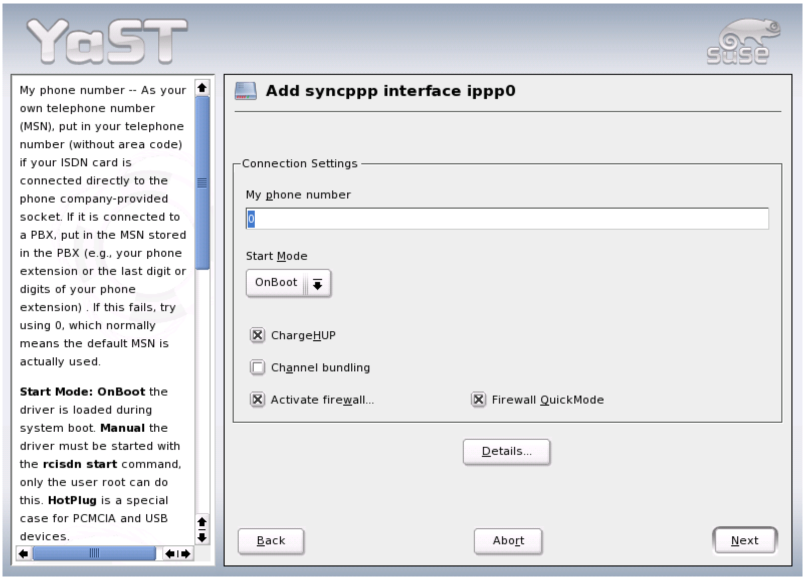
   A standard ISDN line provides for three phone numbers (called multiple subscriber numbers, or MSNs). If the subscriber asked for more, there may be up to ten. One of these MSNs must be entered here, but without your area code. If you should enter the wrong number, your phone operator will automatically fall back to the first MSN assigned to your ISDN line.

2. **ISDN card connected to a phone exchange**

   Again the configuration may vary depending on the equipment installed:

   (a) Smaller phone exchanges 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



My phone number -- As your own telephone number (MSN), put in your telephone number (without area code) if your ISDN card is connected directly to the phone company-provided socket. If it is connected to a PBX, put in the MSN stored in the PBX (e.g., your phone extension or the last digit or digits of your phone extension) . If this fails, try using 0, which normally means the default MSN is actually used.

Start Mode: OnBoot the driver is loaded during system boot. **Manual** the driver must be started with the **rcisdn start** command, only the user root can do this. **HotPlug** is a special case for PCMCIA and USB devices.

Add syncppp interface ippp0

Connection Settings

My phone number: 0

Start Mode: OnBoot

☒ ChargeHUP

☐ Channel bundling

☒ Activate firewall...

☒ Firewall QuickMode

Figure 21.8: *ISDN Interface Configuration*

enabled for direct outward dialing. If this does not work, you might also try a single zero as a last resort. For further information, consult the documentation that came with your phone exchange.

(b) 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, you may also try each of the digits from 1 to 9.

If you want the connection to be terminated just before the next charge unit is due, enable the 'ChargeHUP' checkbox. However, remember that this may not work with every ISP yet. You can also enable channel bundling (multilink PPP) by selecting the corresponding checkbox. Finally, you can enable SuSEfirewall2 for your link by selecting the checkbox 'Activate Firewall'.

The 'Details...' button takes you to a dialog where you can implement more complex connection schemes, which however is not relevant for normal home users. You can leave this dialog by selecting ‘Next’.

The next dialog lets you specify 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, both according to the specifications of your ISP. If the interface that you are configuring shall act as the default route into the Internet, select the ‘Defaultroute’ checkbox — but always remember that 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 to select an ISP. The IPSs 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 where you can enter all the details concerning for your ISP. The ‘ISDN Interface Type’ is set to ‘ISDN SyncPPP’ by default. When entering the phone number, make sure you do not include any blanks or commas among the digits. Finally, enter your login and the password as provided by the ISP. When you are finished with the dialog, select ‘Next’.

To be able to use ‘Dial on Demand’ on a stand-alone workstation, you also have to specify the name server (DNS server). Most ISPs nowadays support dynamic DNS, which is to say that the IP address of a name server is communicated by the ISP each time you connect. For a single workstation, however, you still need to provide a placeholder address in the DNS server field, for instance `192.168.22.99`. If your ISP does not support dynamic DNS, specify the IP address(es) of the name server(s) that the ISP has given to you. The dialog also lets you specify a time-out for the connection, that is to say the period of network inactivity (in seconds) after which the connection should be automatically terminated. Lastly, confirm your settings with ‘Next’. YaST will display a summary of the configured interfaces. To make all these settings active, select ‘Finish’.

21.4.6 Hotplug and PCMCIA

Note

s390, zseries: Hotplug Support

On IBM’s S/390 and zSeries platforms, all network cards are detected and initialized by the hotplug subsystem.

Note

Hotplug network cards, like PCMCIA or USB devices, are managed in a somewhat special way. Normal network cards are fixed components assigned a permanent device name, such as `eth0`. By contrast, PCMCIA cards are assigned a free device name dynamically on an as-needed basis. To avoid conflicts with fixed network cards, hotplug and PCMCIA services are loaded after the network has been started.

PCMCIA-related configuration and start scripts are located in the directory `/etc/sysconfig/pcmcia`. The scripts are executed as soon as cardmgr, the PCMCIA device manager, detects a newly inserted PCMCIA card — which is why PCMCIA services do not need to be started before the network during boot.

21.4.7 Configuring IPv6

Note

s390, zseries: IPv6 Support

IPv6 is not supported by CTC and IUCV cards.

Note

To configure IPv6, you will not normally need to make any changes on the individual workstations. However, IPv6 support must be loaded. To do this, 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 manual page of ifup (`man ifup`) to get information about how to set up various types of tunnels using the `/etc/sysconfig/network` files.

21.5 Routing in SUSE LINUX

The routing table is set up in SUSE LINUX via the configuration files `/etc/sysconfig/network/routes` and `/etc/sysconfig/network/`

`ifroute-*`. 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           GATEWAY NETMASK   INTERFACE [ TYPE ] [ OPTIONS ]
DESTINATION           GATEWAY PREFIXLEN INTERFACE [ TYPE ] [ OPTIONS ]
DESTINATION/PREFIXLEN GATEWAY -         INTERFACE [ TYPE ] [ OPTIONS ]
```

To omit GATEWAY, NETMASK, PREFIXLEN, or INTERFACE, write `-` instead. The entries TYPE and OPTIONS may just be omitted.

- 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 host name.
- The second column contains the default gateway or a gateway through which a host or a network can be accessed.
- The third column contains the netmask for networks or hosts behind a gateway. The mask is `255.255.255.255`, for example, for a host behind a gateway.
- The last 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.

The following scripts in the directory `/etc/sysconfig/network/scripts/` assist with the handling of routes:

ifup-route for setting up a route

ifdown-route for disabling a route

ifstatus-route for checking the status of the routes

21.6 SLP services in the network

The *Service Location Protocol* (SLP for short) was developed with the aim of simplifying the configuration of networked clients within a local network. In order to configure a network client, including all required services, the administrator traditionally needs detailed knowledge of the servers available in the network. SLP is used to make the availability of a certain service type known to all clients in the local network. Applications that support SLP can use the information distributed via SLP and can therefore be configured automatically.

Note

SLP support in SUSE LINUX Enterprise Server

Services that offer SLP support include: cupsd, rsyncd, ypserv, openldap2, openwbem (CIM), ksysguardd, saned, kdm vnc login, smpppd, rpasswd, postfix, sshd (via fish.)

Note

21.6.1 SLP support in SUSE LINUX

SUSE LINUX supports the installation of installation sources brokered via 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 installation server, YOU server, file server or print server on your SUSE LINUX Enterprise Server.

Installation via SLP

If you offer an installation server with SUSE LINUX installation media within your network, then this can be registered with SLP (for details see 4.1 on page 146). 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 for selection purposes.

Registering your own services

Manu applications under SUSE LINUX already have integrated SLP support through the use of the `libslp` library. If you want to make other services that have not compiled SLP support available via SLP, then a number of choices are open to you:

Static registration via `/etc/slp.reg.d/`

Create a separate registration file for each new service. There follows 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 so-called *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 by the full host name. The name of the TCP port on which the relevant service can be found now follows, separated by a colon. You should now enter the language in which the service is to appear and the duration of registration in seconds. These should be separated from the service URL by commas. The value for the duration of registration can be set between `0` and `65535`. `0` would render registration invalid, while `65535` removes all restrictions.

The registration file also contains the two variables `watch-tcp-port` and `description`. The former links the SLP service announcement to whether the relevant service is active because slpd checks the status of the service. The second variable contains a more precise description of the service which is displayed in suitable browsers.

Note

YaST and SLP

Some services brokered by YaST, such as ‘installation server’ or ‘YOU server’, perform this registration for you automatically when you activate the SLP checkboxes in the module dialogs. YaST then creates registration files for these services.

Note

Static registration `/etc/slp.reg`

The only difference from the procedure described above is the grouping together of all services within a central file.

Dynamic registration with slptool If a service is to be registered for SLP from proprietary scripts, you should use the slptool command line front-end.

SLP front-ends in SUSE LINUX

SUSE LINUX contains several front-ends that enable SLP information to be checked and used by means of a network:

slptool slptool is a simple command line program that can be used to announce SLP inquiries in the network or to announce proprietary services. `slptool --help` lists all available options and functions. slptool can also be called from scripts that are to process SLP information.

YaST SLP browser YaST contains a separate SLP browser that lists all services in the local network announced via SLP in a tree diagram under 'Network Services' → 'SLP browser'

Konqueror When used as a network browser, Konqueror can display all SLP services available in the local network by calling `slp:/`. Click on the icons in the main window to obtain more detailed information about the relevant service.

If you call Konqueror with `service:/`, then a single click on the relevant icon in the browser window will set up a connection with the selected service.

Activating SLP

Note

Activating slpd

slpd must run on your system if you want to offer your own server services. It is not necessary to start this daemon simply to make service inquiries.

Note

Like most system services under SUSE LINUX, the slpd daemon is controlled by means of a separate init script. The daemon is inactive by default. If you want to activate it for the duration of a session, you should use the `rcslpd start` command as `root` to start it and `rcslpd stop` to stop it. You can activate a restart or status check with `restart` or `status`. If slpd is to be active by default, call the `insserv slpd` command once as `root`. This automatically includes slpd in the set of services to be started when a system is booted.

21.6.2 Further information

The following sources are available for further information on the subject of 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.com The homepage of the OpenSLP project.

file:/usr/share/doc/packages/openslp/*
This directory contains all available documentation on SLP, including a `README.SuSE` containing the SUSE LINUX details, the RFCs mentioned above and two introductory HTML documents. Programmers who want to use the SLP functions should install the `openslp-devel` package to be able to use the supplied *Programmers Guide*.

21.7 DNS — Domain Name System

DNS (domain name system) is needed to resolve the domain and host names into IP addresses. In this way, the IP address 192.168.0.0 is assigned to the host name `earth`, for example. Before setting up your own name server, read the general information about DNS in Section 21.1.3 on page 417. The configuration examples below are only valid for BIND 9.

21.7.1 Starting the Name Server BIND

On a SUSE LINUX system, the name server BIND (short for *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 the 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 as `/usr/share/doc/packages/bind/sample-config`.

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 to use it, as BIND would otherwise not forward any more requests for this domain. The provider's web server, 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 in an unexpected way, 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 File 21.10 are just examples. Change these entries according to your own setup.

Example 21.10: *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, for `localhost`, `0.0.127.in-addr.arpa`, and the `type hint` entry under ".", which should always be present. The corresponding files do not need to be modified and should work as is. 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 the command `rcnamed reload`. You can achieve the same by stopping and restarting the name server with the command `rcnamed restart`. The server can also be stopped at any time by entering the command `rcnamed stop`.

21.7.2 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 host names, IP addresses, and so on, are stored in separate files in the `/var/lib/named` directory. The details of this are described further below.

The `/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 as well as `acl` (access control list) entries are optional. Comment lines begin with a "#" sign or "//". A minimalistic `/etc/named.conf` looks like File 21.11.

***Example 21.11:** 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";
};
```

Note

Further Information on BIND

Refer to `/usr/share/doc/packages/bind/README.SuSE` for further up-to-date information on BIND in SUSE LINUX.

Note

21.7.3 Important Configuration Options

directory "/var/lib/named"; specifies the directory where BIND can find the files containing the zone data.

forwarders 10.0.0.1;; specifies the name servers (mostly of the provider) to which DNS requests should forwarded if they cannot be resolved directly.

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; 192.168.0.1;; tells BIND to which network interface and port to listen. The `port 53` specification can be left out, as 53 is the default port. If this entry is completely omitted, BIND accepts requests on all interfaces.

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; 192.168.1/24;; defines the networks from which clients can post DNS requests. 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`. Specify 0 to completely suppress such statistics or specify 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 nonexisting interfaces. If this value is set to `0`, this is not done and BIND only listens at the interfaces detected at start-up. Otherwise, the interval can be defined in minutes. The default is 60 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.

21.7.4 The Configuration Section Logging

What, how, and where logging takes place can be extensively configured in BIND. Normally, the default settings should be sufficient. File 21.12 represents the simplest form of such an entry and completely suppresses any logging.

Example 21.12: *Entry to Disable Logging*

```
logging {
        category default { null; };
};
```

21.7.5 Zone Entry Structure

Example 21.13: *Zone Entry for my-domain.de*

```
zone "my-domain.de" in {
      type master;
      file "my-domain.zone";
      notify no;
};
```

After `zone`, the name of the domain to administer is specified, `my-domain.de`, followed by `in` and a block of relevant options enclosed in curly braces, as shown in File 21.13. To define a *slave zone*, the `type` is simply switched to `slave` and a name server is specified that administers this zone as `master` (which, in turn, may be a slave of another master), as shown in File 21.14.

Example 21.14: *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. 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, as this data is fetched from another name server. To differentiate master and slave files, the directory `slave` is specified for the slave files.

masters 10.0.0.1; ; 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.

21.7.6 Structure of Zone Files

Two types of zone files are needed. One serves to assign IP addresses to host names and the other does the reverse — supplies a host name for an IP address.

Note

Using the Dot in Zone Files

The `.` has an important meaning in the zone files. If host names are given without a final `.`, the zone is appended. Therefore, complete host names specified with a full domain name must end with a `.`, to avoid that the domain is added to it again. A missing or wrongly placed dot is probably the most frequent cause of name server configuration errors.

Note

The first case to consider is the zone file `world.zone`, responsible for the domain `world.cosmos`, shown in File 21.15.

Example 21.15: *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
```

```
13            IN A        192.168.1.1
14 sun        IN A        192.168.0.2
15 moon       IN A        192.168.0.3
16 earth      IN A        192.168.1.2
17 mars       IN A        192.168.1.3
18 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, a `@` 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 will follow. Because the `@` sign already has a special meaning, `.` is entered here instead, so for `root@world.cosmos` the entry must read `root.world.cosmos.`. Again the `.` sign must be included at the end to prevent the zone from being added.
- The `(` is used to include 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 ten-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 will be 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 host name 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 will be made with the next higher value.

Lines 12–17: These are the actual address records where one or more IP addresses are assigned to host names. 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 an `A`. If the address is an IPv6 address, the entry is marked with `A6`. (The previous token for IPv6 addresses was `AAAA`, which is now obsolete.)

Line 18: The alias `www` can be used to address `mond` (`CNAME` = *canonical name*).

The pseudodomain `in-addr.arpa` is used for the reverse lookup of IP addresses into host names. 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 File 21.16 on the following page.

***Example 21.16:** Reverse Lookup*

```
1  $TTL 2D
2  1.168.192.in-addr.arpa. IN SOA gateway.world.cosmos. root.world.cosmos. (
3                          2003072441      ; serial
4                          1D              ; refresh
5                          2H              ; retry
6                          1W              ; expiry
7                          2D )            ; minimum
8
9                          IN NS           gateway.world.cosmos.
10
11 1                       IN PTR          gateway.world.cosmos.
12 2                       IN PTR          earth.world.cosmos.
13 3                       IN PTR          mars.world.cosmos.
```

Line 1: $TTL defines the standard TTL that applies to all entries here.

Line 2: The configuration file is supposed to activate reverse lookup for the network `192.168.1.0`. Given that the zone is called `1.168.192.in-addr.arpa`, we would not want to add this to the host names. Therefore, all host names 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.

21.7.7 Secure Transactions

Secure transactions can be carried out 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 the 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.

A TSIG key can be generated 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 instance). 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==;
};
```

Caution

File Permissions of /etc/named.conf

Make sure 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`.

Caution

To enable the server `host1` to use the key for `host2` (which has the address `192.168.2.3` in our 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`.

In addition to any ACLs that are defined for IP addresses and address ranges, add TSIG keys for these 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`.

21.7.8 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 21.7.7 on page 464.

21.7.9 DNSSEC

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. Currently the DSA encryption algorithm is 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`.

21.7.10 Configuration with YaST

You can use the DNS module of YaST to configure a DNS server for your local network. The module can work in two different modes:

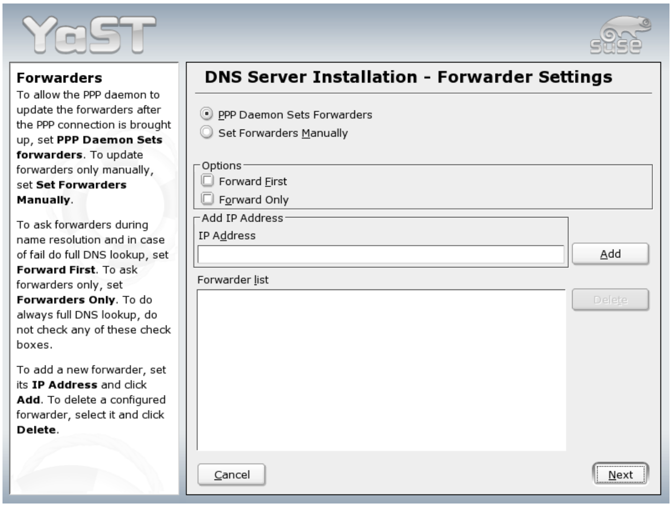
Wizard Configuration When starting the module for the first time, you will be prompted to make just a few basic decisions concerning the server administration. Completing this initial setup produces a very basic server configuration which should be functioning in its essential aspects.

Expert Configuration The expert mode can be used to deal with the more advanced configuration tasks, such as setting up ACLs, logging, TSIG keys, and other options.

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.

DNS Server Installation: Forwarder Settings
When starting the module for the first time, you will see the dialog as shown in Figure 21.9 on the following page. It allows you decide whether the PPP daemon should provide a list of forwarders upon dialout via DSL or ISDN ('PPP Daemon Sets Forwarders'), or whether you want to supply your own list ('Set Forwarders Manually').



Forwarders

To allow the PPP daemon to update the forwarders after the PPP connection is brought up, set **PPP Daemon Sets forwarders**. To update forwarders only manually, set **Set Forwarders Manually**.

To ask forwarders during name resolution and in case of fail do full DNS lookup, set **Forward First**. To ask forwarders only, set **Forwarders Only**. To do always full DNS lookup, do not check any of these check boxes.

To add a new forwarder, set its **IP Address** and click **Add**. To delete a configured forwarder, select it and click **Delete**.

DNS Server Installation - Forwarder Settings

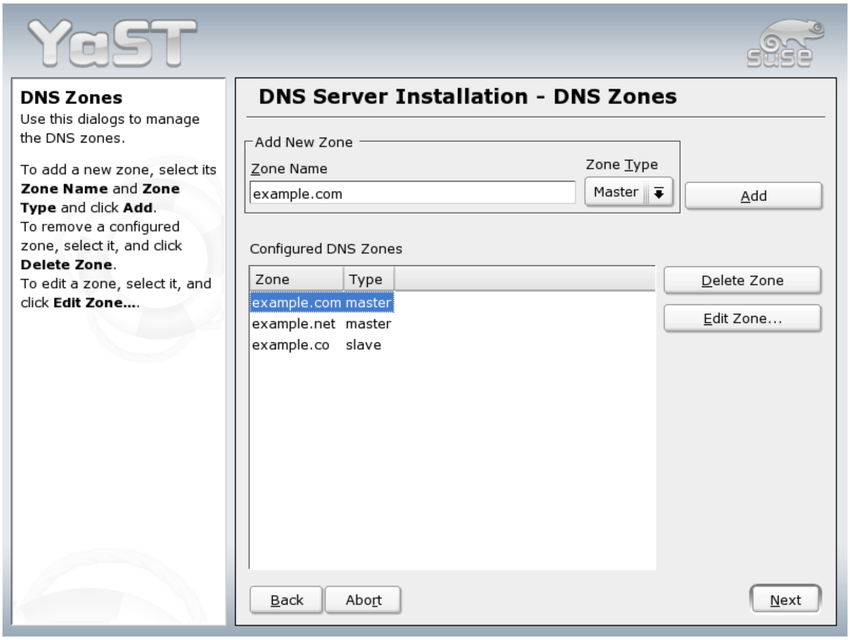
● PPP Daemon Sets Forwarders
○ Set Forwarders Manually

Options
☐ Forward First
☐ Forward Only

Figure 21.9: DNS Server Installation: Forwarder Settings

DNS Server Installation: DNS Zones

The individual entries shown in Figure 21.9 are explained in the discussion of the expert configuration (see 21.7.10 on page 471).



DNS Zones

Use this dialogs to manage the DNS zones.

To add a new zone, select its **Zone Name** and **Zone Type** and click **Add**. To remove a configured zone, select it, and click **Delete Zone**. To edit a zone, select it, and click **Edit Zone...**.

DNS Server Installation - DNS Zones

Add New Zone

Zone Name: example.com

Zone Type: Master

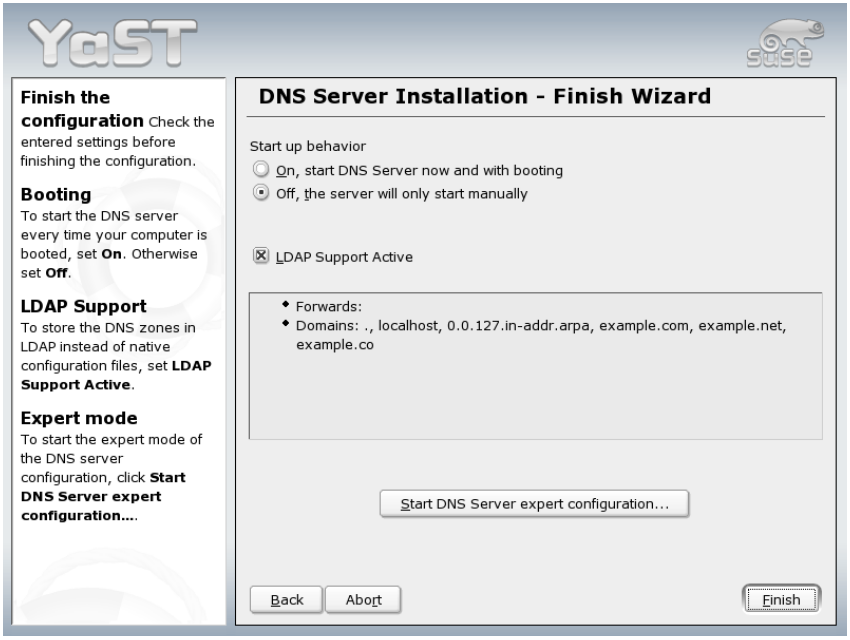
Configured DNS Zones

Zone	Type
example.com	master
example.net	master
example.co	slave

Figure 21.10: DNS Server Installation: DNS Zones

DNS Server Installation: Finish Wizard

In the final step, specify whether the DNS server should always be started or not (on or off) as part of the bot procedure, and whether it should use LDAP support (see Figure 21.11 on the next page).



Finish the configuration Check the entered settings before finishing the configuration.

Booting
To start the DNS server every time your computer is booted, set **On**. Otherwise set **Off**.

LDAP Support
To store the DNS zones in LDAP instead of native configuration files, set **LDAP Support Active**.

Expert mode
To start the expert mode of the DNS server configuration, click **Start DNS Server expert configuration...**.

DNS Server Installation - Finish Wizard

Start up behavior

○ On, start DNS Server now and with booting

● Off, the server will only start manually

☒ LDAP Support Active

- Forwards:
- Domains: ., localhost, 0.0.127.in-addr.arpa, example.com, example.net, example.co

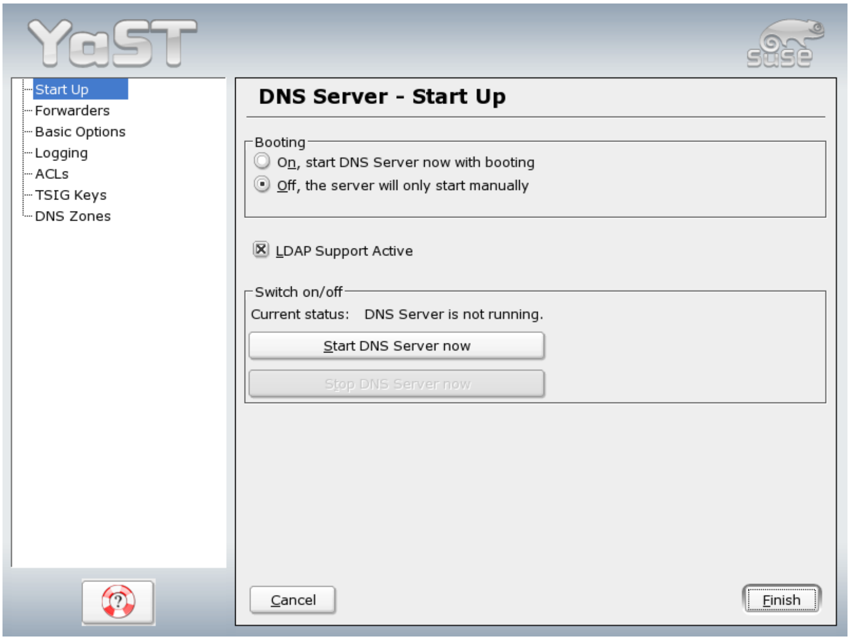
Figure 21.11: *DNS Server Installation: Finish Wizard*

Expert Configuration

After starting the module, YaST opens a window displaying several configuration options. Completing the window gives you a DNS server configuration with the basic functions in place:

DNS Server: Start up Under the 'Booting' heading, you can define whether the DNS server should be 'On' or 'Off' by default. To start the DNS server right away, select the button 'Start DNS Server now'.

By selecting 'LDAP Support Active', you can have the zone files managed by an LDAP database. Any changes of zone data as written to the LDAP database will be picked up by the DNS server as soon as it is restarted or prompted to reload its configuration.


YaST
Start Up
Forwarders
Basic Options
Logging
ACLs
TSIG Keys
DNS Zones
Cancel
Finish

DNS Server - Start Up

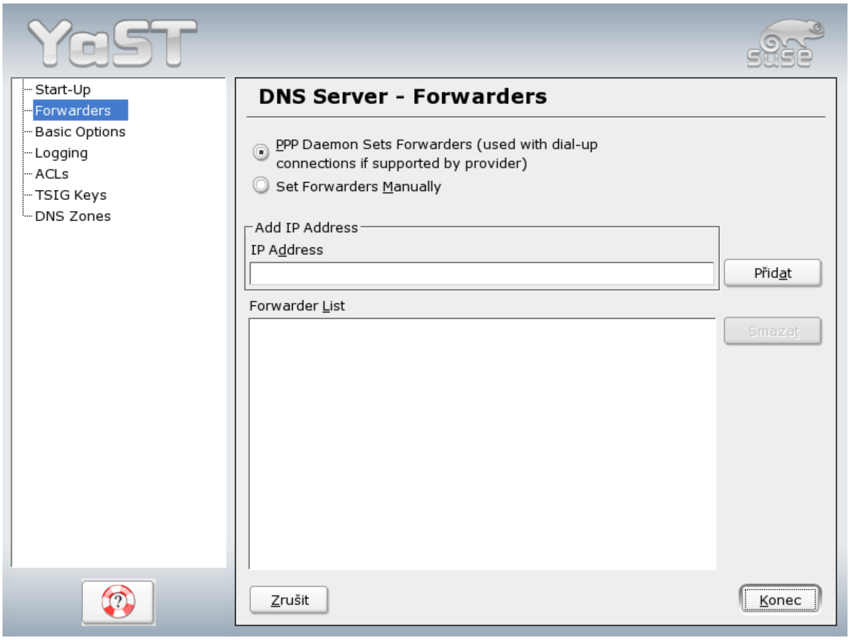
Booting
○ On, start DNS Server now with booting
● Off, the server will only start manually

☒ LDAP Support Active

Switch on/off
Current status: DNS Server is not running.
Start DNS Server now
Stop DNS Server now

Figure 21.12: DNS Server: Start up

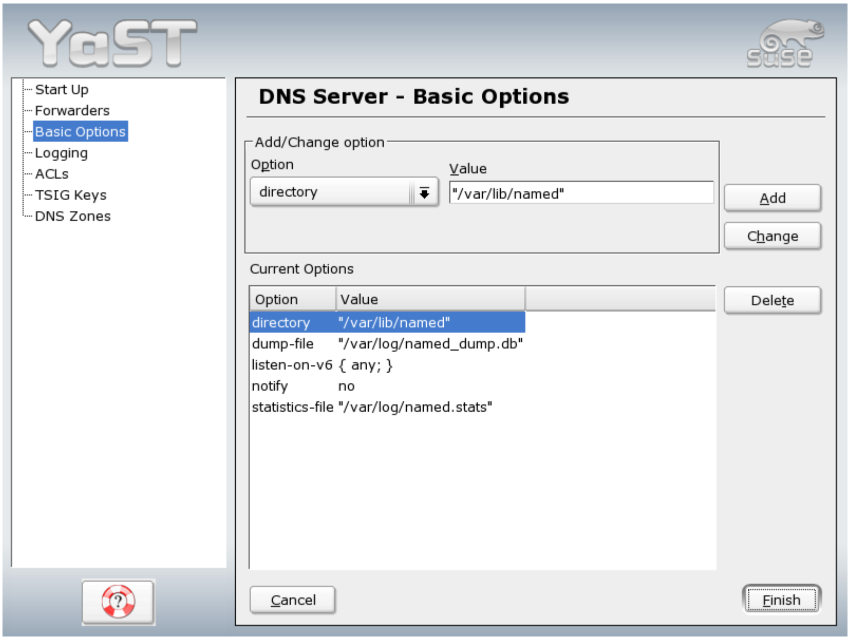
DNS Server: Forwarders This is the same dialog as the one opened after starting the wizard configuration (see 21.7.10 on page 467).


YaST
SuSE
Start-Up
Forwarders
Basic Options
Logging
ACLs
TSIG Keys
DNS Zones
DNS Server - Forwarders
Add IP Address
IP Address
Přidat
Forwarder List
Smazat
Zrušit
Konec

● PPP Daemon Sets Forwarders (used with dial-up connections if supported by provider)
○ Set Forwarders Manually

Figure 21.13: *DNS Server: Forwarders*

DNS Server: Basic Options This section allows you to set basic server options. From the 'Option' menu, select the item that you are interested in, and then specify the value in the corresponding entry field. The new entry is included in the list by selecting 'Add'.


YaST
Start Up
Forwarders
Basic Options
Logging
ACLs
TSIG Keys
DNS Zones
DNS Server - Basic Options
Add/Change option
Option
Value
Add
Change
Current Options
Delete
Cancel
Finish

Option: directory
Value: "/var/lib/named"

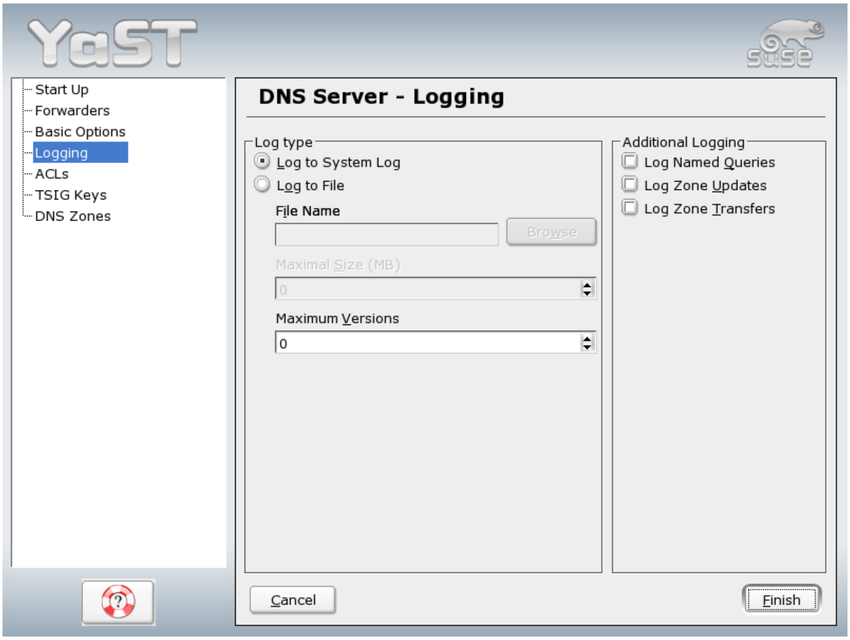
Option	Value
directory	"/var/lib/named"
dump-file	"/var/log/named_dump.db"
listen-on-v6	{ any; }
notify	no
statistics-file	"/var/log/named.stats"

Figure 21.14: DNS Server: Basic Options

DNS Server: Logging This section allows you to set options concerning the contents and the location of the DNS server's log data.

Under the 'Log type' heading, you can specify where the DNS server should write its log data. You can either use the system-wide log file `/var/log/messages`, by selecting 'Log to System Log'. Or you can specify a different file, by selecting 'Log to File'. In the latter case, you can additionally specify the maximum file size in megabytes and the number of log files to be stored.

Further options are available under 'Additional Logging': Enabling 'Log Named Queries' causes *each and every* query to be logged, in which case the log file could grow to enormous proportions. Therefore, 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 the option 'Log Zone Updates'. Similarly, to log the data traffic during a zone transfer from master to slave, enable the option 'Log Zone Transfer' (see Figure 21.15 on the next page).


YaST
Start Up
Forwarders
Basic Options
Logging
ACLs
TSIG Keys
DNS Zones
DNS Server - Logging
Log type
Additional Logging
Browse
Cancel
Finish

● Log to System Log
○ Log to File

File Name:

Maximal Size (MB): 0

Maximum Versions: 0

☐ Log Named Queries
☐ Log Zone Updates
☐ Log Zone Transfers

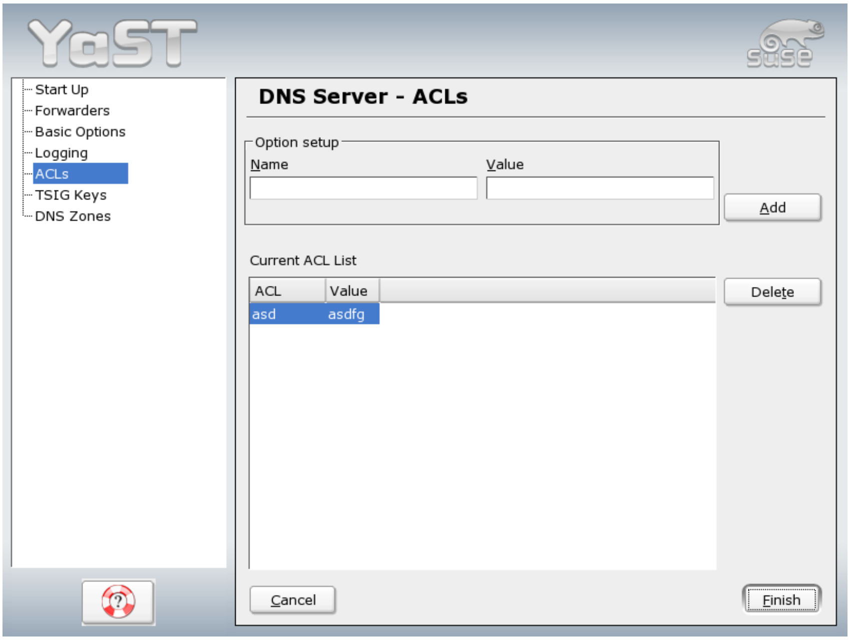
Figure 21.15: DNS Server: Logging

DNS Server: ACLs This window allows to to define ACLs (Access Control Lists) in order to enforce access restrictions.

After providing a distinct name under 'Name', you need to specify an IP address (with or without netmask) under 'Value', in the following fashion:

```
{ 10.10/16; }
```

As you can see, the syntax of the configuration file requires that the address ends with a semicolon and is put into braces.



YaST
SuSE
Start Up
Forwarders
Basic Options
Logging
ACLs
TSIG Keys
DNS Zones
Add
Delete
Cancel
Finish

DNS Server - ACLs

Option setup

Name

Value

Current ACL List

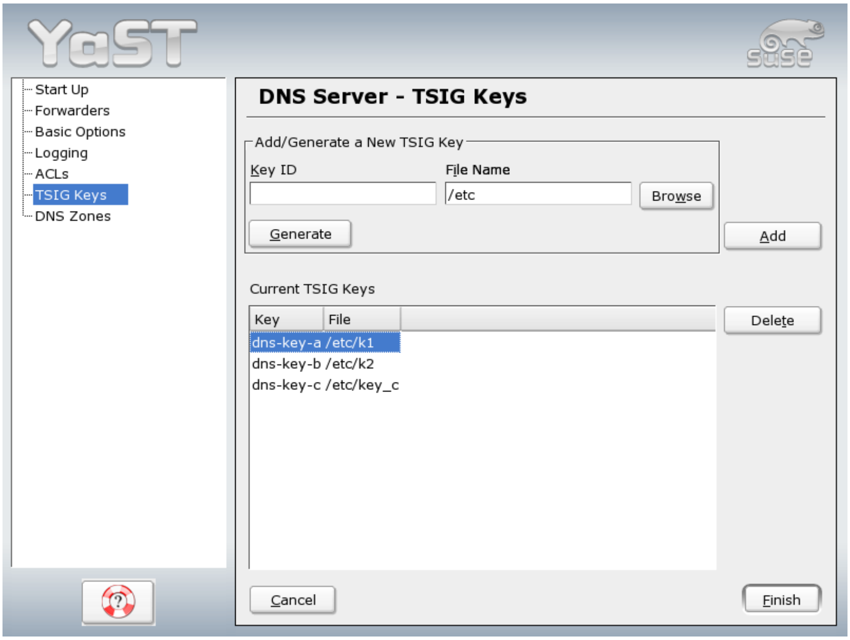
ACL	Value
asd	asdfg

Figure 21.16: *DNS Server: ACLs*

DNS Server: TSIG Keys The main purpose of TSIGs (Transaction SIGnatures) is to secure communications between DHCP and DNS servers. They are described in Section 21.7.7 on page 464.

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 ('File Name'). 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'.


YaST
Start Up
Forwarders
Basic Options
Logging
ACLs
TSIG Keys
DNS Zones
DNS Server - TSIG Keys
Add/Generate a New TSIG Key
Key ID
File Name
Browse
Generate
Add
Current TSIG Keys
Delete
Cancel
Finish

File Name: /etc

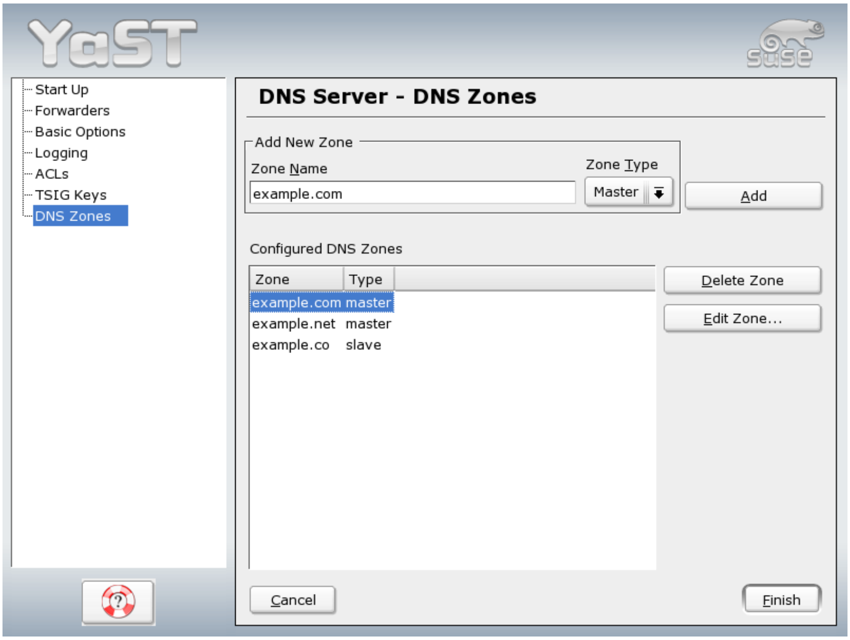
Current TSIG Keys:

Key	File
dns-key-a	/etc/k1
dns-key-b	/etc/k2
dns-key-c	/etc/key_c

Figure 21.17: DNS Server: TSIG Keys

DNS Server: DNS Zones This part of the configuration comprises several dialogs to configure the management of zone files (see Section 21.7.6 on page 461).

For a new zone, provide a name for it in the 'Zone Name' entry field. To add a reverse zone, the name must end in `.in-addr.arpa`. Finally select the 'Zone Type' (master or slave) (see Figure 21.18 on the facing page). For each zone you can set additional options after selecting 'Edit Zone ...'.


YaST
Start Up
Forwarders
Basic Options
Logging
ACLs
TSIG Keys
DNS Zones
Add
Delete Zone
Edit Zone...
Cancel
Finish

DNS Server - DNS Zones

Add New Zone

Zone Name: example.com

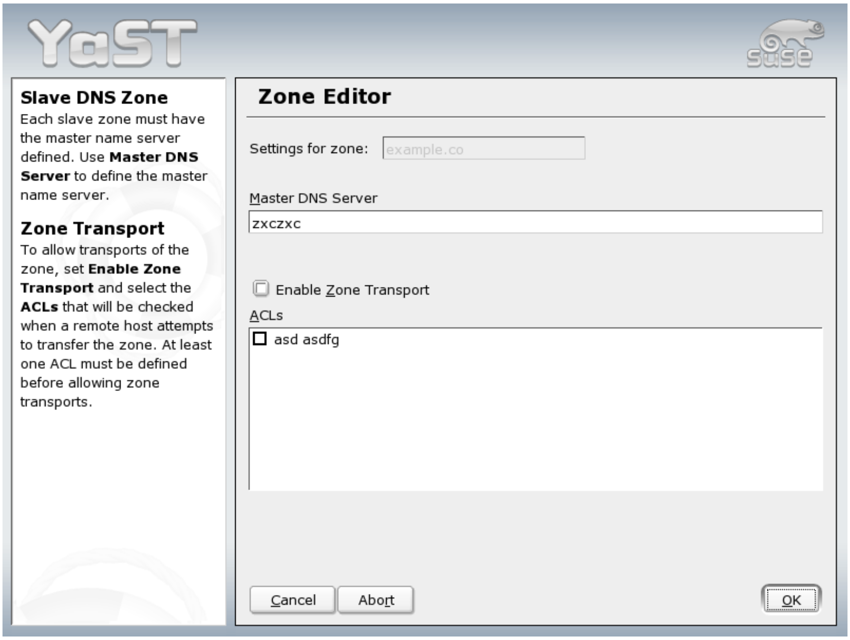
Zone Type: Master

Configured DNS Zones

Zone	Type
example.com	master
example.net	master
example.co	slave

Figure 21.18: *DNS Server: DNS Zones*

DNS Server: Slave Zone Editor This dialog is opened if you selected the zone type 'Slave' in the step described in 21.7.10 on the preceding page. Under 'Master DNS Server', specify the master from which the slave shall fetch its data. To limit access to the server, you can select one of the previously defined ACLs from the list (see Figure 21.19 on the following page).



Slave DNS Zone
Each slave zone must have the master name server defined. Use **Master DNS Server** to define the master name server.

Zone Transport
To allow transports of the zone, set **Enable Zone Transport** and select the **ACLs** that will be checked when a remote host attempts to transfer the zone. At least one ACL must be defined before allowing zone transports.

Zone Editor

Settings for zone: example.co

Master DNS Server: zxczxc

☐ Enable Zone Transport

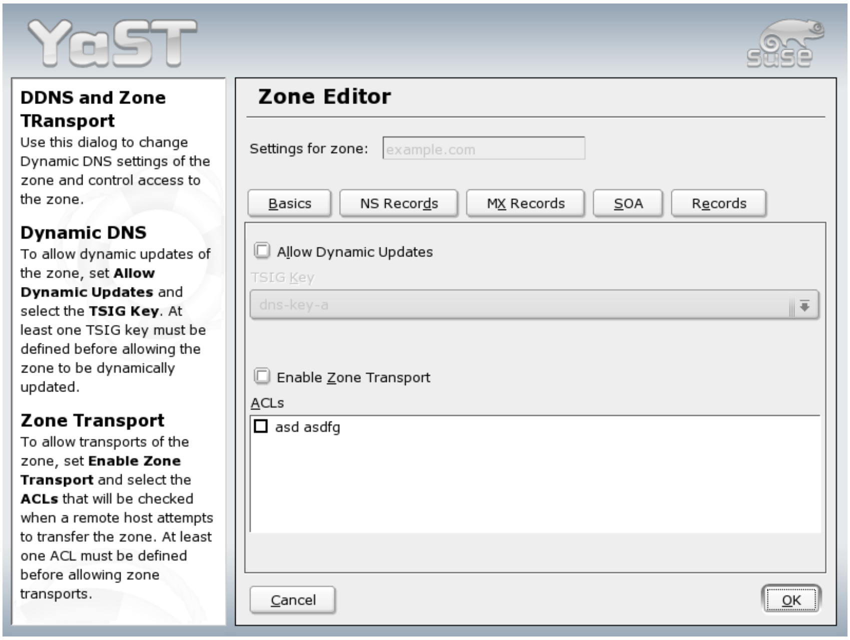
ACLs:
☐ asd asdfg

Figure 21.19: DNS Server: Slave Zone Editor

DNS Server: Master Zone Editor This dialog is opened if you selected the zone zone type 'Master' in the step described in 21.7.10 on page 476. The dialog comprises several pages: Basic (the one opened first), NS Records, MX Records, SOA, and Records. For each of these pages you find a description in the following paragraphs.

The dialog shown in Figure 21.20 on the facing page 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. Note however that the key must have been defined before the update action starts.

To enable zone transfers, you need select the corresponding ACLs. ACLs must have been defined before in order to be able to do so.



DDNS and Zone TRansport
Use this dialog to change Dynamic DNS settings of the zone and control access to the zone.

Dynamic DNS
To allow dynamic updates of the zone, set **Allow Dynamic Updates** and select the **TSIG Key**. At least one TSIG key must be defined before allowing the zone to be dynamically updated.

Zone Transport
To allow transports of the zone, set **Enable Zone Transport** and select the **ACLs** that will be checked when a remote host attempts to transfer the zone. At least one ACL must be defined before allowing zone transports.

Zone Editor

Settings for zone: example.com

☐ Allow Dynamic Updates

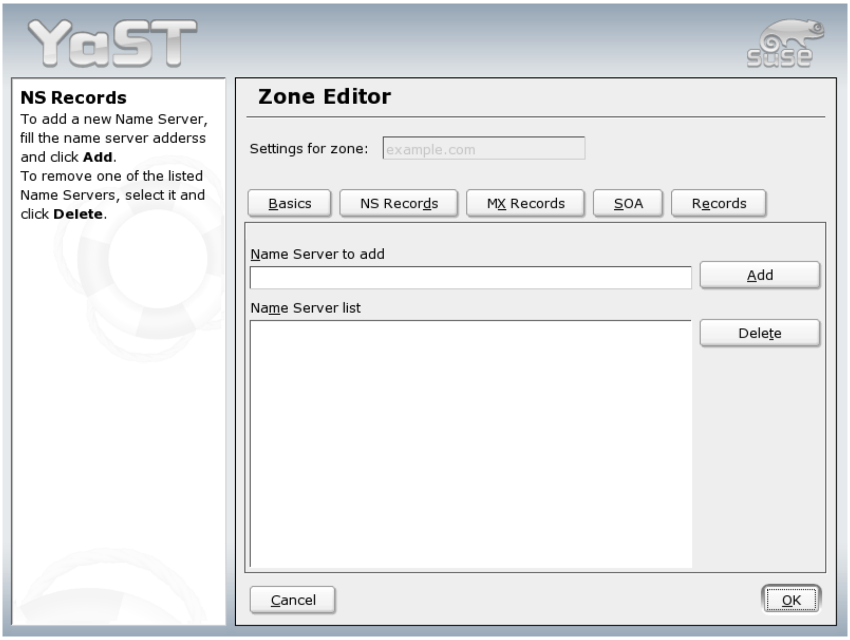
TSIG Key: dns-key-a

☐ Enable Zone Transport

ACLs: ☐ asd asdfg

Figure 21.20: *DNS Server: Zone Editor (Basic)*

DNS Server: 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' and then confirm with 'Add' (see Figure 21.21 on the next page).



NS Records

To add a new Name Server, fill the name server adderss and click **Add**.

To remove one of the listed Name Servers, select it and click **Delete**.

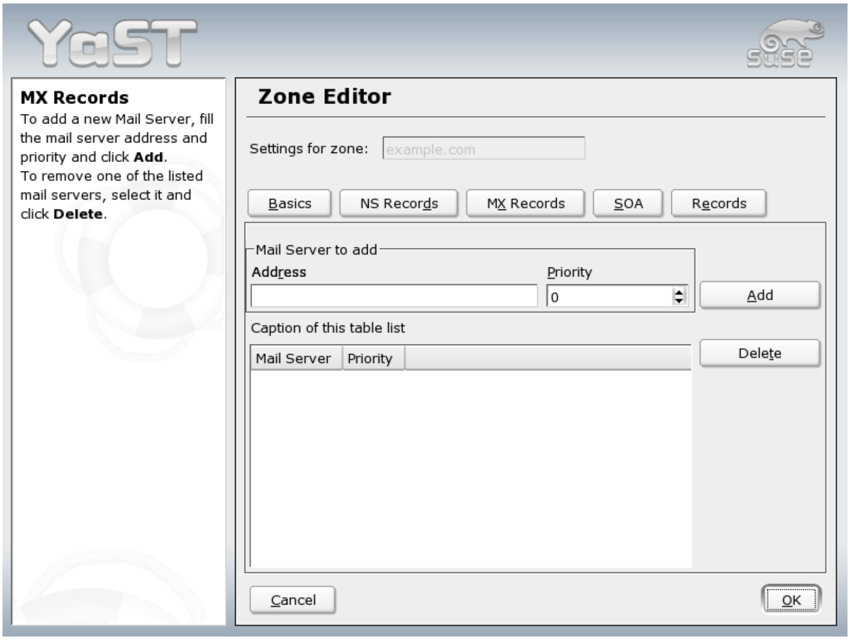
Settings for zone: example.com

YaST
SuSE
Zone Editor
Basics
NS Records
MX Records
SOA
Records
Name Server to add
Add
Name Server list
Delete
Cancel
OK

Figure 21.21: DNS Server: Zone Editor (NS Records)

DNS Server: Zone Editor (MX Records)

To add a mail server for the current zone to the existing list, enter the corresponding address and the priority value. After doing so, confirm by selecting 'Add' (see Figure 21.22 on the facing page).



MX Records
To add a new Mail Server, fill the mail server address and priority and click **Add**.
To remove one of the listed mail servers, select it and click **Delete**.

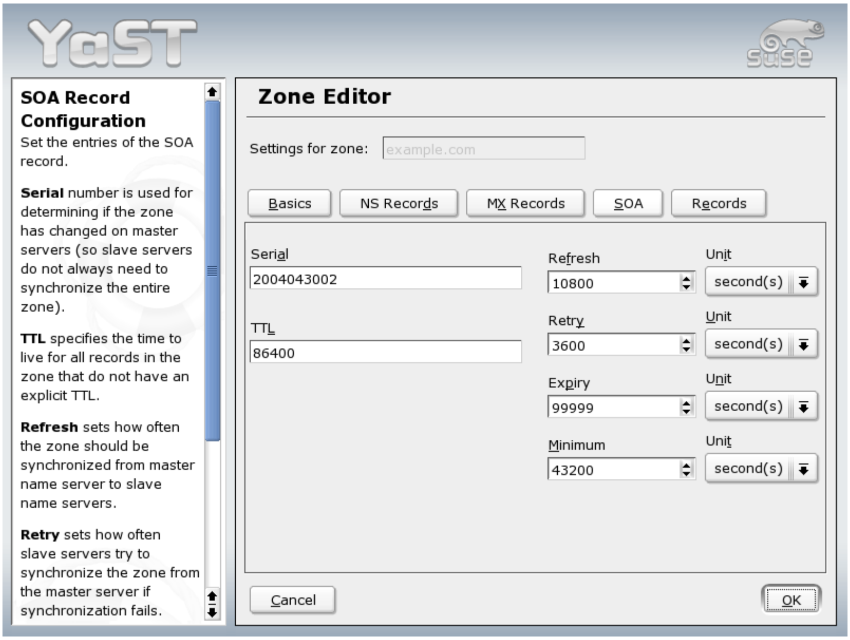
Settings for zone: example.com

Priority: 0

YaST
SUSE
Zone Editor
Basics
NS Records
MX Records
SOA
Records
Mail Server to add
Address
Priority
Add
Caption of this table list
Mail Server
Priority
Delete
Cancel
OK

Figure 21.22: DNS Server: Zone Editor (MX Records)

DNS Server: Zone Editor (SOA) This page allows you to create SOA (Start of Authority) records. For an explanation of the individual options, refer to Example 21.15 on page 461.



SOA Record Configuration
Set the entries of the SOA record.

Serial number is used for determining if the zone has changed on master servers (so slave servers do not always need to synchronize the entire zone).

TTL specifies the time to live for all records in the zone that do not have an explicit TTL.

Refresh sets how often the zone should be synchronized from master name server to slave name servers.

Retry sets how often slave servers try to synchronize the zone from the master server if synchronization fails.

Settings for zone: example.com

Serial: 2004043002

TTL: 86400

Refresh: 10800 Unit: second(s)

Retry: 3600 Unit: second(s)

Expiry: 99999 Unit: second(s)

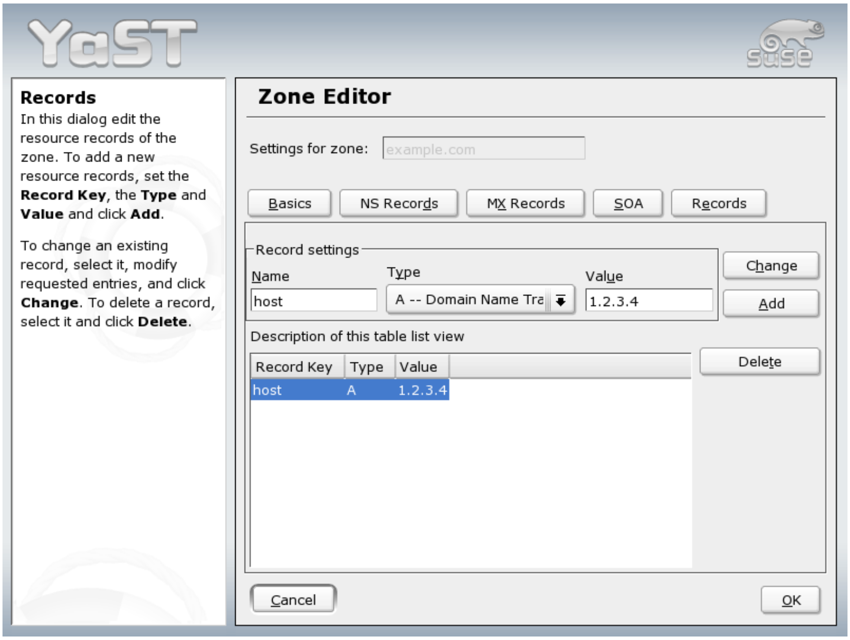
Minimum: 43200 Unit: second(s)

Figure 21.23: *DNS Server: Zone Editor (SOA)*

DNS Server: Zone Editor (Records)
This dialog lets you manage a list of IP addresses and the corresponding names as assigned to them. In the ‘Name’ field, enter the host name and then select its type (using the pop-up menu with the same label). ‘A-Record’ represents the main entry, while ‘CNAME’ is an alias, and under ‘MX-Relay’ the entry (name) is replaced with its value.

21.7.11 For More Information

For additional information, refer to the *BIND Administrator Reference Manual*, 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.



Records
In this dialog edit the resource records of the zone. To add a new resource records, set the **Record Key**, the **Type** and **Value** and click **Add**.

To change an existing record, select it, modify requested entries, and click **Change**. To delete a record, select it and click **Delete**.

Zone Editor

Settings for zone: example.com

Record settings
Name: host
Type: A -- Domain Name Tra
Value: 1.2.3.4

Description of this table list view

Record Key	Type	Value
host	A	1.2.3.4

Figure 21.24: DNS Server: Zone Editor (Records)

21.8 LDAP — A Directory Service

It is crucial within a networked environment to keep important information structured and quickly available. Data chaos does not only loom when using the Internet. The search for important data in the company network can just as quickly grow disproportionately. “What is the extension number of colleague XY? What is his e-mail address?” This problem is solved by 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 a wide range of applications can access them. 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 concerned information. The use of an open and standardized protocol like LDAP (Lightweight Directory Access Protocol) 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, the writing 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 writing accesses can only be executed in a restricted fashion, a directory service is employed for administering 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 instance, 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 the balance over the whole 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.

Many directory services have previously existed and still exist both in Unix and outside it. Novell NDS, Microsoft ADS, Banyan's Street Talk, and the OSI standard X.500 are just a few examples. LDAP was originally planned as a lean flavor of the DAP, the Directory Access Protocol, which was developed for accessing X.500. The X.500 standard regulates the hierarchical organization of directory entries.

LDAP is relieved of a few functions of the DAP. Without having to miss the X.500 entry hierarchy you profit from LDAP's cross-platform capabilities and save resources. and can be employed, The use of TCP/IP makes it substantially easier to establish interfaces between a docking application and the LDAP service.

LDAP, meanwhile, has evolved and is increasingly employed as a stand-alone solution without X.500 support. LDAP supports *referrals* with LDAPv3 (the protocol version in package openldap2), making it possible to realize distributed databases. The usage of SASL (Simple Authentication and Security Layer) is also new.

LDAP is not limited to querying data from X.500 servers, as it was originally planned. There is an open source server slapd, which can store object information in a local database. There is also an extension called slurpd, which is responsible for replicating multiple LDAP servers.

The `openldap2` package consists of:

slapd A stand-alone LDAPv3 server that administers object information in a BerkeleyDB-based database.

slurpd This program enables the replication of modifications to data on the local LDAP server to other LDAP servers installed on the network.

additional tools for system maintenance
slapcat, slapadd, slapindex

21.8.1 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, which makes its employment as a central data administrator in a heterogeneous network impossible.

Unlike NIS, the LDAP service is not restricted to pure Unix networks. Windows servers (from 2000) support LDAP as a directory service. Novell also offers an LDAP 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.

This list can be extended because LDAP is extensible as opposed to NIS. The clearly-defined hierarchical structure of the data greatly helps the administration of very large amounts of data, because it can be searched better.

21.8.2 Structure of an LDAP Directory Tree

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* or, for short, DIT. The complete path to the desired entry, which unambiguously identifies it, is called *distinguished name* or DN. The single nodes along the path to this entry are 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 21.25 on the facing page.

The complete diagram comprises 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 SUSE employee `Geeko Linux`, in this case, is `cn=Geeko Linux,ou=doc,dc=suse,dc=de`. It is composed by adding the RDN `cn=Geeko Linux` to the DN of the preceding entry `ou=doc,dc=suse,dc=de`.

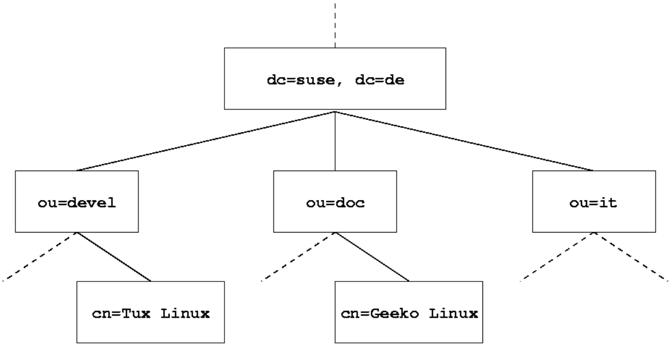


Figure 21.25: *Structure of an LDAP Directory*

The global determination of which types of objects should be stored in the DIT is done 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 21.9 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 21.9: *Commonly Used Object Classes and Attributes*

Object Class	Meaning	Example Entry	Compulsory Attributes
dcObject	*domainComponent* (name components of the domain)	suse	dc
organizationalUnit	*organizationalUnit* (organizational unit)	doc	ou

inetOrgPerson	*inetOrgPerson* (person-related data for the intranet or Internet)	Geeko Linux	sn and cn

Example 21.17 shows an excerpt from a scheme directive with explanations offering some understanding of the new schemes.

Example 21.17: *Excerpt from schema.core (line numbering for explanatory reasons)*

```
#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 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 allowed to be used in conjunction with this object class.

A very good introduction in 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`.

21.8.3 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 prefix (#) are inactive. This comment character must be removed to activate them.

Global Directives in slapd.conf

Example 21.18: *slapd.conf: Include Directive for Schemes*

```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/inetorgperson.schema
```

This first directive in `slapd.conf`, shown in Example 21.18, specifies the scheme by which the LDAP directory is organized. The entry `core.schema` is compulsory. Additionally required schemes are appended to this directive (`inetorgperson.schema` has been added here as an example). More available schemes can be found in the directory `/etc/openldap/schema`. For replacing NIS with an analogous LDAP service, include the two schemes `rfc2307.schema` and `cosine.schema`. Information can be found in the included OpenLDAP documentation.

Example 21.19: *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 21.20: *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 21.20 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 into 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 explicitly be protected with separate rules. It is also possible to process whole 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 21.10 on the next page are possible.

Table 21.10: *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. It is distinguished from the options listed below in Table 21.11.

Table 21.11: *Types of Access*

Tag	Scope of Access
`none`	no access
`auth`	for contacting the server
`compare`	to objects for comparison access
`search`	for the employment of search filters
`read`	read access
`write`	write access

slapd compares the access right requested by the client with those granted in `slapd.conf`. The client is granted access if the rules allow a higher or equal right than the requested one. If the client requests higher rights than those declared in the rules, it is denied access.

Example 21.21 shows a simple example for a simple access control that can be arbitrarily developed using regular expressions.

Example 21.21: *slapd.conf: Example for Access Control*

```
access to  dn.regex="ou=([^,]+),dc=suse,dc=de"
by dn.regex="cn=administrator,ou=$1,dc=suse,dc=de" 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.

Note

Establishing Access Rules

If there is no `access to` rule or no matching `by who` 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.

Note

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 ACI, access control information. 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.

Database-Specific Directives in slapd.conf

Example 21.22: *slapd.conf: Database-Specific Directives*

```
database ldbm
suffix "dc=suse,dc=de"
rootdn "cn=admin,dc=suse,dc=de"
# Cleartext passwords, especially for the rootdn, should
# be avoided.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw secret
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd/tools. Mode 700 recommended.
directory /var/lib/ldap
# Indices to maintain
index   objectClass     eq
```

The type of database, LDBM in this case, is determined in the first line of this section (see Example 21.22). The second line determines, with `suffix`, for which portion of the LDAP tree this server should be responsible.

The following `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. The administrator password is set with `rootpw`. Instead of using `secret` here, it is possible to enter the hash of the administrator password created by slappasswd. The `directory` directive indicates the directory (in the file system) where the database directories are stored on the server. The last directive, `index objectClass eq`, results in the maintenance of an index over all object classes. Attributes for which users search most often can be added here according to experience. Custom `Access` rules defined here for the database are used instead of the global `Access` rules.

Starting and Stopping the Servers

Once the LDAP server is fully configured and all desired entries have been made according to the pattern described below in Section 21.8.4, 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 11.5 on page 266, 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 starting and stopping scripts with the `insserv` command from a command prompt as described in Section 11.4.1 on page 265.

21.8.4 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.

Inserting Data into an LDAP Directory

Once the configuration of your LDAP server in `/etc/openldap/lsapd.conf` is correct and ready to go, meaning that 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) to accomplish this. An LDIF file is a simple text file that can contain an arbitrary number of pairs of attribute and value. 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 21.25 on page 487 would look like that in File 21.23.

Example 21.23: *Example for an LDIF File*

```
# The SUSE Organization
dn: dc=suse,dc=de
objectClass: dcObject
objectClass: organization
o: SUSE AG dc: suse

# The organizational unit development (devel)
dn: ou=devel,dc=suse,dc=de
objectClass: organizationalUnit
ou: devel

# The organizational unit documentation (doc)
dn: ou=doc,dc=suse,dc=de
objectClass: organizationalUnit
ou: doc

# The organizational unit internal IT (it)
dn: ou=it,dc=suse,dc=de
objectClass: organizationalUnit
ou: it
```

Note

Encoding of LDIF Files

LDAP works with UTF-8 (Unicode). Umlauts therefore must be encoded correctly. Use an editor that supports UTF-8 (such as Kate or recent versions of Emacs). Otherwise, it would be necessary to avoid umlauts and other special characters or to use recode to recode the input to UTF-8.

Note

The file is saved with the `.ldif` suffix and is passed to the server with the following command:

```
ldapadd -x -D <dn of the administrator> -W -f <file>.ldif
```

The first option `-x` switches off the authentication with SASL in this case. The `-D` switch 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 would be `cn=admin,dc=suse,dc=de`. The switch `-W` circumvents entering the password on the command line (in clear text) and activates a separate password requesting prompt. This password was previously determined in `slapd.conf` with `rootpw`. The `-f` switch passes the file name. See the details of running `ldapadd` in Example 21.24 on the following page.

Example 21.24: *ldapadd with example.ldif*

```
ldapadd -x -D cn=admin,dc=suse,dc=de -W -f example.ldif

Enter LDAP password: adding new entry "dc=suse,dc=de"
adding new entry "ou=devel,dc=suse,dc=de"
adding new entry "ou=doc,dc=suse,dc=de"
adding new entry "ou=it,dc=suse,dc=de"
```

The user data of the individual colleagues can be prepared in separate LDIF files. The following example, shown in Example 21.25, adds the colleague `Tux` to the new LDAP directory.

Example 21.25: *LDIF Data for Tux*

```
# coworker Tux
dn: cn=Tux Linux,ou=devel,dc=suse,dc=de
objectClass: inetOrgPerson
cn: Tux Linux
givenName: Tux
sn: Linux
mail: tux@suse.de
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.

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`, the LDIF file must be edited like in Example 21.26 on the facing page.

Example 21.26: *Modified LDIF File tux.ldif*

```
# coworker Tux
dn: cn=Tux Linux,ou=devel,dc=suse,dc=de
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=admin,dc=suse,dc=de -W -f tux.ldif
```

Alternatively, pass the attributes to change directly to `ldapmodify`. The procedure for this is described below:

- Start `ldapmodify` and enter your password:

  ```
  ldapmodify -x -D cn=admin,dc=suse,dc=de -W
  Enter LDAP password:
  ```

- Enter the changes while carefully complying with the syntax in the order it is presented below:

  ```
  dn: cn=Tux Linux,ou=devel,dc=suse,dc=de
  changetype: modify
  replace: telephoneNumber
  telephoneNumber: +49 1234 567-10
  ```

Read detailed information about `ldapmodify` and its syntax in its corresponding man page.

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=suse,dc=de "(objectClass=*)"
```

The option `-b` determines the search base — the section of the tree within which the search should be performed. In the current case, this is `dc=suse,dc=de`. To perform a more finely-grained search in specific subsections of the LDAP directory (for instance, only within the `devel` department), pass this section to ldapsearch with `-b`. The `-x` switch requests the 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. More information about the use of `ldapsearch` can be found in the corresponding man page (`man ldapsearch`).

Deleting Data from an LDAP Directory

Delete unwanted entries with `ldapdelete`. The syntax is similar to that of the commands described above. To delete, for example, the complete entry for `Tux Linux`, issue the following command:

```
ldapdelete -x -D cn=admin,dc=suse,dc=de -W cn=Tux \
Linux,ou=devel,dc=suse,dc=de
```

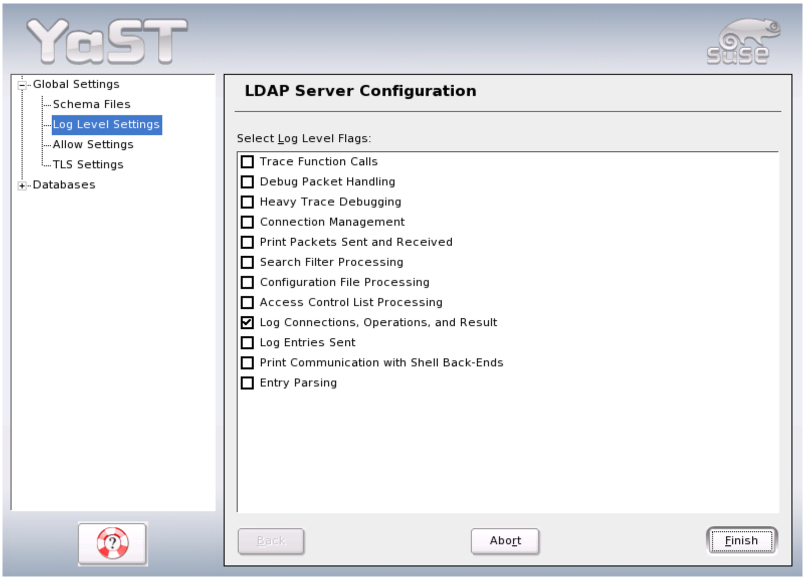
21.8.5 LDAP Server Configuration with YaST

To set up an LDAP server, you can also use YaST Such a server can not only handle user account data, but also manage other information, such as the configuration of mail, DNS and DHCP servers. Setting up the server for the latter purposes forms part of the installation procedure. The present section describes the configuration of an LDAP server with the corresponding YaST module, which can be started by selecting 'Network Services' → 'LDAP Server'.

In the dialog that opens, you can decide whether the LDAP server should be started or not. Selecting the 'Configure' button will then take you to the actual configuration dialogs (see Figure 21.26 on the next page).

In the left part of the window, you will see a tree view which lets you select the configurable features of the LDAP server. They include the 'Global Settings' ('Loglevel Settings', 'Allow Settings', and 'TLS Settings') on the one hand, and the 'Databases' on the other. The right part of the window displays the configuration dialog which corresponds to the currently selected item in the tree.

Selecting 'Loglevel Settings' allows you to configure the degree of logging activity (verbosity) of the LDAP server. From the pre-defined list, select or



LDAP Server Configuration

Select Log Level Flags:

☐ Trace Function Calls
☐ Debug Packet Handling
☐ Heavy Trace Debugging
☐ Connection Management
☐ Print Packets Sent and Received
☐ Search Filter Processing
☐ Configuration File Processing
☐ Access Control List Processing
☑ Log Connections, Operations, and Result
☐ Log Entries Sent
☐ Print Communication with Shell Back-Ends
☐ Entry Parsing

Figure 21.26: *YaST OpenLDAP Server Configuration: Loglevel*

deselect the logging options according to your needs. The more options are enabled, the larger your log files will grow.

The 'Allow Settings' dialog lets you define which connection types should be allowed by the LDAP server (see Figure 21.27 on the following page).

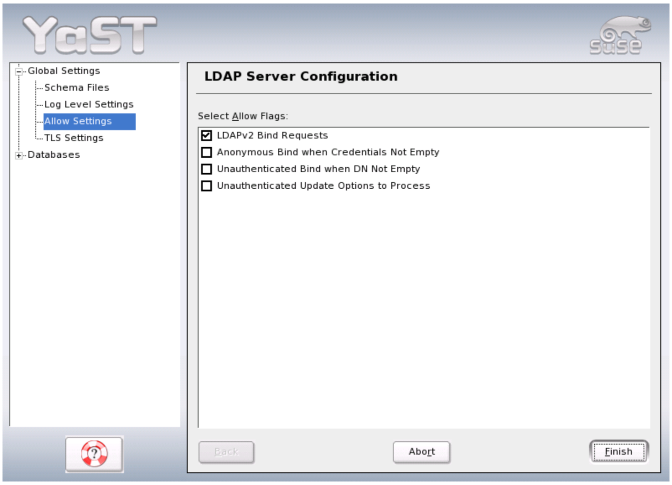
The individual 'Allow Flags' have the following meaning:

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 will deny any authentication attempts with empty credentials (DN or password). Enabling this option, however, makes it possible to connect with a password and no DN, in order to establish an anonymous connection.

bind_anon_dn Enabling this option makes it possible to connect in a non-authenticated (anonymous) fashion using a DN but no password.

update_anon Enabling this option allows non-authenticated (anonymous) update operations. Access is restricted according to ACLs and other rules (see Section 21.8.3 on page 489).



☑ LDAPv2 Bind Requests
☐ Anonymous Bind when Credentials Not Empty
☐ Unauthenticated Bind when DN Not Empty
☐ Unauthenticated Update Options to Process

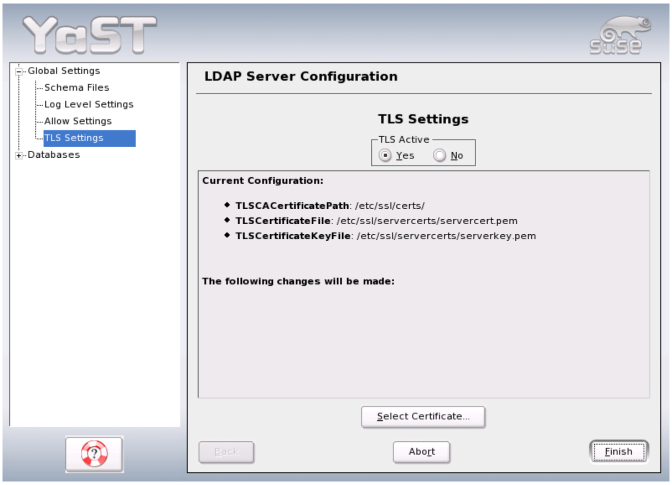
Figure 21.27: *YaST OpenLDAP Server Configuration: Allow*

After changing the 'Allow Flags', you can proceed to the configuration of the 'TLS Settings', to define how the data traffic between server and client should be secured (see Figure 21.28 on the facing page).

First use the radiobuttons to decide whether the data traffic between server and client should be TLS/SSL encrypted. Then use the 'Select Certificate...' to choose a certificate. In the popup dialog opened (see Figure 21.29 on the next page), you can select the type of certificate to be used: either the certificate automatically created by YaST during the installation of SUSE LINUX Enterprise Server ('Use Common Server Certificate'), or a certificate imported from an external source ('Import Certificate'). You will be directly taken to an import dialog if no common server certificate has been created during the installation.

If you decide to import a certificate, YaST will prompt you to specify the name and the path of the corresponding file, its key file, and the CA certificate (see Figure 21.30 on page 502). After entering these data, leave the dialog by selecting 'Ok'.

After completing the global configuration of the LDAP server, configure the databases to be managed by the server. To do so, select 'Databases' in the tree. The right part of the window should now display a list of the available databases. To add a new one, select 'Add Database' (see Figure 21.31 on page 503).


YaST
Global Settings
Schema Files
Log Level Settings
Allow Settings
TLS Settings
Databases
Select Certificate...
Back
Abort
Finish

LDAP Server Configuration

TLS Settings

TLS Active: ● Yes ○ No

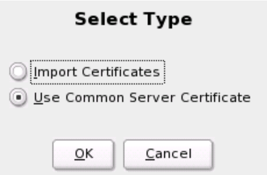
Current Configuration:

- **TLSCACertificatePath**: /etc/ssl/certs/
- **TLSCertificateFile**: /etc/ssl/servercerts/servercert.pem
- **TLSCertificateKeyFile**: /etc/ssl/servercerts/serverkey.pem

The following changes will be made:

Figure 21.28: *YaST OpenLDAP Server Configuration: TLS*

YaST will now show you a dialog where you need to provide the necessary entries (see Figure 21.32 on page 504). In the 'Base DN' field, enter the base DN of your LDAP server and in the 'Root DN' field, enter the DN of the administrator in charge of the server. If you check the 'Append Base DN' option, you only need to provide the `cn` of the administrator, and the system will fill in the rest automatically. Lastly, enter the root password for the server administrator, and select the algorithm to be used for password encryption (either '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



Select Type

○ Import Certificates
● Use Common Server Certificate

Figure 21.29: *YaST OpenLDAP Server Configuration: Selecting a Certificate*

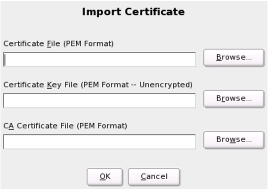

Import Certificate
Certificate File (PEM Format)
Browse...
Certificate Key File (PEM Format -- Unencrypted)
Browse...
CA Certificate File (PEM Format)
Browse...
OK
Cancel

Figure 21.30: *YaST OpenLDAP Server Configuration: Importing a Certificate*

to the previous dialog, select 'OK'.

To edit a previously created database, select its base DN in the tree to the left. In right part of the window, YaST will display a dialog similar to the one used for the creation of a new database — with the main difference that the base ID is not supposed to be changed and therefore greyed out (see Figure 21.33 on page 505).

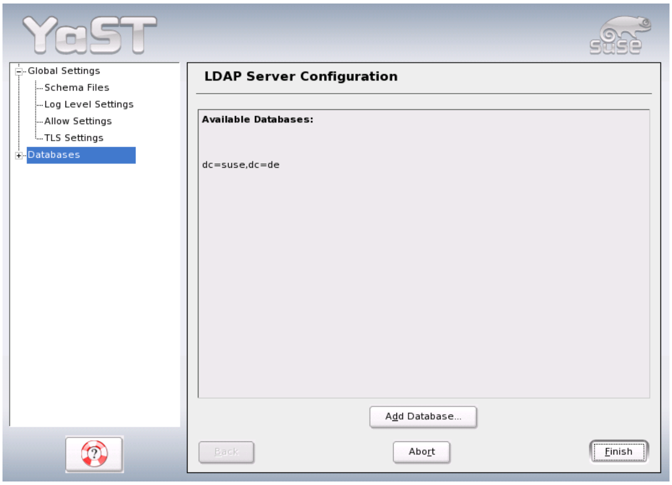
After leaving this dialog by selecting 'Quit', you are ready to go with a basic working configuration for your LDAP server. To fine-tune this setup, you can edit the file `/etc/openldap/slapd.conf` accordingly and then restart the server.

21.8.6 The YaST LDAP Client

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 will automatically enable any PAM and NSS related changes as required by LDAP and described below, and install the necessary files.

Standard Procedure

The processes acting in the background of a client machine must be known to understand the workings of the YaST LDAP client module. 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 21.27 on the facing page).


YaST
Global Settings
Schema Files
Log Level Settings
Allow Settings
TLS Settings
Databases
Add Database...
Back
Abort
Finish

LDAP Server Configuration

Available Databases:

dc=suse,dc=de

Figure 21.31: *YaST OpenLDAP Server Configuration: Adding a Database*

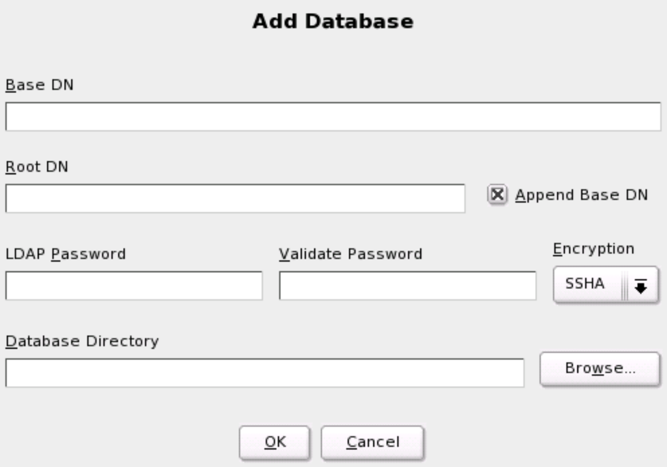
Example 21.27: *pam_unix2.conf Adapted to LDAP*

```
auth:        use_ldap nullok
account:     use_ldap
password:    use_ldap nullok
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/`.

The name resolution of `glibc` 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. More about the workings of `nsswitch.conf` can be found in Section 21.3.1 on page 430. The following lines must be present in `nsswitch.conf` for user administration and authentication with LDAP (compare with Example 21.28):

Example 21.28: *Adaptations in nsswitch.conf*



☒ Append Base DN

Encryption: SSHA

Figure 21.32: *YaST OpenLDAP Server Configuration: New Database*

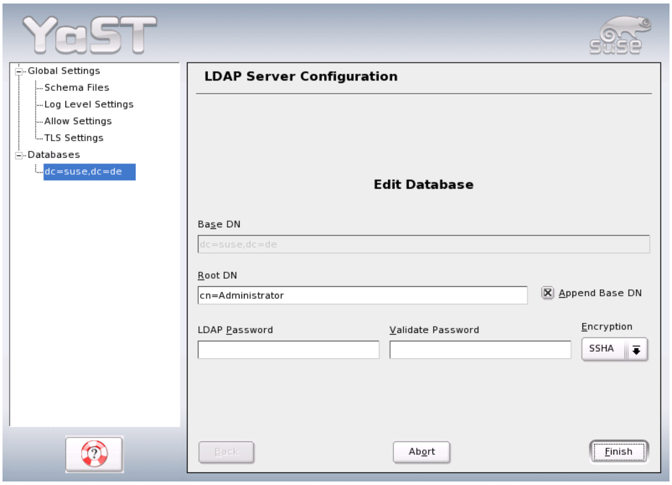
```
passwd: compat
group: compat

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 that regular users managed through LDAP can log in to the server with ssh or login, the files `/etc/passwd` and `/etc/group` each need to include an additional line. In `/etc/passwd`, this is the line `+::::::/sbin/nologin` and in `/etc/group`, it is `+:::` .

Configuration of the LDAP Client

After `nss_ldap` and `pam_ldap`, as well as `/etc/passwd` and `/etc/group` have been modified by YaST in the required way, the actual configuration work can begin on the first YaST input form (see Figure 21.34 on page 506).



LDAP Server Configuration

Edit Database

Base DN: dc=suse,dc=de

Root DN: cn=Administrator ☒ Append Base DN

LDAP Password:

Validate Password:

Encryption: SSHA

Figure 21.33: YaST OpenLDAP Server Configuration: Editing a Database

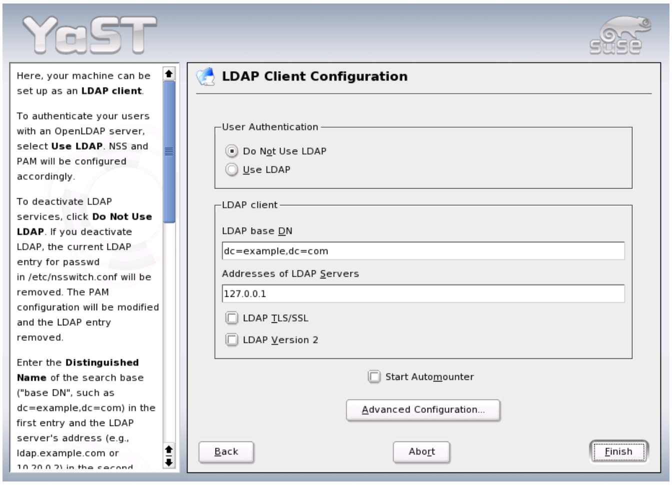
Activate the use of LDAP for user authentication via radio button in the first dialog. Enter the search base on the server below which all data is stored on the LDAP server in 'LDAP base DN'. Enter the address at which the LDAP server can be reached in 'Addresses of LDAP Servers'. If the server supports StartTLS, check 'LDAP TLS/SSL' to activate encrypted communication between the client and the server. To automatically mount directories on remote hosts, select the 'Start Automounter' checkbox. To modify data on the server as administrator, click 'Advanced Configuration' (see Figure 21.35 on page 507).

The next dialog has two parts: In the upper area, you can set general options for users and groups, as reflected by the YaST user module. In the lower area, you need to provide the data required to obtain access to the LDAP server. The user and group settings comprise the following items:

File Server If the current system is a file server, with `/home` containing individual users' directories, enabling the option will ensure that the YaST module deals with the user directories in the proper way.

Allow Login of LDAP Users Enable this option to give the users administered through LDAP permission to log in on the system.

Group Member Attribute This lets you specify the type of LDAP group to be used, either 'member' (default setting) or 'uniquemember'.



Here, your machine can be set up as an **LDAP client**.

To authenticate your users with an OpenLDAP server, select **Use LDAP**. NSS and PAM will be configured accordingly.

To deactivate LDAP services, click **Do Not Use LDAP**. If you deactivate LDAP, the current LDAP entry for passwd in /etc/nsswitch.conf will be removed. The PAM configuration will be modified and the LDAP entry removed.

Enter the **Distinguished Name** of the search base ("base DN", such as dc=example,dc=com) in the first entry and the LDAP server's address (e.g., ldap.example.com or 10.20.0.2) in the second

LDAP Client Configuration

User Authentication

● Do Not Use LDAP

○ Use LDAP

LDAP client

LDAP base DN: dc=example,dc=com

Addresses of LDAP Servers: 127.0.0.1

☐ LDAP TLS/SSL

☐ LDAP Version 2

☐ Start Automounter

Figure 21.34: *YaST: Configuration of the LDAP Client*

Enter the required access data for modifying configurations on the LDAP server here. These are 'Configuration Base DN' below which all configuration objects are stored and 'Bind DN'. In this case the BIND DN is your administrator DN.

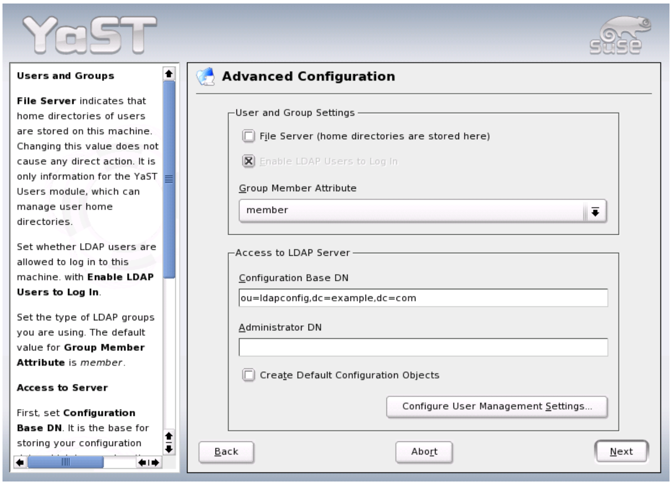
Click 'Configure Settings Stored on Server' to edit entries on the LDAP server. In the pop-up that appears, enter your LDAP password for authentication with the server. Access to the configuration modules on the server is then granted according to the ACLs and ACIs stored on the server.

Note

Employing the YaST Client

Use the YaST LDAP client to adapt the YaST modules for user and group administration and to extend them as needed. It is furthermore possible to define templates with default values for the individual attributes to simplify the actual registration of the data. The presets created here are stored themselves as LDAP objects in the LDAP directory. The registration of user data is still done with the regular YaST module input forms. The registered information is stored as objects in the LDAP directory.

Note



Users and Groups

File Server indicates that home directories of users are stored on this machine. Changing this value does not cause any direct action. It is only information for the YaST Users module, which can manage user home directories.

Set whether LDAP users are allowed to log in to this machine. with **Enable LDAP Users to Log In**.

Set the type of LDAP groups you are using. The default value for **Group Member Attribute** is *member*.

Access to Server

First, set **Configuration Base DN**. It is the base for storing your configuration

Advanced Configuration

User and Group Settings

☐ File Server (home directories are stored here)

☒ Enable LDAP Users to Log In

Group Member Attribute: member

Access to LDAP Server

Configuration Base DN: ou=ldapconfig,dc=example,dc=com

Administrator DN:

☐ Create Default Configuration Objects

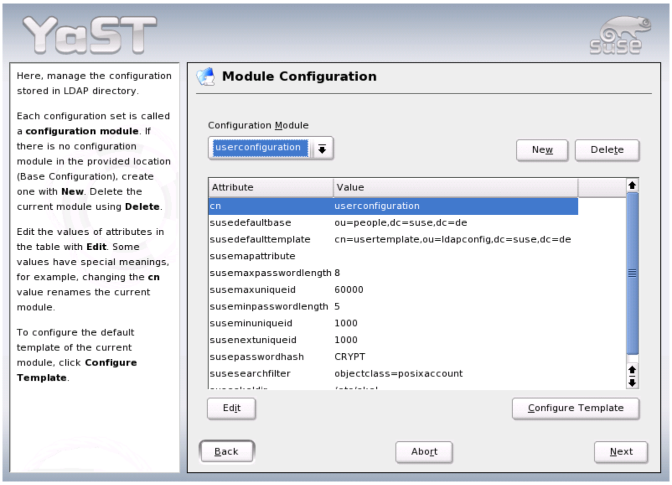
Figure 21.35: *YaST: Advanced Configuration*

The dialog for module configuration (Figure 21.36 on the next page) allows selection and modification existing configuration modules, creation of new modules, and design and modification of templates for such modules. To modify a value in a configuration module or rename a module, select the module type in the combobox above the content view of the current module. 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.

To copy a module, it is only necessary to change `cn`. To modify individual attribute values, select them from the content list then click ‘Edit’. A dialog window opens in which to change all settings belonging to the attribute. The changes are accepted with ‘OK’.

If a new module should be added to the existing modules, click ‘New’, located above the content overview. Enter the name and the object class of the new module in the dialog that appears (either `suseuserconfiguration` or `susegroupconfiguration`). When the dialog is closed with ‘OK’, the new module is added to the selection list of the existing modules and can then be selected or deselected in the combobox. Clicking ‘Delete’ deletes the currently selected module.

The YaST modules for group and user administration embed templates



Here, manage the configuration stored in LDAP directory.

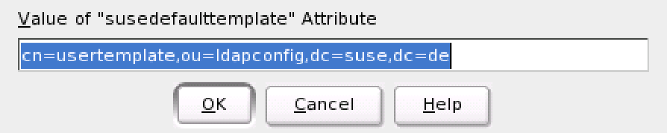
Each configuration set is called a **configuration module**. If there is no configuration module in the provided location (Base Configuration), create one with **New**. Delete the current module using **Delete**.

Edit the values of attributes in the table with **Edit**. Some values have special meanings, for example, changing the **cn** value renames the current module.

To configure the default template of the current module, click **Configure Template**.

Attribute	Value
cn	userconfiguration
susedefaultbase	ou=people,dc=suse,dc=de
susedefaulttemplate	cn=usertemplate,ou=ldapconfig,dc=suse,dc=de
susemapattribute	
susemaxpasswordlength	8
susemaxuniqueid	60000
suseminpasswordlength	5
suseminuniqueid	1000
susenextuniqueid	1000
susepasswordhash	CRYPT
susesearchfilter	objectclass=posixaccount

Figure 21.36: *YaST: Module Configuration*

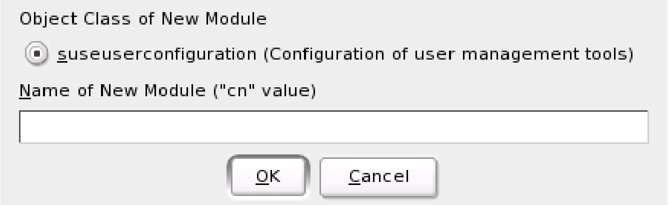


Value of "susedefaulttemplate" Attribute: cn=usertemplate,ou=ldapconfig,dc=suse,dc=de

Figure 21.37: *YaST: Changing Attributes in the Module Configuration*

with sensible standard values, if these were previously defined with the YaST LDAP clients. To edit a template as desired, click 'Configure Template'. The drop-down menu either contains already existing, modifiable templates or an empty entry. Select one and configure the properties of this template in the 'Object Template Configuration' form (see Figure 21.39 on page 510). This form is subdivided into two overview windows in table form. The upper window lists all general template attributes. Determine the values according to your mission scenario or leave some of them empty. Empty attributes are deleted on the LDAP server.

The second view ('Default Values for New Objects') lists all attributes of the corresponding LDAP object (in this case group or user configuration) for which a standard value is defined. Additional attributes and their stan-


Object Class of New Module

● suseuserconfiguration (Configuration of user management tools)

Name of New Module ("cn" value)

Figure 21.38: *YaST: Creating a New Module*

dard values can be added, existing attribute-value pairs can be edited, and entire attributes can be deleted. Copy a template by changing the `cn` entry. Connect the template to its module, as already described, by setting the `susedefaulttemplate` attribute value of the module to the DN of the adapted template.

Note

The default values for an attribute can be created from other attributes by using a variable style 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`.

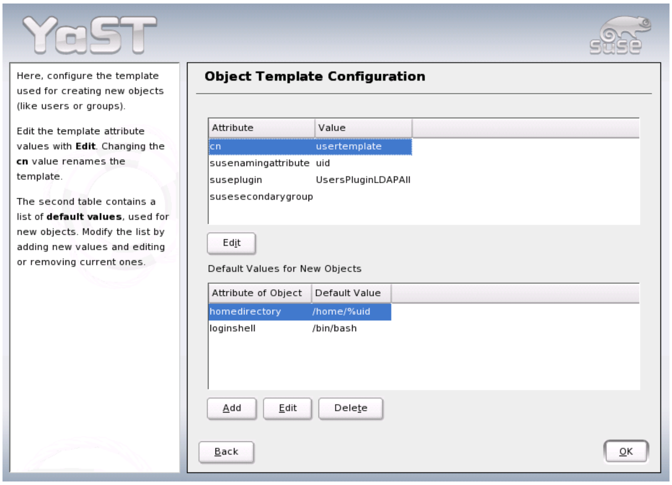
Note

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.

Users and Groups — Configuration with 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.

Access the YaST user administration with ‘Security & Users’ → ‘User Administration’. An input form is displayed for the registration of the most important user data, like name, login, and password. ‘Details’ accesses a form for the configuration of group membership, login shell, and the home directory. The default values for the input fields have previously been defined with the procedure described in Section 21.8.6 on page 504. When LDAP is used, this form leads to another form for the registration of LDAP-specific attributes. It is shown in Figure 21.41 on page 512. Select all attributes for which to change the value then click ‘Edit’. Closing the form



Here, configure the template used for creating new objects (like users or groups).

Edit the template attribute values with **Edit**. Changing the **cn** value renames the template.

The second table contains a list of **default values**, used for new objects. Modify the list by adding new values and editing or removing current ones.

Object Template Configuration

Attribute	Value
cn	usertemplate
susenamingattribute	uid
suseplugin	UsersPluginLDAPAll
susesecondarygroup	

Default Values for New Objects

Attribute of Object	Default Value
homedirectory	/home/%uid
loginshell	/bin/bash

Figure 21.39: YaST: Configuration of an Object Template

that opens with with 'Continue' returns to the initial input form for user administration.

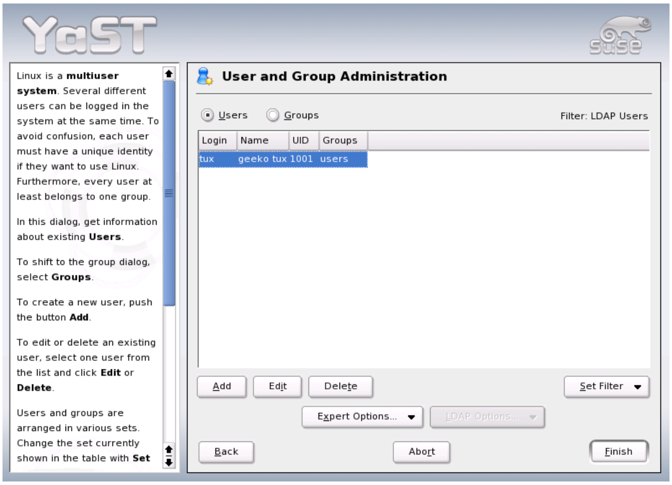
The initial input form of user administration, shown in Figure 21.40 on the next page, offers 'LDAP Options'. This gives the possibility to apply LDAP search filters to the set of available users, or to go to the module for the configuration of LDAP users and groups by selecting 'LDAP User and Group Configuration'.

21.8.7 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* (see below for references).

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 employment of OpenLDAP. `http://www.openldap.org/faq/data/cache/1.html`.



Linux is a **multiuser system**. Several different users can be logged in the system at the same time. To avoid confusion, each user must have a unique identity if they want to use Linux. Furthermore, every user at least belongs to one group.

In this dialog, get information about existing **Users**.

To shift to the group dialog, select **Groups**.

To create a new user, push the button **Add**.

To edit or delete an existing user, select one user from the list and click **Edit** or **Delete**.

Users and groups are arranged in various sets. Change the set currently shown in the table with **Set**

● Users ○ Groups

Filter: LDAP Users

Login	Name	UID	Groups
tux	geeko tux	1001	users

Figure 21.40: YaST: User Administration

Quick Start Guide Brief step-by-step instructions for installing your first LDAP server.

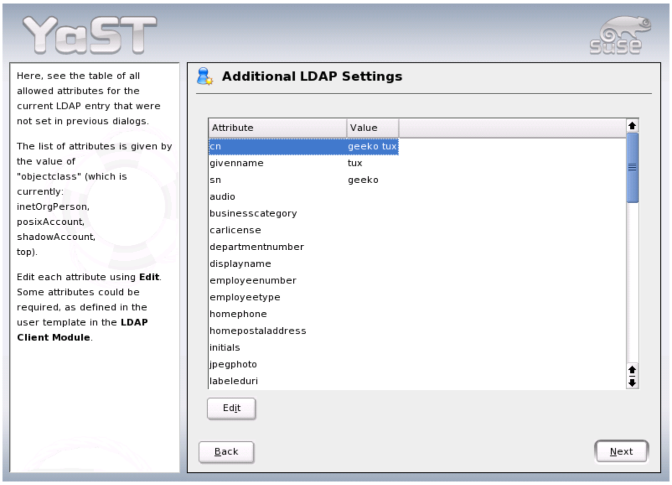
`http://www.openldap.org/doc/admin22/quickstart.html` or on an installed system in `/usr/share/doc/packages/openldap2/admin-guide/quickstart.html`

OpenLDAP 2.2 Administrator's Guide
A detailed introduction to all important aspects of LDAP configuration, including access controls and encryption. `http://www.openldap.org/doc/admin21/` or on an installed system in `/usr/share/doc/packages/openldap2/admin-guide/index.html`

The following redbooks from IBM exist regarding the subject of LDAP:

Understanding LDAP A detailed general introduction to the basic principles of LDAP: `http://www.redbooks.ibm.com/redbooks/pdfs/sg244986.pdf`.

LDAP Implementation Cookbook
The target audience consists of administrators of *IBM SecureWay Directory*. However, important general information about LDAP is also



Here, see the table of all allowed attributes for the current LDAP entry that were not set in previous dialogs.

The list of attributes is given by the value of "objectclass" (which is currently: inetOrgPerson, posixAccount, shadowAccount, top).

Edit each attribute using **Edit**. Some attributes could be required, as defined in the user template in the **LDAP Client Module**.

Additional LDAP Settings

Attribute	Value
cn	geeko tux
givenname	tux
sn	geeko
audio	
businesscategory	
carlicense	
departmentnumber	
displayname	
employeenumber	
employeetype	
homephone	
homepostaladdress	
initials	
jpegphoto	
labeleduri	

Figure 21.41: *YaST: Additional LDAP Settings*

contained here: `http://www.redbooks.ibm.com/redbooks/pdfs/sg245110.pdf`.

Printed literature about LDAP:

- Howes, Smith, and Good: *Understanding and Deploying LDAP Directory Services*. Addison-Wesley, 2nd ed., 2003. (ISBN 0-672-32316-8)
- Hodges: *LDAP System Administration*. O'Reilly & Associates, 2003. (ISBN 1-56592-491-6)

The ultimate reference material for the subject of LDAP is the corresponding RFCs (request for comments), 2251 to 2256.

21.9 NIS — Network Information Service

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 the user: whatever machine a user uses, he will always find himself in exactly the same environment. This is made possible by means of NIS and NFS services. NFS distributes file systems over a network and is discussed in Section 21.10 on page 518.

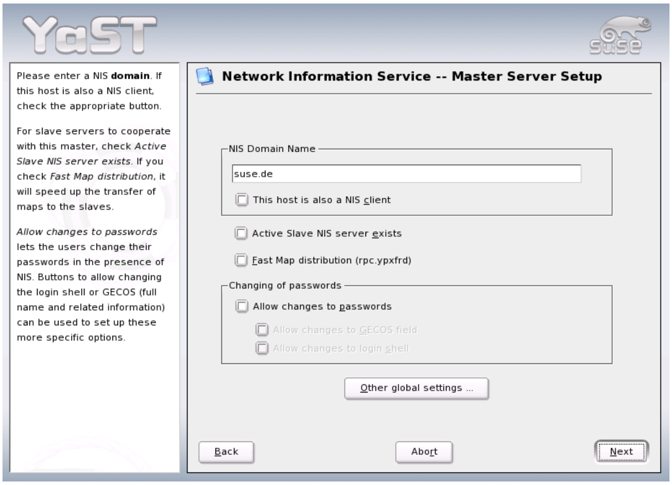
NIS (Network Information Service) can be described as a database-like service that provides access to the contents of `/etc/passwd`, `/etc/shadow`, and `/etc/group` across networks. NIS can also be used for other purposes (to make available the contents of files like `/etc/hosts` or `/etc/services`, for instance), but this is beyond the scope of this introduction. People often refer to NIS under the term of “YP”, which simply stands for the idea of the network’s “yellow pages”.

21.9.1 NIS Master and Slave Servers

For the configuration, select ‘NIS Server’ from the YaST module ‘Network Services’. If no NIS server existed so far in your network, activate ‘Install and set up a Master NIS Server’ in the next screen. If you already have a NIS server (a *master*), you can add a NIS slave server (for example, if you want to configure a new subnetwork). First, the configuration of the master server is described.

If some needed packages are missing, insert the respective CD or DVD as requested to install the packages automatically. Enter the domain name at the top of the configuration dialog, which is shown in Figure 21.42 on the next page. With the check box below, define whether the host should also be a NIS client, enabling users to log in and access data from the NIS server.

To configure additional NIS servers (*slave servers*) in your network afterwards, activate ‘Active Slave NIS Server Exists’ now. Select ‘Fast Map Distribution’ to set fast transfer of the database entries from the master to the slave server.



Please enter a NIS **domain**. If this host is also a NIS client, check the appropriate button.

For slave servers to cooperate with this master, check *Active Slave NIS server exists*. If you check *Fast Map distribution*, it will speed up the transfer of maps to the slaves.

Allow changes to passwords lets the users change their passwords in the presence of NIS. Buttons to allow changing the login shell or GECOS (full name and related information) can be used to set up these more specific options.

Network Information Service -- Master Server Setup

NIS Domain Name

suse.de

☐ This host is also a NIS client

☐ Active Slave NIS server exists

☐ Fast Map distribution (rpc.ypxfrd)

Changing of passwords

☐ Allow changes to passwords

☐ Allow changes to GECOS field

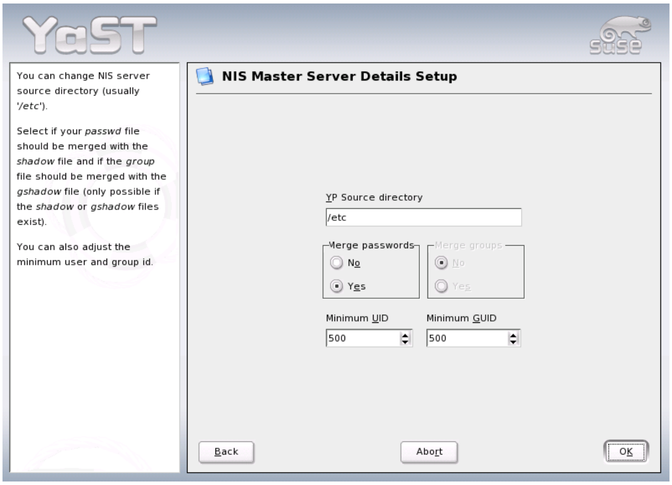
☐ Allow changes to login shell

Figure 21.42: *YaST: NIS Server Configuration Tool*

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`), activate the corresponding option. This makes '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.

By clicking 'Other Global Settings...', access a screen, shown in Figure 21.43 on the facing page, in which to change the source directory of the NIS server (`/etc/` by default). In addition, passwords and groups can be merged here. The setting should be 'Yes' so the files (`/etc/passwd`, `/etc/shadow`, and `/etc/group`) can be synchronized. Also determine the smallest user and group ID. Press 'OK' to confirm your settings and return to the previous screen. Then click 'Next'.

If you previously enabled 'Active Slave NIS Server Exists', enter the host names used as slaves and click 'Next'. 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, so normally they should be left unchanged.



You can change NIS server source directory (usually '/etc').

Select if your *passwd* file should be merged with the *shadow* file and if the *group* file should be merged with the *gshadow* file (only possible if the *shadow* or *gshadow* files exist).

You can also adjust the minimum user and group id.

YP Source directory: /etc

Merge passwords: ○ No ● Yes

Merge groups: ● No ○ Yes

Minimum UID: 500

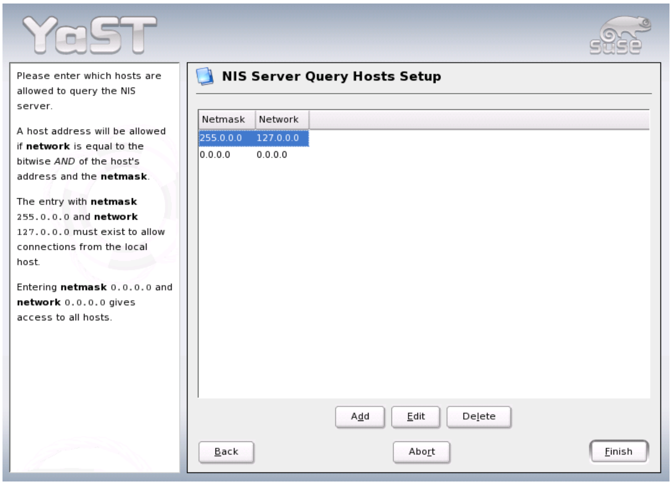
Minimum GUID: 500

Figure 21.43: *YaST: Changing the Directory and Synchronizing Files for a NIS Server*

'Next' continues to the last dialog, shown in Figure 21.44 on the next page. 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 one 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.



Please enter which hosts are allowed to query the NIS server.

A host address will be allowed if **network** is equal to the bitwise *AND* of the host's address and the **netmask**.

The entry with **netmask** `255.0.0.0` and **network** `127.0.0.0` must exist to allow connections from the local host.

Entering **netmask** `0.0.0.0` and **network** `0.0.0.0` gives access to all hosts.

NIS Server Query Hosts Setup

Netmask	Network
255.0.0.0	127.0.0.0
0.0.0.0	0.0.0.0

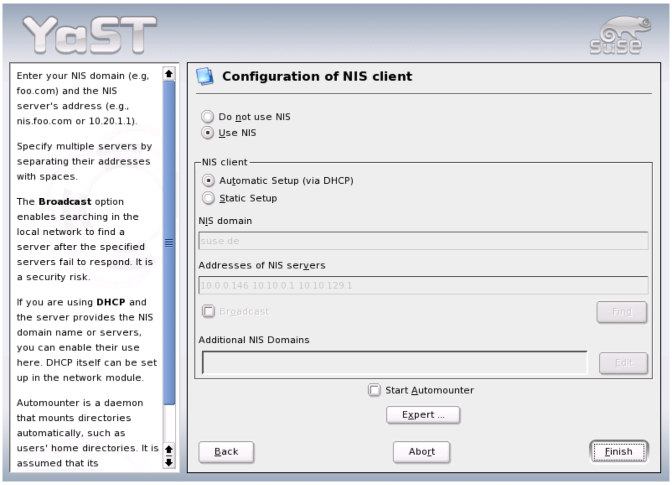
Figure 21.44: *YaST: Setting Request Permissions for a NIS Server*

21.9.2 The NIS Client Module of YaST

This module facilitates the configuration of the NIS client. After choosing to use NIS and, depending on the circumstances, the automounter, this dialog opens. Select whether the host has a fixed IP address or receives one issued by DHCP. DHCP also provides the NIS domain and the NIS server. For further information about DHCP, see Section 21.11 on page 522. If a static IP address is used, specify the NIS domain and the NIS server manually (see Figure 21.45 on the next page). The button 'Find' makes YaST search for an active NIS server in your network.

In addition, you can specify multiple domains with one default domain. Use 'Add' to specify multiple servers including the broadcast function for the individual domains.

In the expert settings, check 'Answer to the Local Host Only', 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`.



Enter your NIS domain (e.g. foo.com) and the NIS server's address (e.g., nis.foo.com or 10.20.1.1).

Specify multiple servers by separating their addresses with spaces.

The **Broadcast** option enables searching in the local network to find a server after the specified servers fail to respond. It is a security risk.

If you are using **DHCP** and the server provides the NIS domain name or servers, you can enable their use here. DHCP itself can be set up in the network module.

Automounter is a daemon that mounts directories automatically, such as users' home directories. It is assumed that its

Configuration of NIS client

○ Do not use NIS
● Use NIS

NIS client
● Automatic Setup (via DHCP)
○ Static Setup

NIS domain: suse.de

Addresses of NIS servers: 10.0.0.146 10.10.0.1 10.10.129.1

☐ Broadcast

Additional NIS Domains

☐ Start Automounter

Figure 21.45: Setting Domain and Address of NIS Server

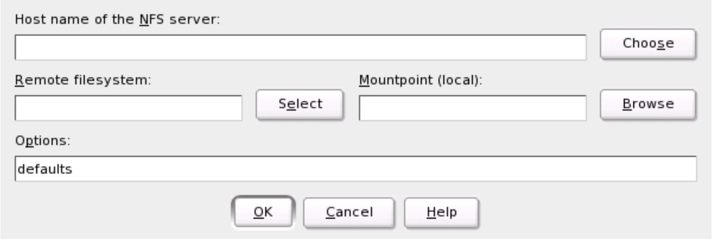
21.10 NFS — Shared File Systems

As mentioned in Section 21.9 on page 513, NFS (together with NIS) makes a network transparent to the user. With NFS, it is possible to distribute file systems over the network. It does not matter at which terminal a user is logged in. He will always find himself in the same environment.

As with NIS, NFS is an asymmetric service. There are NFS servers and NFS clients. A machine can be both — it can supply file systems over the network (export) and mount file systems from other hosts (import). Generally, these are servers with a very large hard disk capacity, whose file systems are mounted by other clients.

21.10.1 Importing File Systems with YaST

Any user authorized to do so can mount NFS directories from an NFS server into his own file tree. This can be achieved most easily using the YaST module 'NFS Client'. Just enter the host name of the NFS server, the directory to import, and the mount point at which to mount this directory locally. All this is done after clicking 'Add' in the first dialog (Figure 21.46).



Options: defaults

Figure 21.46: *NFS Client Configuration with YaST*

21.10.2 Importing File Systems Manually

File systems can easily be imported manually from an NFS server. The only prerequisite is a running RPC port mapper, which can be started by entering the command `rcportmap start` as `root`. Once this prerequisite is met, remote file systems exported on the respective machines can be mounted in the file system just like local hard disks using the command `mount` with the following syntax:

```
mount host:remote-path local-path
```

If user directories from the machine sun, for example, should be imported, the following command can be used:

```
mount sun:/home /home
```

21.10.3 Exporting File Systems with YaST

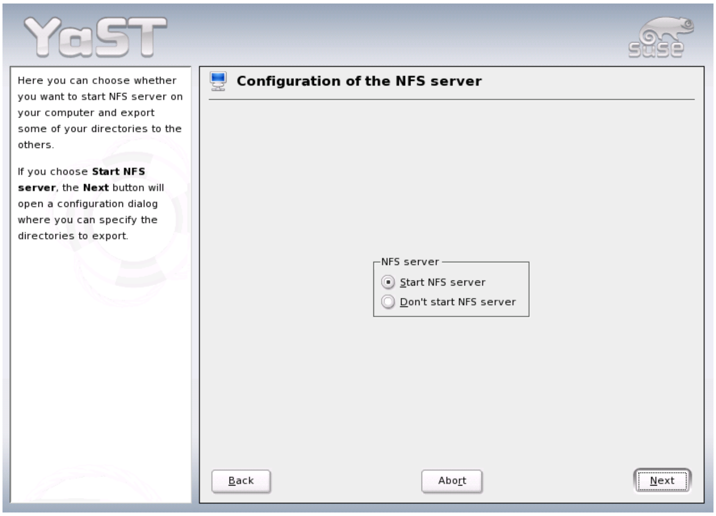
With YaST, turn a host in your network into an NFS server — a server that exports directories and files to all hosts granted access to it. This could be done to provide applications to all coworkers of a group without installing them locally on each and every host. To install such a server, start YaST and select 'Network Services' → 'NFS Server' (see Figure 21.47 on the following page).

Next, activate 'Start NFS Server' and 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 21.48 on page 522. There are four options that can be set for each host: `single host`, `netgroups`, `wildcards`, and `IP networks`. A more thorough explanation of these options is provided by `man exports`. 'Exit' completes the configuration.

21.10.4 Exporting File Systems Manually

If you do not want to use YaST, make sure the following systems run on the NFS server:

- RPC portmapper (portmap)
- RPC mount daemon (rpc.mountd)
- RPC NFS daemon (rpc.nfsd)



Here you can choose whether you want to start NFS server on your computer and export some of your directories to the others.

If you choose **Start NFS server**, the **Next** button will open a configuration dialog where you can specify the directories to export.

Configuration of the NFS server

NFS server

● Start NFS server

○ Don't start NFS server

Figure 21.47: *NFS Server Configuration Tool*

For these services to be started by the scripts `/etc/init.d/portmap` and `/etc/init.d/nfsserver` when the system is booted, enter the commands `insserv /etc/init.d/nfsserver` and `insserv /etc/init.d/portmap`. Also define which file systems should be exported to which host in the configuration file `/etc/exports`.

For each directory to export, one line is needed to set which machines may access that directory with what permissions. All subdirectories of this directory are automatically exported as well. Authorized machines are usually specified with their full names (including domain name), but it is possible to use wild cards like `*` or `?` (which expand the same way as in the Bash shell). If no machine is specified here, any machine is allowed to import this file system with the given permissions.

Set permissions for the file system to export in brackets after the machine name. The most important options are:

Table 21.12: *Permissions for Exported File System*

option	meaning
`ro`	File system is exported with read-only permission (default).

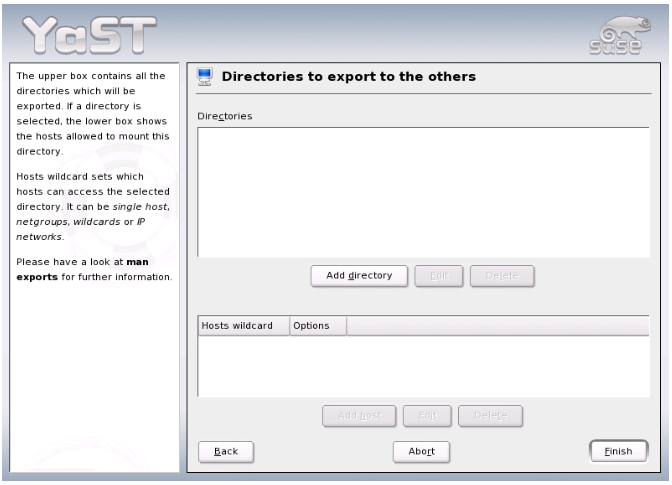
`rw`	File system is exported with read-write permission.
`root_squash`	This makes sure the user `root` of the given machine does not have `root` permissions on this file system. This is achieved by assigning user ID `65534` to users with user ID `0` (`root`). This user ID should be set to `nobody` (which is the default).
`no_root_squash`	Does not assign user ID `0` to user ID `65534`, keeping the `root` permissions valid.
`link_relative`	Converts absolute links (those beginning with `/`) to a sequence of `../`. This is only useful if the entire file system of a machine is mounted (default).
`link_absolute`	Symbolic links remain untouched.
`map_identity`	User IDs are exactly the same on both client and server (default).
`map_daemon`	Client and server do not have matching user IDs. This tells nfsd to create a conversion table for user IDs. The **ugidd** daemon is required for this to work.

Your `exports` file might look like File 21.29.

Example 21.29: */etc/exports*

```
#
# /etc/exports
#
/home            sun(rw)   venus(rw)
/usr/X11         sun(ro)   venus(ro)
/usr/lib/texmf   sun(ro)   venus(rw)
/                earth(ro,root_squash)
/home/ftp        (ro)
# End of exports
```

`/etc/exports` is read by mountd and nfsd. If you change anything in this file, restart mountd and nfsd for your changes to take effect. This can easily be done with `rcnfsserver restart`.


YaST
Directories to export to the others
Directories
Add directory
Edit
Delete
Hosts wildcard
Options
Add host
Edit
Delete
Back
Abort
Finish

The upper box contains all the directories which will be exported. If a directory is selected, the lower box shows the hosts allowed to mount this directory.

Hosts wildcard sets which hosts can access the selected directory. It can be *single host*, *netgroups*, *wildcards* or *IP networks*.

Please have a look at **man exports** for further information.

Figure 21.48: *Configuring an NFS Server with YaST*

21.11 DHCP

21.11.1 The DHCP Protocol

Note

s390, zseries: DHCP Support

On IBM's S/390 and zSeries platforms, DHCP only works on interfaces using the OSA and OSA Express network cards. These cards are the only ones with a MAC, which in turn is required for DHCP's autoconfiguration features.

Note

The purpose of the *Dynamic Host Configuration Protocol* is to assign network settings centrally from a server rather than configuring them locally on each and every workstation. A client 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.

One way to use DHCP is to identify each client using the hardware address of its network card (which is fixed in most cases) then supply that client with identical settings each time it connects to the server. DHCP can also be configured so the server assigns addresses to each interested host dynamically from an address pool set up for that purpose. In the latter case, the DHCP server will try to assign the same address to the client each time it receives a request from it (even over longer periods). This, of course, will not work if there are more client hosts in the network than network addresses available.

With these possibilities, DHCP can make life easier for system administrators in two ways. Any changes (even bigger ones) related to addresses and the network configuration in general can be implemented centrally by editing the server's configuration file. This is much more convenient than reconfiguring lots of client machines. Also it is much easier to integrate machines, particularly new machines, into the network, as they can be given an IP address from the pool. Retrieving the appropriate network settings from a DHCP server can be especially useful in the case of laptops regularly used in different networks.

A DHCP server not only supplies the IP address and the netmask, but also the host name, domain name, gateway, and name server addresses to be used by the client. In addition to that, DHCP allows for 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.

The following section gives an overview of DHCP without describing the service in every detail. In particular, we want to show how to use the DHCP server dhcpd in your own network to easily manage its entire setup from one central point.

21.11.2 DHCP Software Packages

Both a DHCP server and DHCP clients are available for SUSE LINUX. The DHCP server available is dhcpd (published by the Internet Software Consortium). On the client side, choose between two different DHCP client programs: dhclient (also from ISC) and the DHCP client daemon in the `dhcpcd` package.

SUSE LINUX installs dhcpcd by default. The program is very easy to handle and will be launched automatically on each system boot to watch for a DHCP server. It does not need a configuration file to do its job and should work out of the box in most standard setups. For more complex situations, use the ISC dhclient, which is controlled by means of the configuration file `/etc/dhclient.conf`.

21.11.3 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 as 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 a basic sample `/etc/dhcpd.conf` file:

Example 21.30: *The Configuration File /etc/dhcpd.conf*

```
default-lease-time 600;         # 10 minutes
max-lease-time 7200;            # 2  hours

option domain-name "kosmos.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 a semicolon is inserted at the end of each line, because otherwise dhcpd will not be started.

The above sample file can be divided into three sections. The first one defines how many seconds an IP address is leased to a requesting host 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 host names (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 host name for each dynamic address and vice versa. To learn how to configure your own name server, read Section 21.7 on page 455.
- The line `option broadcast-address` defines the broadcast address to be used by the requesting host.
- With `option routers`, tell the server where to 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 is there to define 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 our example, 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. You can also 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 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.

21.11.4 Hosts with Fixed IP Addresses

As mentioned above, DHCP can also be used to assign a predefined, static address to a specific host for each request. As might be expected, addresses assigned explicitly always take priority over addresses from the pool of dynamic addresses. Furthermore, a static address never expires in the way a dynamic address would, for example, if there were not enough addresses available so the server needed to redistribute them among hosts.

To identify a host 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 File 21.31, are added to the configuration file of File 21.30 on page 524, the DHCP daemon will assign the same set of data to the corresponding host under all circumstances.

Example 21.31: *Additions to the Configuration File*

```
host earth {
hardware ethernet 00:00:45:12:EE:F4;
fixed-address 192.168.1.21;
}
```

The structure of this entry is almost self-explanatory: The name of the respective host (`host` <host name>) is entered in the first line and the MAC address in the second line. On Linux hosts, this address can be determined with the command `ifstatus` followed by the network device (for example `eth0`). If necessary, activate the network card first: `ifup eth0`. The output should contain something like

```
link/ether 00:00:45:12:EE:F4
```

In the above example, a host 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 host name 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.

21.11.5 The SUSE LINUX Version

To improve security, the SUSE 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 permissions of `nobody`
- run in a chroot environment (`/var/lib/dhcp/`)

To make this possible, the configuration file `/etc/dhcpd.conf` needs to be located in `/var/lib/dhcp/etc/`. The corresponding init script automatically copies the file to this directory upon starting.

The server's behavior with regard to this feature can be controlled through the configuration file `/etc/sysconfig/dhcpd`. To continue running dhcpd without the chroot environment, set the variable `DHCPD_RUN_CHROOTED` in `/etc/sysconfig/dhcpd` to "no".

To enable dhcpd to resolve host names even from within the chroot environment, some other configuration files need to be copied as well, namely:

- `/etc/localtime`
- `/etc/host.conf`
- `/etc/hosts`
- `/etc/resolv.conf`

These files are copied to `/var/lib/dhcp/etc/` when starting the init script. These copies must be taken 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 host names).

If your configuration includes additional files that should be copied into the chroot environment, specify these under the variable `DHCPD_CONF_INCLUDE_FILES` in the file `etc/sysconfig/dhcpd`. To make sure the DHCP logging facility keeps working even after a restart of the syslog daemon, it is necessary to add the option `"-a /var/lib/dhcp/dev/log"` under `SYSLOGD_PARAMS` in the file `/etc/sysconfig/syslog`.

21.11.6 DHCP Configuration with YaST

Note

LDAP Support

In this version of the SUSE LINUX Enterprise Server, the DHCP server as configured with YaST can be set up to store the server configuration locally (on the host which runs the DHCP server), or alternatively to have its configuration data managed by an LDAP server.

Note

The DHCP module of YaST allows you to set up your own DHCP server for the local network. The module can work in two different modes:

Initial Configuration When starting the module for the first time, you will be prompted to make just a few basic decisions concerning the server administration. After completing this initial setup, the server is ready to go, with a configuration that should be suitable for most basic scenarios.

Expert Configuration This expert mode lets you configure more advanced settings, such as those related to dynamic DNS, TSIG management, and others.

Note

Navigating the Module, Obtaining Help

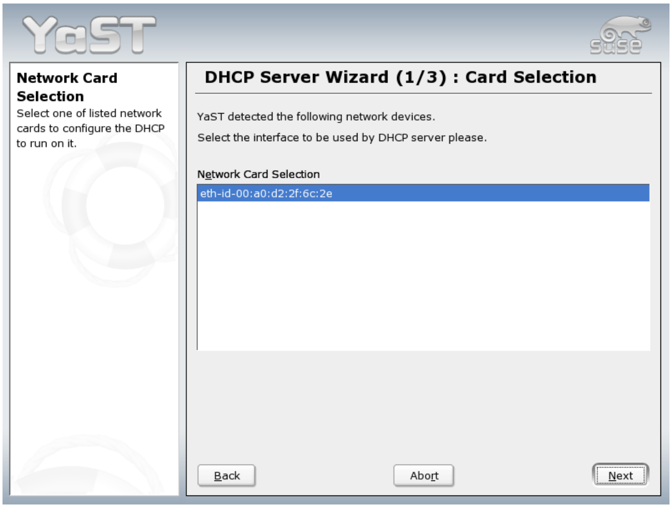
All dialogs of the DHCP module have a similar layout. The left part of the dialog window displays a tree view which lets you go to the individual sections of the configuration, while the corresponding configuration dialog is displayed to the right. To get help on the current dialog, click on the lifebelt icon at the left bottom of the window. To close the help window and go back to the tree, click on the icon depicting a tree structure.

Note

Initial Configuration

After launching the module for the first time, YaST starts a four-part configuration assistant. You can set up a basic DHCP server for your network by completing this assistant.

Selecting the Network Interface In the first step, YaST looks for the network interfaces available on your system, and then displays them in a list. From the list, select the interface on which the DHCP server shall listen. (see Figure 21.49).



Network Card Selection

Select one of listed network cards to configure the DHCP to run on it.

DHCP Server Wizard (1/3) : Card Selection

YaST detected the following network devices.

Select the interface to be used by DHCP server please.

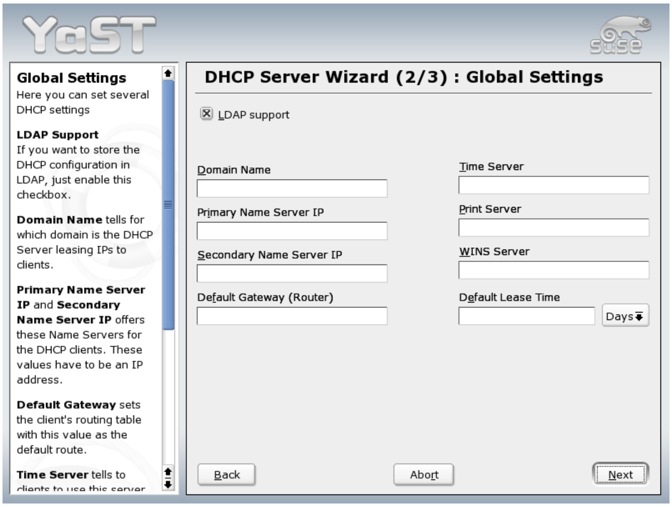
Network Card Selection

eth-id-00:a0:d2:2f:6c:2e

Figure 21.49: *DHCP Server: Selecting the Network Interface*

Global Settings A checkbox lets you define whether your DHCP settings should be automatically stored by an LDAP server. In the entry fields, you need to provide the network specifics for all of the clients to be managed by the DHCP server. These are the domain name, the address of a time server, the addresses of the primary and the secondary name server, the addresses of a print and a WINS server (in case you have a mixed network with both Windows and Linux clients), the gateway address, as well as the lease time (see Figure 21.50 on the next page).

Dynamic DHCP In this step, you can configure how dynamic IP addresses should by assigned to clients. To do so, you need to specify an IP range from which the server can assign addresses to DHCP clients. All these addresses must be covered by the same netmask. You also need to specify the lease time during which a client may keep its IP address without having to request an extension of the lease. Optionally, you may specify the maximum lease time, that is to say the pe-



Global Settings
Here you can set several DHCP settings

LDAP Support
If you want to store the DHCP configuration in LDAP, just enable this checkbox.

Domain Name tells for which domain is the DHCP Server leasing IPs to clients.

Primary Name Server IP and **Secondary Name Server IP** offers these Name Servers for the DHCP clients. These values have to be an IP address.

Default Gateway sets the client's routing table with this value as the default route.

Time Server tells to

DHCP Server Wizard (2/3) : Global Settings

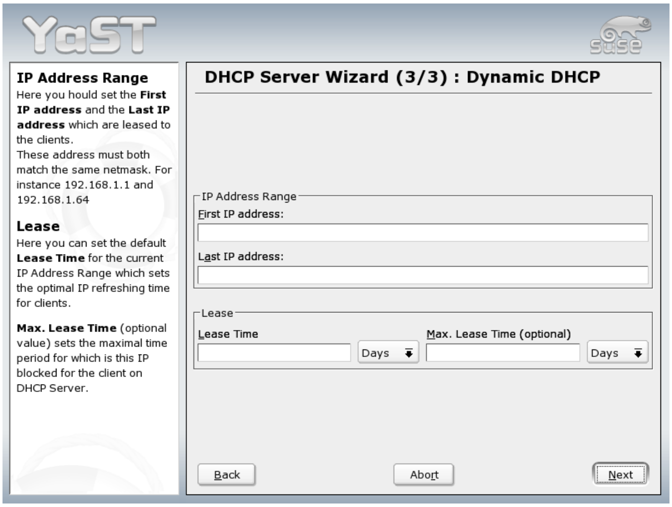
☒ LDAP support

Figure 21.50: *DHCP Server: Global Settings*

riod during which the server reserves an IP address for a particular client (see Figure 21.51 on the facing page).

Finishing the Configuration and Setting the Start Mode
After the third part of the configuration assistant, a last dialog will be shown where you can define how the DHCP server should be started. Selecting 'On' causes the DHCP to be started automatically as part of the boot procedure. If you select 'Off', the server needs to be started manually. To finish the server configuration, select 'Ok'. Alternatively, you can select 'Host Management' in the tree to the left, to go beyond the basic setup and add a special configuration for individual hosts (see Figure 21.52 on page 532).

Host Management Instead of using dynamic DHCP in the way described above, 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 hosts to be managed in this way. Specifically, you need to provide the 'Name' and the 'IP Address' to be given to such a host, the 'Hardware Address' and the 'Network Type' (Token Ring or Ethernet). You can modify the list of hosts, which is shown in the upper part, with the buttons 'Add', 'Edit', and 'Delete' (see Figure 21.52 on page 532).



IP Address Range
Here you hould set the **First IP address** and the **Last IP address** which are leased to the clients.
These address must both match the same netmask. For instance 192.168.1.1 and 192.168.1.64

Lease
Here you can set the default **Lease Time** for the current IP Address Range which sets the optimal IP refreshing time for clients.

Max. Lease Time (optional value) sets the maximal time period for which is this IP blocked for the client on DHCP Server.

Figure 21.51: *DHCP Server: Dynamic DHCP*

After having completed all the steps of the configuration assistant (with or without additional host management), select 'Ok' to apply the configuration and to start the server.

Expert Configuration

In addition to the configuration methods as discussed above, there is also an expert configuration mode which allows you to tweak the DHCP server setup in every detail. The expert configuration is started by selecting 'Expert settings' in the tree view in the left part of the dialog.

Chroot Environment and Declarations

In this first dialog, you need to 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, in order 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 will prevent 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. You can modify these with 'Add', 'Delete', and

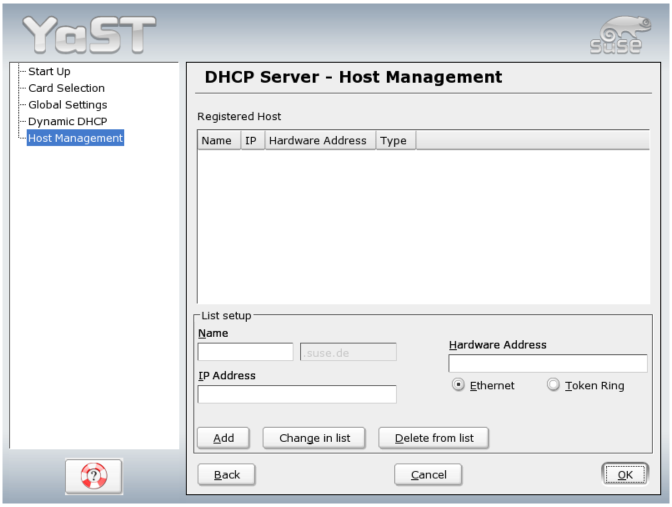

Start Up
Card Selection
Global Settings
Dynamic DHCP
Host Management
DHCP Server - Host Management
Registered Host
Name
IP
Hardware Address
Type
List setup
Name
.suse.de
IP Address
Hardware Address
● Ethernet
○ Token Ring
Add
Change in list
Delete from list
Back
Cancel
OK

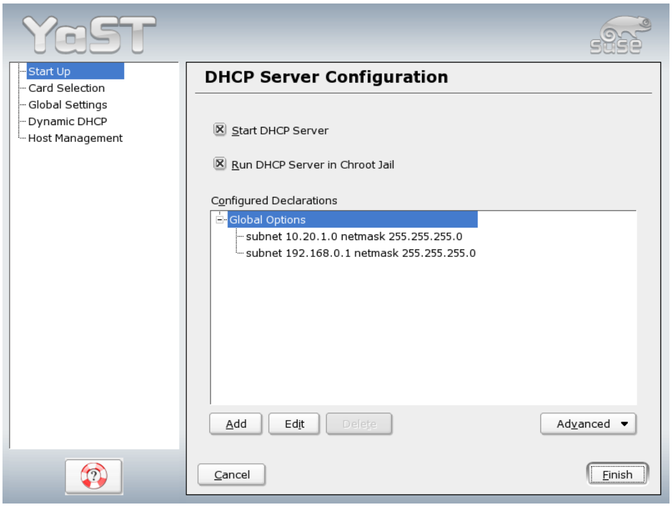
Figure 21.52: *DHCP Server: Host Management*

'Edit'. Selecting 'Advanced' takes you to additional expert dialogs. (see Figure 21.53 on the next page). After selecting 'Add', you need to define the type of the declaration to be added. The 'Advanced' button allows you to view the log file of the server, to configure TSIG key management, and to adjust the configuration of the firewall according to the setup of the DHCP server.

Selecting the Declaration Type The 'Global Options' of the DHCP server are made up of a number of so-called declarations. This dialog lets you set the declaration types 'Subnet', 'Host', 'Shared Network', 'Group', 'Pool of Addresses', and 'Class'. Our example shows the selection of a new subnetwork (see Figure 21.54 on page 534).

Subnet Configuration This dialog allows you specify a new subnet with its IP address and netmask. In the middle part of the dialog you can modify the DHCP server start options for the selected subnet, by using the buttons 'Add', 'Edit', and 'Delete'. If you want to set up dynamic DNS for the subnet, select the 'Dynamic DNS' button.

TSIG Key Management If you chose to configure dynamic DNS in the previous dialog (see Section 21.7.8 on page 466), you can now configure the key management for a secure zone transfer (see Fig-



DHCP Server Configuration

☒ Start DHCP Server

☒ Run DHCP Server in Chroot Jail

Configured Declarations

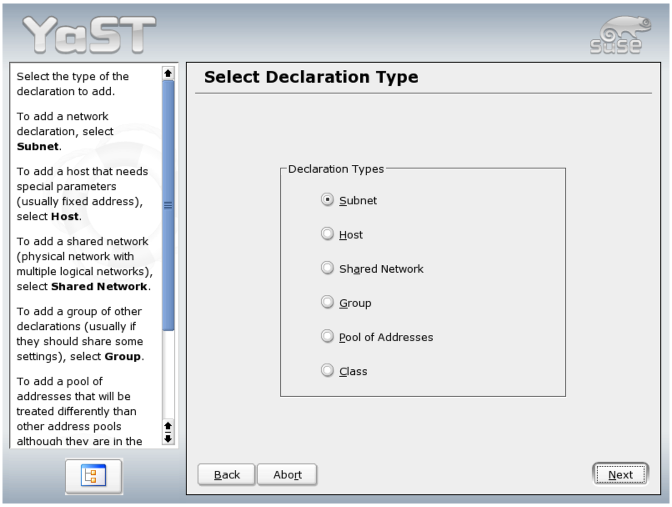
- Global Options
  - subnet 10.20.1.0 netmask 255.255.255.0
  - subnet 192.168.0.1 netmask 255.255.255.0

Figure 21.53: *DHCP Server: Chroot Jail and Declarations*

ure 21.56 on page 536. Selecting 'OK' takes you to another dialog to configure the interface for dynamic DNS (see Figure 21.57 on page 537).

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 dropdown menu to choose the TSIG keys for forward and reverse zones, making sure that the keys are the same for the DNS and the DHCP server. With the checkbox labeled 'Update Global Dynamic DNS Settings', you can enable the automatic update and adjustment of the global DHCP server settings according to the dynamic DNS environment. Lastly, you can 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' takes you back to the subnet configuration dialog (see Figure 21.55 on page 535). Selecting 'Ok' again takes you back to the original expert configuration dialog.

Network Interface Configuration To define the interfaces where the DHCP server should listen, and to adjust the firewall configuration,



Select the type of the declaration to add.

To add a network declaration, select **Subnet**.

To add a host that needs special parameters (usually fixed address), select **Host**.

To add a shared network (physical network with multiple logical networks), select **Shared Network**.

To add a group of other declarations (usually if they should share some settings), select **Group**.

To add a pool of addresses that will be treated differently than other address pools although they are in the

Select Declaration Type

Declaration Types

● Subnet
○ Host
○ Shared Network
○ Group
○ Pool of Addresses
○ Class

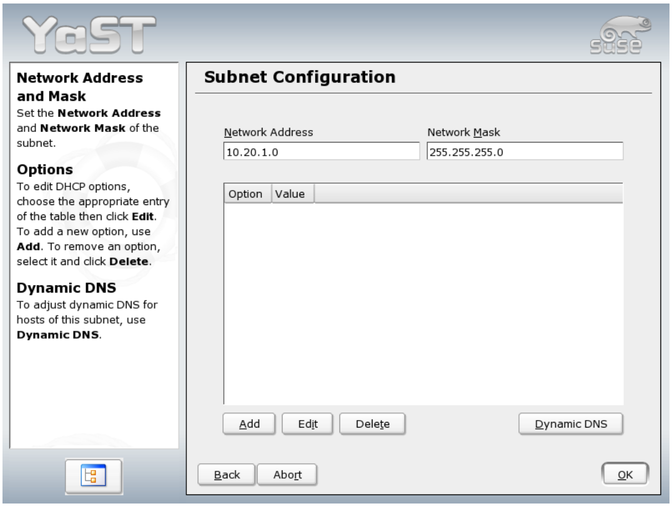
Figure 21.54: *DHCP Server: Selecting a Declaration Type*

select 'Advanced' → 'Interface Configuration' from the expert configuration dialog. From the list of interfaces displayed, select one or more which should be attended by the the DHCP server. If clients in all of the subnets shall be able to communicate with the server, and if the server host also runs a firewall, the latter needs to be adjusted accordingly. To do so, select the checkbox labeled 'Adapt Firewall Settings'. YaST will now adjust the rules of SuSEfirewall2 to the new conditions (see Figure 21.58 on page 538), after which you can go back to the original dialog by selecting 'Ok'.

After having completed all of the configuration steps, close the dialog with 'Ok'. The server is now started with its new configuration.

21.11.7 For More Information

For more information, the page of the *Internet Software Consortium* on the subject (`http://www.isc.org/products/DHCP/`) is a good source to read about the details of DHCP, including about version 3 of the protocol, currently in beta testing. Apart from that, you can always rely on the man pages for further help. Try `man dhcpd`, `man dhcpd.conf`,



Network Address and Mask
Set the **Network Address** and **Network Mask** of the subnet.

Options
To edit DHCP options, choose the appropriate entry of the table then click **Edit**. To add a new option, use **Add**. To remove an option, select it and click **Delete**.

Dynamic DNS
To adjust dynamic DNS for hosts of this subnet, use **Dynamic DNS**.

Subnet Configuration

Network Address: 10.20.1.0
Network Mask: 255.255.255.0

Option	Value

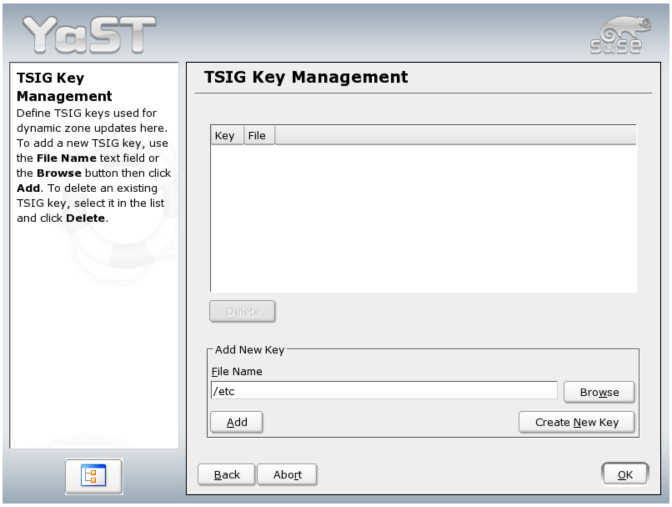
Figure 21.55: DHCP Server: Configuring Subnets

`man dhcpd.leases`, and `man dhcp-options`. Also, several books about DHCP have been published over the years that take an in-depth look at the topic.

With dhcpd, it is even possible to offer a file to a requesting host, as defined with the parameter `filename`, and for this file to contain a bootable Linux kernel. This allows you to build client hosts that do not need a hard disk — they are enabled to load both their operating system and their network data over the network (*diskless clients*), which could be an interesting option for both cost and security reasons.

21.12 Time Synchronization with xntp

The exact time plays an important role in many processes in a computer system. For this purpose, all computers usually have a built-in clock. Unfortunately, these clocks often do not meet the requirements of applications like databases. Therefore, the local clock must regularly be corrected manually or over a network. In the best case, the computer clock should never be set back, and the amount by which it is set forward should not exceed


TSIG Key Management

Define TSIG keys used for dynamic zone updates here. To add a new TSIG key, use the **File Name** text field or the **Browse** button then click **Add**. To delete an existing TSIG key, select it in the list and click **Delete**.

TSIG Key Management

File Name: /etc

Figure 21.56: DHCP Server: TSIG Configuration

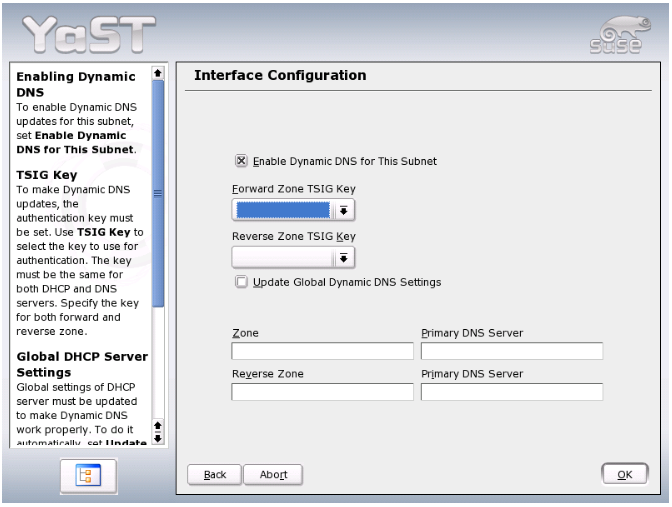
certain time intervals. The computer clock can easily be corrected with ntpdate from time to time. However, this causes a sudden time difference that may not be tolerated by all applications.

xntp provides an interesting approach for solving this problems. Firstly, xntp regularly corrects the local computer clock on the basis of collected correction data. Secondly, it continuously corrects the local time with the help of time servers in the network. Thirdly, it enables the management of local reference clocks, such as radio-controlled clocks.

21.12.1 Configuration in the Network

xntp is preset to use the local computer clock as time reference. The easiest way to use a time server in the network is the specification of "server" parameters. For example, if a time server called `ntp.example.com` is available in the network, this server can be added to the file `/etc/ntp.conf` in the form `server ntp.example.com`.

To add further time servers, insert additional lines with the keyword "server". After initializing xntpd with the command `rcxntpd start`, it takes one hour until the time is stabilized and the "drift" file for correcting the local computer clock is created. In the long run, the advantage of the



Enabling Dynamic DNS
To enable Dynamic DNS updates for this subnet, set **Enable Dynamic DNS for This Subnet**.

TSIG Key
To make Dynamic DNS updates, the authentication key must be set. Use **TSIG Key** to select the key to use for authentication. The key must be the same for both DHCP and DNS servers. Specify the key for both forward and reverse zone.

Global DHCP Server Settings
Global settings of DHCP server must be updated to make Dynamic DNS work properly. To do it automatically, set **Update**

Interface Configuration

☒ Enable Dynamic DNS for This Subnet

Forward Zone TSIG Key

Reverse Zone TSIG Key

☐ Update Global Dynamic DNS Settings

Zone | Primary DNS Server

Reverse Zone | Primary DNS Server

Figure 21.57: *DHCP Server: Interface Configuration for Dynamic DNS*

"drift" file is that the drift of the hardware clock can be projected as soon as the computer is powered on. The correction is activated immediately, resulting in a high stability of the computer time.

If the time server in your network can be reached via broadcast, you do not need the server name. In this case, enter the command `broadcastclient` in the configuration file `/etc/ntp.conf`. To avoid an incorrect time server in the network from changing the computer time, set up the authentication mechanisms.

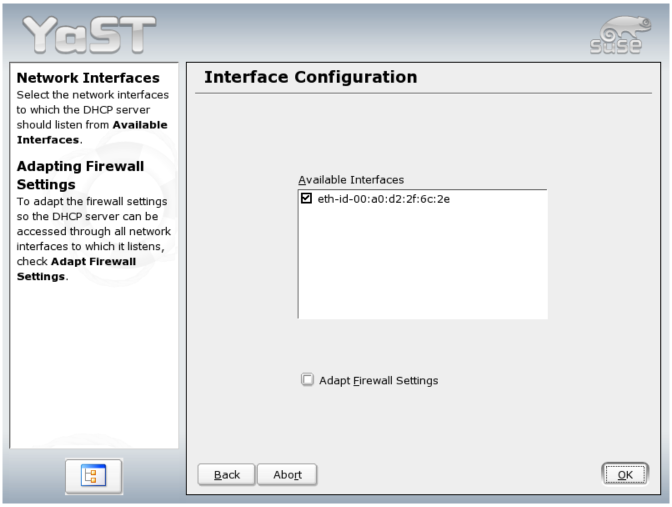
Normally, every xntpd in the network can also be addressed as time server. If you want to run xntpd with broadcasts, configure the `broadcast` option:

```
broadcast 192.168.0.255
```

Adjust the broadcast address to your circumstances. Make sure the time server uses the correct time. This can be done with reference clocks.

21.12.2 Setting up a Local Reference Clock

The software package xntp also contains drivers for connecting local reference clocks. A list of supported clocks is available in the `xntp-doc` pack-



Network Interfaces
Select the network interfaces to which the DHCP server should listen from **Available Interfaces**.

Adapting Firewall Settings
To adapt the firewall settings so the DHCP server can be accessed through all network interfaces to which it listens, check **Adapt Firewall Settings**.

Available Interfaces
☑ eth-id-00:a0:d2:2f:6c:2e

☐ Adapt Firewall Settings

Figure 21.58: *DHCP Server: Network Interface and Firewall*

age in the file `file:/usr/share/doc/packages/xntp-doc/html/refclock.htm`. Every driver is associated with a number. In xntp, the actual configuration takes place by means of pseudo IPs. 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*⟩. Refer to the above-mentioned file containing the list of reference clocks to get the value for ⟨*t*⟩. ⟨*u*⟩ is the device number that is only higher than `0` if you use several clocks of the same type on the computer. For example, a “Type 8 Generic Reference Driver (PARSE)” has the pseudo IP address `127.127.8.0`.

Normally, the individual drivers have special parameters that describe configuration details. The file `file:/usr/share/doc/packages/xntp-doc/html/refclock.htm` provides links to the respective driver pages describing these parameters. For example, the “Type 8” clock requires an additional mode that specifies the clock more precisely. For example, the “Conrad DCF77 receiver module” has “mode 5”. To make xntp accept this clock as a reference, specify the keyword `prefer`. Thus, the complete `server` line for a “Conrad DCF77 receiver module” would be as follows:

```
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/html`.

22

The Apache Web Server

With a share of more than sixty percent, Apache is the world's most widely-used web server (source: `http://www.netcraft.com`). For web applications, Apache is often combined with Linux, the database MySQL, and the programming languages PHP and Perl. This combination is commonly referred to as *LAMP*.

22.1 Basics 542
22.2 Setting up the HTTP Server with YaST 543
22.3 Apache Modules 544
22.4 New Features of Apache 2 545
22.5 Threads 546
22.6 Installation 546
22.7 Configuration 548
22.8 Using Apache 553
22.9 Active Contents 553
22.10 Virtual Hosts 560
22.11 Security 563
22.12 Troubleshooting 564
22.13 For More Information 564

22.1 Basics

22.1.1 Web Server

A web server issues `HTML` pages requested by a client. These pages can be stored in a directory (passive or static pages) or generated in response to a query (active contents).

22.1.2 HTTP

The clients are usually web browsers, like Konqueror or Mozilla. Communication between the browser and the web server takes place by way of the *H*yper*T*ext *T*ransfer *P*rotocol (HTTP). The current version, HTTP 1.1, is documented in RFC 2068 and in the update RFC 2616. These RFCs are available at `http://www.w3.org`.

22.1.3 URLs

Clients use URLs, such as `http://www.suse.com/index_us.html`, to request pages from the server. A URL consists of:

- A protocol. Frequently-used protocols:
  - ▷ `http://` HTTP protocol
  - ▷ `https://` Secure, encrypted version of HTTP
  - ▷ `ftp://` file transfer protocol for uploading and downloading files
- A domain, in this example, `www.suse.com`. The domain can be subdivided in two parts. The first part (`www`) points to a computer. The second part (`suse.com`) is the actual domain. Together, they are referred to as FQDN (fully qualified domain name).
- A resource, in this example, `index_us.html`. This part specifies the full path to the resource. The resource can be a file, as in this example. However, it can also be a CGI script, a Java server page, or some other resource.

The responsible Internet mechanism (such as the domain name system, DNS) conveys the query to the domain, directing it to one or several computers hosting the resource. Apache then delivers the actual resource (in this example, the page `index_us.html`) from its file directory. In this case, the file is located in the top level of the directory. However, resources can also be located in subdirectories, as in `http://www.suse.com/us/business/services/support/index.html`.

The file path is relative to the `DocumentRoot`, which can be changed in the configuration file. Section 22.7.2 on page 549 describes how this is done.

22.1.4 Automatic Display of a Default Page

If no default page is specified, Apache automatically appends one of the common names to the URL. The most frequently-used name for such pages is `index.html`. This function, together with the actual page names to be used by the server, can be configured as described in Section 22.7.2 on page 550. In our example, `http://www.suse.com` is sufficient to prompt the server to deliver the page `http://www.suse.com/index_us.html`.

22.2 Setting up the HTTP Server with YaST

Apache 2 can easily be set up with the help of YaST, but some knowledge about the subject is needed to set up a web server this way. After selecting 'Network Services' → 'HTTP Server' in the YaST control center, you may be prompted for the installation of some packages that are still missing. As soon as everything is installed, YaST displays the configuration dialog.

In this dialog, first enable the HTTP service itself, which requires setting the following three options: 'Server Name', 'Server Administrator E-Mail', and 'Listen on'. The last option is already set to the default of port 80. You can then use 'Add' to set further options. Selecting 'Edit' enables changing the value of the highlighted option. Selecting 'Delete' allows removing it completely.

With 'Advanced', view the logs ('Show Access Log' and 'Show Error Log') or specify the 'Server Modules' to load. The latter opens a dialog in which to enable or disable modules by selecting 'Toggle Status' and add modules by selecting 'Add Module'.

22.3 Apache Modules

By means of modules, Apache can be expanded with a wide range of functions. For example, Apache can execute CGI scripts in diverse programming languages by means of modules. Apart from Perl and PHP, additional scripting languages, such as Python or Ruby, are also available. Furthermore, there are modules for secure data transmission (Secure Sockets Layer, SSL), user authentication, expanded logging, and other functions.

By means of custom modules, Apache can be adapted to all kinds of requirements and preferences. This requires a certain amount of know-how. For further information, refer to Section 22.13.4 on page 565.

When Apache processes a query, several "handlers" can be specified for handling the query (by means of directives in the configuration file). These handlers can be part of Apache or a module invoked for processing the query. Thus, this procedure can be arranged in a very flexible way. It is also possible to use your own custom modules with Apache to influence the way in which requests are processed.

The modularization in Apache 2 has reached an advanced level, where almost everything except some minor tasks is handled by means of modules. In Apache 2, even HTTP is processed by way of modules. Accordingly, Apache 2 does not necessarily need to be a web server. It can also be used for completely different purposes with other modules. For example, there is a proof-of-concept mail server (POP3) module based on Apache.

Apache supports a number of useful features, some of which are described below.

Virtual Hosts Support for virtual hosts means that a single instance of Apache and a single machine can be used for several web sites. For the users, the web server appears as several independent web servers. The virtual hosts can be configured on different IP addresses or on the basis of names. Thus, you can save the acquisition costs and administration workload for additional machines.

Flexible URL Rewriting Apache offers a number of possibilities for manipulating and rewriting URLs. Check the Apache documentation for details.

Content Negotiation Apache can deliver a page that is adapted to the capabilities of the client (browser). For example, simple versions without frames can be delivered for older browsers or browsers that only operate in text mode, such as Lynx. In this way, the notorious

JavaScript incompatibility of various browsers can be circumvented by delivering a special page version for every browser (provided you are prepared to adapt the JavaScript code for each individual browser).

Flexible Error Handling You can react in a flexible way and provide a suitable response in the event of an error, such as nonexistent pages. The response can even be generated actively, for example, with CGI.

22.4 New Features of Apache 2

The following is a list of the main new features of Apache 2. For detailed information about version 2.0 of the Apache HTTP server, refer to `http://httpd.apache.org/docs-2.0/en/`.

- Multiple queries may be processed as threads or processes. The process management has been relocated to a separate module, called the multiprocessing module (MPM). Depending on the MPM, Apache 2 responds to queries in different ways, with different effects on the performance and the use of modules. Details are provided below.
- The innards of Apache have been thoroughly revised. Apache now uses a new, special base library (Apache Portable Runtime, APR) as the interface to system functions and for the memory management. Moreover, important and widespread modules, such as mod_gzip (successor: mod_deflate) and mod_ssl, which have a profound impact on the processing of requests, are now integrated more fully in Apache.
- Apache 2 supports the Internet protocol IPv6.
- A new mechanism now enables manufacturers of modules to specify the desired loading sequence for modules. Thus, users are no longer required to do this themselves. The order in which modules are executed is often significant. Previously, it was determined by means of the loading sequence. For example, a module that only gives authenticated users access to certain resources must be run first to prevent the pages from being displayed to users who do not have any access permissions.
- Queries to and replies from Apache can be processed with filters.

- Support for files that are larger than 2 GB (large file support, LFS) on 32-bit systems.
- Some of the newer modules are only available for Apache 2.
- Multilanguage error responses.

22.5 Threads

A thread is a "lighter" form of a process. The advantage of a thread over a process is its lower resource consumption. Therefore, the use of threads instead of processes increases the performance. The disadvantage is that applications executed in a thread environment must be thread-safe. This means that:

- Functions (or the methods in object-oriented applications) must be reentrant — a function with the same input always returns the same result, even if other threads concurrently execute the same function. Accordingly, functions must be programmed in such a way that they can be executed simultaneously by severals threads.
- The access to resources (usually variables) must be arranged in such a way that concurrent threads do not conflict.

Apache 2 handles queries as separate processes or in a mixed mode combining processes and threads. The MPM *prefork* is responsible for the execution as process. The MPM *worker* prompts the execution as thread. Select the MPM to use during the installation (see Section 22.6). The third mode — *perchild* — is not yet fully mature and is therefore not available for installation in SUSE LINUX.

22.6 Installation

22.6.1 Package Selection in YaST

For a basic installation, it is sufficient to select the Apache package `apache2`. Additionally, you may install one of the MPM (multiprocessing module) packages, such as `apache2-prefork` or `apache2-worker`. When choosing an MPM, remember that the thread-based worker MPM cannot be used with `mod_php4`, as some of the libraries of `mod_php4` are not yet thread-safe.

22.6.2 Activating Apache

After installation, Apache will not be started automatically. To start Apache, activate it in the runlevel editor. To start it permanently when the system is booted, check runlevels 3 and 5 in the runlevel editor. To test whether Apache is running, go to `http://localhost/` in a browser. If Apache is active, you will see an example page, provided `apache2-example-pages` is installed.

22.6.3 Modules for Active Contents

To use active contents with the help of modules, install the modules for the respective programming languages. These are `apache2-mod_perl` for Perl, `mod_php4` for PHP, and `mod_python` for Python. The use of these modules is covered in Section 22.9.5 on page 556.

22.6.4 Other Recommended Packages

Additionally, you should install the extensive documentation provided in `apache2-doc`. An alias (Section 22.7 on the next page explains what an alias is) is available for the documentation, enabling you to access it with the URL `http://localhost/manual` following the installation.

To develop modules for Apache or compile third-party modules, install `apache2-devel` and the needed development tools. These include the `apxs` tools, which are described in Section 22.6.5.

22.6.5 Installation of Modules with apxs

`apxs2` is an important tool for module developers. This program enables the compilation and installation of modules from source code with a single command (including the required changes to the configuration files). Furthermore, you can also install modules available as object files (extension `.o`) or static libraries (extension `.a`). When installing from sources, `apxs2` creates a dynamic shared object (DSO), which is directly used by Apache as a module.

The installation of a module from source code can be performed with a command like `apxs2 -c -i -a mod_foo.c`. Other options of `apxs2` are described in its man page.

`apxs2` is available in several versions: `apxs2`, `apxs2-prefork`, and `apxs2-worker`. `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 (prefork or worker). `apxs2` installs modules in `/usr/lib/apache2/` and `apxs2-prefork` installs modules in `/usr/lib/apache2-prefork/`.

The option `-a` should not be used with Apache 2, as this would cause the changes to be written directly to `/etc/apache2/httpd.conf`. Rather, modules should be activated by means of the entry `APACHE_MODULES` in `/etc/sysconfig/apache2` as described in Section 22.7.1.

22.7 Configuration

Following the installation of Apache, additional changes are only necessary if you have special needs or preferences. Apache can be configured either with SuSEconfig or by directly editing the file `/etc/apache2/httpd.conf`.

22.7.1 Configuration with SuSEconfig

The settings made in `/etc/sysconfig/apache2` are applied to the Apache configuration files by SuSEconfig. The offered configuration options should be sufficient for most scenarios. Each variable found in the file is provided with a comment to explain its effect.

Custom Configuration Files

Instead of performing changes directly in the configuration file `/etc/apache2/httpd.conf`, you can designate your own configuration file (such as `httpd.conf.local`) with the help of the variable `APACHE_CONF_INCLUDE_FILES`. Consequently, the file will be interpreted by the main configuration file. In this way, changes to the configuration will be retained even if the file `/etc/apache2/httpd.conf` is overwritten during a new installation.

Modules

Modules installed with YaST can be activated by including the name of the module in the list specified under the variable `APACHE_MODULES`. This variable is defined in the file `/etc/sysconfig/apache2`.

Flags

`APACHE_SERVER_FLAGS` can be used to specify flags that activate or deactivate certain sections of the configuration file. If a section in the configuration file is enclosed in

```
<IfDefine someflag>
.
.
.
</IfDefine>
```

it will only be activated if the respective flag is set in `ACTIVE_SERVER_FLAGS`: `ACTIVE_SERVER_FLAGS = ... someflag ...` . In this way, extensive sections of the configuration file can easily be activated or deactivated for test purposes.

22.7.2 Manual Configuration

You can edit the configuration file `/etc/apache2/httpd.conf` to enable features that are not available through the settings defined in `/etc/sysconfig/apache2`. The following sections describe some of the parameters that can be set. They are listed below in the order in which they appear in the file.

DocumentRoot

One basic setting is the `DocumentRoot` — the directory under which Apache expects web pages the server should deliver. For the default virtual host, it is set to `/srv/www/htdocs`. Normally, this setting does not need to be changed.

Timeout

Specifies the waiting period after which the server reports a time-out for a request.

MaxClients

The maximum number of clients Apache can handle concurrently. The default setting is `150`, but this value may be too small for a heavily frequented web site.

LoadModule

The `LoadModule` directives specify the modules to load. In the case of Apache 2, the loading sequence is determined by the modules themselves (see Section 22.4 on page 545). These directives also specify the file containing the module.

Port

Specifies the port on which Apache listens for queries. Usually, this is port 80, the default port for HTTP. Normally, this setting should not be changed. One reason for letting Apache listen to another port may be the test of a new version of a web site. In this way, the operational version of the web site continues to be accessible via default port 80.

Another reason may be that you merely want to make pages available on the intranet, as they contain information that is not intended for the public. For this purpose, set the port to a value like `8080` and block external access to this port by means of the firewall. In this way, the server can be protected against external access.

Directory

This directive can be used to set the access permissions and other permissions for a directory. A directive of this kind also exists for the `DocumentRoot`. The directory name specified here must be changed whenever the `DocumentRoot` is changed.

DirectoryIndex

Here, determine for which files Apache should search to complete a URL lacking a file specification. The default setting is `index.html`. For example, if the client requests the URL `http://www.xyz.com/foo/bar` and the directory `foo/bar` containing a file called `index.html` exists under the `DocumentRoot`, Apache returns this page to the client.

AllowOverride

Every directory from which Apache delivers documents may contain a file that can override the global access permissions and other settings for this directory. These settings are applied recursively to the current directory and its subdirectories until they are overridden by another such file in a subdirectory. Accordingly, settings specified in such a file are applied globally if it is located in the `DocumentRoot`. Such files normally have the name `.htaccess`, but this can be changed as described in Section 22.7.2.

Use `AllowOverride` to determine if the settings specified in local files may override the global settings. Possible values are `None`, `All`, and any combination of `Options`, `FileInfo`, `AuthConfig`, and `Limit`. The meanings of these values are described in detail in the Apache documentation. The (safe) default setting is `None`.

Order

This option determines the order in which the settings for `Allow` and `Deny` access permissions are applied. The default setting is:

```
Order allow,deny
```

Accordingly, the access permissions for allowed accesses are applied first, followed by the access permissions for denied accesses. The underlying approach is based on one of the following:

allow all allow every access and define exceptions

deny all deny every access and define exceptions

Example for `deny all`:

```
Order deny,allow
Deny from all
Allow from example.com
Allow from 10.1.0.0/255.255.0.0
```

AccessFileName

Here, set the name for the files that can override the global access permissions and other settings for directories delivered by Apache (see Section 22.7.2 on the facing page). The default setting is `.htaccess`.

ErrorLog

Specifies the name of the file in which Apache logs error messages. The default setting is `/var/log/httpd/errorlog`. Error messages for virtual hosts (see Section 22.10 on page 560) are also logged in this file, unless a special log file was specified in the `VirtualHost` section of the configuration file.

LogLevel

Error messages are classified according to various severity levels. This setting specifies the severity level from which error messages are logged. Setting it to a level causes error messages of this and higher severity levels to be logged. The default setting is `warn`.

Alias

Using an alias, specify a shortcut for a directory that enables direct access to this directory. For example, the alias `/manual/` enables access to the directory `/srv/www/htdocs/manual` even if the `DocumentRoot` is set to a directory other than `/srv/www/htdocs` (the alias will make no difference at all if the `DocumentRoot` is set to that directory). With this alias, `http://localhost/manual` enables direct access to the respective directory. To define the permissions for the new target directory as specified with an `Alias` directive, you may want to specify a `Directory` directive for it (see Section 22.7.2 on page 550)

ScriptAlias

This directive is similar to `Alias`. In addition, it indicates that the files in the target directory should be treated as CGI scripts.

Server-Side Includes

Server-side includes can be activated by searching all executable files for SSIs. This can be done with the following instruction:

```
<IfModule mod_include.c>
XBitHack on </IfModule>
```

To search a file for SSIs, use the command `chmod +x` ⟨*`filename`*⟩ to make the file executable. Alternatively, explicitly specify the file type to search for SSIs. This can be done with the following instruction:

```
AddType text/html .shtml
AddHandler server-parsed .shtml
```

It is *not* advisable to simply state `.html`, as this will cause Apache to search all pages for SSIs (even those that definitely do not contain any), which greatly impedes the performance. In SUSE LINUX, these two directives are already included in the configuration files, so normally no changes are necessary.

UserDir

With the help of the module `mod_userdir` and the directive `UserDir`, specify a directory in a user's home directory from which files may be published through Apache. This can be configured in SuSEconfig by setting the variable `HTTPD_SEC_PUBLIC_HTML` accordingly. To enable the publishing of files, the variable must be set to `yes`. This results in the following entry in the file `/etc/httpd/suse_public_html.conf` (which is interpreted by `/etc/apache2/httpd.conf`).

```
<IfModule mod_userdir.c>
UserDir public_html
</IfModule>
```

22.8 Using Apache

To display static web pages with Apache, simply place your files in the correct directory. In SUSE LINUX, the correct directory is `/srv/www/htdocs`. A few small example pages may already be installed there. Use these pages to check if Apache was installed correctly and is currently active. Subsequently, you can simply overwrite or uninstall these pages. Custom CGI scripts are installed in `/srv/www/cgi-bin`.

During operation, Apache writes log messages to the file `/var/log/httpd/access_log` or `/var/log/apache2/access_log`. These messages show which resources were requested and delivered at what time and with which method (`GET`, `POST`, etc.). Error messages are logged to `/var/log/apache2`.

22.9 Active Contents

Apache provides several possibilities for the delivery of active contents. Active contents are HTML pages that are generated on the basis of variable input data from the client, such as search engines that respond to the input of one or several search strings (possibly interlinked with logical operators like AND or OR) by returning a list of pages containing these search strings.

Apache offers three ways of generating active contents:

Server Side Includes (SSI) These are directives that are embedded in an HTML page by means of special comments. Apache interprets the content of the comments and delivers the result as part of the HTML page.

Common Gateway Interface (CGI)
These are programs that are located in certain directories. Apache forwards the parameters transmitted by the client to these programs and returns the output of the programs. This kind of programming is quite easy, especially since existing command-line programs can be designed in such a way that they accept input from Apache and return their output to Apache.

Module Apache offers interfaces for executing any modules within the scope of request processing. Apache gives these programs access to important information, such as the request or the HTTP headers. Programs can take part in the generation of active contents as well as in other functions (such as authentication). The programming of such modules requires some expertise. The advantages of this approach are high performance and possibilities that exceed those of SSI and CGI.

While CGI scripts are executed directly by Apache (under the user ID of their owner), modules are controlled by a persistent interpreter that is embedded in Apache. In this way, separate processes do not need to be started and terminated for every request (this would result in a considerable overhead for the process management, memory management, etc.). Rather, the script is handled by the interpreter running under the ID of the web server.

However, this approach has a catch. Compared to modules, CGI scripts are relatively tolerant of careless programming. With CGI scripts, errors, such as a failure to release resources and memory, do not have a lasting effect, because the programs are terminated after the request has been processed. This results in the clearance of memory that was not released by the program due to a programming error. With modules, the effects of programming errors accumulate, as the interpreter is persistent. If the server is not restarted and the interpreter runs for several months, the failure to release resources, such as database connections, can be quite disturbing.

22.9.1 Server Side Includes: SSI

Server-side includes are directives that are embedded in special comments and executed by Apache. The result is embedded in the output. For example, the current date can be printed with `<!--#echo var="DATE_LOCAL" -->`. The # at the end of the opening comment mark `<!&#8211;&#8211;` shows Apache that this is an SSI directive and not a simple comment.

SSIs can be activated in several ways. The easiest approach is to search all executable files for SSIs. Another approach is to specify certain file types to search for SSIs. Both settings are explained in Section 22.7.2 on page 552.

22.9.2 Common Gateway Interface: CGI

CGI is the abbreviation for *Common Gateway Interface*. With CGI, the server does not simply deliver a static HTML page, but executes a program that generates the page. This enables the generation of pages representing the result of a calculation, such as the result of the search in a database. By means of arguments passed to the executed program, the program can return an individual response page for every request.

The main advantage of CGI is that this technology is quite simple. The program merely must exist in a specific directory to be executed by the web server just like a command-line program. The server sends the program output on the standard output channel (`stdout`) to the client.

22.9.3 GET and POST

Input parameters can be passed to the server with `GET` or `POST`. Depending on which method is used, the server passes the parameters to the script in various ways. With `POST`, the server passes the parameters to the program on the standard input channel (`stdin`). The program would receive its input in the same way when started from a console.

With `GET`, the server uses the environment variable `QUERY_STRING` to pass the parameters to the program. An environment variable is a variable made available globally by the system (such as the variable `PATH`, which contains a list of paths the system searches for executable commands when the user enters a command).

22.9.4 Languages for CGI

Theoretically, CGI programs can be written in any programming language. Usually, scripting languages (interpreted languages), such as Perl or PHP, are used for this purpose. If speed is critical, C or C++ may be more suitable.

In the simplest case, Apache looks for these programs in a specific directory (`cgi-bin`). This directory can be set in the configuration file, described in Section 22.7 on page 548).

If necessary, additional directories can be specified. In this case, Apache searches these directories for executable programs. However, this represents a security risk, as any user will be able to let Apache execute programs (some of which may be malicious). If executable programs are restricted to `cgi-bin`, the administrator can easily see who places which scripts and programs in this directory and check them for any malicious intent.

22.9.5 Generating Active Contents with Modules

A variety of modules is available for use with Apache. The term "module" is used in two different senses. First, there are modules that can be integrated in Apache for the purpose of handling specific functions, such as modules for embedding programming languages. These modules are introduced below.

Second, in connection with programming languages, modules refer to an independent group of functions, classes, and variables. These modules are integrated in a program to provide a certain functionality, such as the CGI modules available for all scripting languages. These modules facilitate the programming of CGI applications by providing various functions, such as methods for reading the request parameters and for the HTML output.

22.9.6 mod_perl

Perl is a popular, proven scripting language. There are numerous modules and libraries for Perl, including a library for expanding the Apache configuration file. The home page for Perl is `http://www.perl.com/`. A range of libraries for Perl is available in the Comprehensive Perl Archive Network (CPAN) at `http://www.cpan.org/`.

Setting up mod_perl

To set up `mod_perl` in SUSE LINUX, simply install the respective package (see Section 22.6 on page 546). Following the installation, the Apache configuration file will include the necessary entries (see `/etc/apache2/mod_perl-startup.pl`). Information about `mod_perl` is available at `http://perl.apache.org/`.

mod_perl versus CGI

In the simplest case, run a previous CGI script as a `mod_perl` script by requesting it with a different URL. The configuration file contains aliases that point to the same directory and execute any scripts it contains either via CGI or via `mod_perl`. All these entries already exist in the configuration file. The alias entry for CGI is as follows:

```
ScriptAlias /cgi-bin/ "/srv/www/cgi-bin/"
```

The entries for `mod_perl` are as follows:

```
<IfModule mod_perl.c>
# Provide two aliases to the same cgi-bin directory,
# to see the effects of the 2 different mod_perl modes.
# for Apache::Registry Mode
ScriptAlias /perl/          "/srv/www/cgi-bin/"
# for Apache::Perlrun Mode
ScriptAlias /cgi-perl/      "/srv/www/cgi-bin/"
</IfModule>
```

The following entries are also needed for `mod_perl`. These entries already exist in the configuration file.

```
#
# If mod_perl is activated, load configuration information
#
<IfModule mod_perl.c>
Perlrequire /usr/include/apache/modules/perl/startup.perl
PerlModule Apache::Registry

#
# set Apache::Registry Mode for /perl Alias
#
<Location /perl>
SetHandler  perl-script
PerlHandler Apache::Registry
Options ExecCGI
PerlSendHeader On
```

```
</Location>

#
# set Apache::PerlRun Mode for /cgi-perl Alias
#
<Location /cgi-perl>
SetHandler  perl-script
PerlHandler Apache::PerlRun
Options ExecCGI
PerlSendHeader On
</Location>

</IfModule>
```

These entries create aliases for the `Apache::Registry` and `Apache::PerlRun` modes. The difference between these two modes is as follows:

Apache::Registry All scripts are compiled and kept in a cache. Every script is applied as the content of a subroutine. Although this is good for performance, there is a disadvantage: the scripts must be programmed extremely carefully, as the variables and subroutines persist between the requests. This means that you must reset the variables to enable their use for the next request. If, for example, the credit card number of a customer is stored in a variable in an online banking script, this number could appear again when the next customer uses the application and requests the same script.

Apache::PerlRun The scripts are recompiled for every request. Variables and subroutines disappear from the namespace between the requests (the namespace is the entirety of all variable names and routine names that are defined at a given time during the existence of a script). Therefore, `Apache::PerlRun` does not necessitate painstaking programming, as all variables are reinitialized when the script is started and no values are kept from previous requests. For this reason, `Apache::PerlRun` is slower than `Apache::Registry` but still a lot faster than CGI (in spite of some similarities to CGI), because no separate process is started for the interpreter.

22.9.7 mod_php4

PHP is a programming language that was especially developed for use with web servers. In contrast to other languages whose commands are stored in separate files (scripts), the PHP commands are embedded in an HTML page (similar to SSI). The PHP interpreter processes the PHP commands and embeds the processing result in the HTML page.

The home page for PHP is `http://www.php.net/`. For PHP to work, install `mod_php4-core` and, in addition, `apache2-mod_php4` for Apache 2.

22.9.8 mod_python

Python is an object-oriented programming language with a very clear and legible syntax. An unusual but convenient feature is that the program structure depends on the indentation. Blocks are not defined with braces (as in C and Perl) or other demarcation elements (such as `begin` and `end`), but by their level of indentation. The package to install is `apache2-mod_python`.

More information about this language is available at `http://www.python.org/`. For more information about `mod_python`, visit the URL `http://www.modpython.org/`.

22.9.9 mod_ruby

Ruby is a relatively new, object-oriented high-level programming language that resembles certain aspects of Perl and Python and is ideal for scripts. Like Python, it has a clean, transparent syntax. On the other hand, Python has adopted abbreviations, such as `$.r` for the number of the last line read in the input file — a feature that is welcomed by some programmers and abhorred by others. The basic concept of Ruby closely resembles Smalltalk.

The home page of Ruby is `http://www.ruby-lang.org/`. An Apache module is available for Ruby. The home page is `http://www.modruby.net/`.

22.10 Virtual Hosts

Using virtual hosts, host several domains with a single web server. In this way, save the costs and administration workload for separate servers for each domain. One of the first web servers that offered this feature, Apache offers several possibilities for virtual hosts:

- Name-based virtual hosts
- IP-based virtual hosts
- Operation of multiple instances of Apache on one machine

22.10.1 Name-Based Virtual Hosts

With name-based virtual hosts, one instance of Apache hosts several domains. You do not need to set up multiple IPs for a machine. This is the easiest, preferred alternative. Reasons against the use of name-based virtual hosts are covered in the Apache documentation.

Configure it directly by way of the configuration file (`/etc/apache2/httpd.conf`). To activate name-based virtual hosts, a suitable directive must be specified: `NameVirtualHost *.*` is sufficient to prompt Apache to accept all incoming requests. Subsequently, the individual hosts must be configured:

```
<VirtualHost *>
    ServerName www.mycompany.com
    DocumentRoot /srv/www/htdocs/mycompany.com
    ServerAdmin webmaster@mycompany.com
    ErrorLog /var/log/httpd/www.my.company.com-error_log
    CustomLog /var/log/httpd/www.mycompany.com-access_log common
</VirtualHost>


<VirtualHost *>
    ServerName www.myothercompany.com
    DocumentRoot /srv/www/htdocs/myothercompany.com
    ServerAdmin webmaster@myothercompany.com
    ErrorLog /var/log/httpd/www.myothercompany.com-error_log
    CustomLog /var/log/httpd/www.myothercompany.com-access_log common
</VirtualHost>
```

In the case of Apache 2, however, the paths of log files as shown in the above example (and in any examples further below) should be changed from `/var/log/httpd` to `/var/log/apache2`. A `VirtualHost` entry also must be configured for the domain originally hosted on the server (`www.mycompany.com`). So in this example, the original domain and one additional domain (`www.myothercompany.com`) are hosted on the same server.

Just as in `NameVirtualHost`, a `*` is used in the `VirtualHost` directives. Apache uses the host field in the HTTP header to connect the request with the virtual host. The request is forwarded to the virtual host whose `ServerName` matches the host name specified in this field.

For the directives `ErrorLog` and `CustomLog`, the log files do not need to contain the domain name. Here, use a name of your choice.

`ServerAdmin` designates the e-mail address of the responsible person that can be contacted if problems arise. In the event of errors, Apache gives this address in the error messages it sends to the client.

22.10.2 IP-Based Virtual Hosts

This alternative requires the setup of multiple IPs for a machine. In this case, one instance of Apache hosts several domains, each of which is assigned a different IP. The following example shows how Apache can be configured to host the original IP (`192.168.1.10`) plus two additional domains on additional IPs (`192.168.1.20` and `192.168.1.21`). This particular example only works on an intranet, as IPs ranging from `192.168.0.0` to `192.168.255.0` are not routed on the Internet.

Configuring IP Aliasing

For Apache to host multiple IPs, the underlying machine must accept requests for multiple IPs. This is called multi-IP hosting. For this purpose, IP aliasing must be activated in the kernel. This is the default setting in SUSE LINUX.

Once the kernel has been configured for IP aliasing, the commands `ifconfig` and `route` can be used to set up additional IPs on the host. These commands must be executed as `root`. For the following example, it is assumed that the host already has its own IP (such as `192.168.1.10`), which is assigned to the network device `eth0`.

Enter the command `ifconfig` to find out the IP of the host. Further IPs can be added with commands such as the following:

```
/sbin/ifconfig eth0:0 192.168.1.20
/sbin/ifconfig eth0:1 192.168.1.21
```

All these IPs will be assigned to the same physical network device (`eth0`).

Virtual Hosts with IPs

Once IP aliasing has been set up on the system or the host has been configured with several network cards, Apache can be configured. Specify a separate `VirtualHost` block for every virtual server:

```
<VirtualHost 192.168.1.20>
    ServerName www.myothercompany.com
    DocumentRoot /srv/www/htdocs/myothercompany.com
    ServerAdmin webmaster@myothercompany.com
    ErrorLog /var/log/httpd/www.myothercompany.com-error_log
    CustomLog /var/log/httpd/www.myothercompany.com-access_log common
</VirtualHost>

<VirtualHost 192.168.1.21>
    ServerName www.anothercompany.com
    DocumentRoot /srv/www/htdocs/anothercompany.com
    ServerAdmin webmaster@anothercompany.com
    ErrorLog /var/log/httpd/www.anothercompany.com-error_log
    CustomLog /var/log/httpd/www.anothercompany.com-access_log common
</VirtualHost>
```

`VirtualHost` directives are only specified for the additional domains. The original domain (`www.mycompany.com`) is configured through its own settings (under `DocumentRoot`, etc.) outside the `VirtualHost` blocks.

22.10.3 Multiple Instances of Apache

With the above methods for providing virtual hosts, administrators of one domain can read the data of other domains. To segregate the individual domains, start several instances of Apache, each with its own settings for `User`, `Group`, and other directives in the configuration file.

In the configuration file, use the `Listen` directive to specify the IP handled by the respective Apache instance. For the above example, the directive for the first Apache instance would be as follows:

```
Listen 192.168.1.10:80
```

For the other two instances:

```
Listen 192.168.1.20:80
Listen 192.168.1.21:80
```

22.11 Security

22.11.1 Minimizing the Risk

If you do not need a web server on a machine, deactivate Apache in the runlevel editor, uninstall it, or refrain from installing it in the first place. To minimize the risk, deactivate all unneeded servers. This especially applies to hosts used as firewalls. If possible, do not run any servers on these hosts.

22.11.2 Access Permissions

DocumentRoot Should Belong to root

By default, the `DocumentRoot` directory (`/srv/www/htdocs`) and the CGI directory belong to the user `root`. You should not change this setting. 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 user `wwwrun`. Also, Apache should not have any write permissions for the data and scripts it delivers. Therefore, these should not belong to the user `wwwrun`, but to another user (such as `root`).

To enable users to place files in the document directory of Apache, do not make it writable for all. Instead, create a subdirectory that is writable for all (such as `/srv/www/htdocs/miscellaneous`).

Publishing Documents from Home Directories

Another possibility to make sure that users can publish their files in the network is to specify a subdirectory in users' home directories in the configuration file. Users can then place any files for web presentations in this directory (for instance, `~/public_html`). By default, this is activated in SUSE LINUX. See Section 22.7.2 on page 553 for details.

These web pages can be accessed by specifying the user in the URL. The URL contains the element `~username` as a shortcut for the respective directory in the user's home directory. For example, enter `http://localhost/~tux` in a browser to list the files in the directory `public_html` in the home directory of the user tux.

22.11.3 Staying Updated

If you operate a web server and especially if this web server is publicly accessible, stay informed about bugs and potential vulnerable spots. Sources for exploits and fixes are listed in Section 22.13.3 on page 565.

22.12 Troubleshooting

If problems appear, for example, Apache does not display a page or does not display it correctly, the following procedures can help find the problems.

- First, take a look at the error log and check if the messages it contains reveal the error. The general error log is located in `/var/log/httpd/error_log` or `/var/log/apache2/error_log`.

  A proven approach is to track the log files in a console to see how the server reacts to an access. This can be done by entering `tail -f /var/log/apache2/*_log` in a `root` console.
- Check the online bug database at `http://bugs.apache.org/`.
- Read the relevant mailing lists and newsgroups. The mailing list for users is available at `http://httpd.apache.org/userslist.html`. Recommended newsgroups are `comp.infosystems.www.servers.unix` and related groups.
- If none of these possibilities provide any solution and you are sure that you have detected a bug in Apache, report it at `http://www.suse.de/feedback/`.

22.13 For More Information

22.13.1 Apache

Apache is shipped with detailed documentation. The installation of this documentation is described in Section 22.6 on page 546. Following the installation, access the documentation at `http://localhost/manual`. The latest documentation is available from the Apache home page at `http://httpd.apache.org`.

22.13.2 CGI

More information about CGI is available at the following pages:

- `http://apache.perl.org/`

- `http://perl.apache.org/`
- `http://www.modperl.com/`
- `http://www.modperlcookbook.org/`
- `http://www.fastcgi.com/`
- `http://www.boutell.com/cgic/`

22.13.3 Security

The latest patches for the SUSE packages are made available at `http://www.suse.com/us/security/`. Visit this URL at regular intervals. Here, you can also sign up for the SUSE mailing list for security announcements.

The Apache team promotes an open information policy with regard to bugs in Apache. The latest bug reports and possible vulnerable spots are published at `http://httpd.apache.org/security_report.html`.

If you detect a security bug (check the mentioned pages to make sure it has not already been discovered), report it to `security@suse.de` or to `security@apache.org`.

Other sources for information about security issues of Apache (and other Internet programs):

- `http://www.cert.org/`
- `http://www.vnunet.com/`
- `http://www.securityfocus.com/`

22.13.4 Additional Sources

If you experience difficulties, take a look at the SUSE Support Database at `http://sdb.suse.de/en/`. An online newspaper focusing on Apache is available at `http://www.apacheweek.com/`.

The history of Apache is provided at `http://httpd.apache.org/ABOUT_APACHE.html`. This page also explains why the server is actually called Apache.

Information about upgrading from version 1.3 to 2.0 is available at `http://httpd.apache.org/docs-2.0/en/upgrading.html`.

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 work with all computers and modify the files and subsequently have the latest version of the data available on all computers.

23.1 Available Data Synchronization Software 568
23.2 Determining Factors for Selecting a Program 570
23.3 Introduction to Unison 574
23.4 Introduction to CVS 575
23.5 Introduction to Subversion 578
23.6 An Introduction to rsync 581
23.7 Introduction to mailsync 583

23.1 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.

Caution

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.

Caution

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.

23.1.1 Unison

`Unison` is not a network file system. Rather, the files are simply saved and edited locally. The program `Unison` can be executed manually to synchronize files. When the synchronization is performed for the first time, a database is created on the two hosts, containing check sums, time stamps, and permissions of the selected files. The next time it is executed, `Unison` can recognize which files were changed and propose transmission from or to the other host. Usually all suggestions can be accepted.

23.1.2 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.

23.1.3 subversion

As opposed to the evolved CVS, subversion is a consistently designed project.

subversion was developed to supersede CVS and to alleviate its technical shortcomings.

subversion has been improved in many respects to its predecessor. Due to its history, CVS only maintains files and is oblivious of directories. Directories also have a version history in subversion and can be copied and renamed just like files. It is also possible to add metadata to every file and to every directory. This metadata can be fully maintained with versioning. As opposed to CVS, subversion supports transparent network access over dedicated protocols, like WebDAV.

subversion was, in large part, assembled using already existing application packages. This is why the web server apache and the extension WebDAV are always run in conjunction with subversion.

23.1.4 mailsync

Unlike the synchronization tools covered in the previous sections, mailsync only synchronizes e-mails between mailboxes. The procedure can be applied to local mailbox files as well as to mailboxes on an IMAP server.

Based on the message ID contained in the e-mail header, the individual messages are either synchronized or deleted. Synchronization is possible between individual mailboxes and between mailbox hierarchies.

23.1.5 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.

23.2 Determining Factors for Selecting a Program

23.2.1 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 subversion, CVS, and WebDAV.

The other possibility is to let all networked hosts synchronize their data among each other as peers. This is the concept followed by unison. rsync actually works in client mode, but any client can also act as a server.

23.2.2 Portability

subversion, CVS, and unison are also available for many other operating systems, including various Unix and Windows systems.

23.2.3 Interactive versus Automatic

In subversion, CVS, WebDAV, and Unison,

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.

23.2.4 Conflicts: Incidence and Solution

Conflicts only rarely occur in subversion or 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 unison or CVS can easily be resolved.

Unison reports conflicts, allowing the affected files to be excluded from the synchronization. However, changes cannot be merged as easily as in subversion or CVS.

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.

23.2.5 Selecting and Adding Files

In its standard configuration, Unison synchronizes an entire directory tree. New files appearing in the tree are automatically included in the synchronization.

In subversion or CVS, new directories and files must be added explicitly using the command `svn add` or `cvs add`, respectively. 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 `svn update/status`

or `cvs update` are ignored due to the large number of files.

23.2.6 History

An additional feature of subversion or 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.

23.2.7 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. subversion or 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.

23.2.8 GUI

Unison offers a graphical user interface that displays the synchronization procedures Unison wants to perform. Accept the proposal or exclude individual files from the synchronization. In text mode, interactively confirm the individual procedures.

Experienced users normally run subversion or 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 or subversion. The resolution of conflicts is often much easier to perform with these front-ends.

23.2.9 User Friendliness

Unison and rsync are rather easy to use and are also suitable for newcomers. CVS and subversion are 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` or `svn update`. Then the data must be sent back to the repository with the command `cvs commit` or `svn commit`. Once this procedure has been understood, newcomers are also able to use CVS or subversion with ease.

23.2.10 Security against Attacks

During transmission, the data should ideally be protected against interception and manipulation.

Unison, CVS, rsync, and subversion can easily be used via ssh (secure shell), providing security against attacks of this kind. Running CVS or

Unison via rsh (remote shell) should be avoided. Accessing CVS with the *pserver* mechanism in insecure networks is likewise not advisable. subversion already provides the necessary security measures by running with Apache.

23.2.11 Protection against Data Loss

CVS has been used by developers for a long time to manage program projects and is extremely stable. As the development history is saved, CVS even provides protection against certain user errors, such as the unintentional deletion of a file. Despite subversion not being as common as CVS, it is already being employed in productive environments (for example, by the subversion project itself).

Unison is still relatively new, but boasts a high level of stability. However, it is more sensitive to user errors. Once the synchronization of the deletion of a file has been confirmed, there is no way to restore the file.

Table 23.1: *Features of the File Synchronization Tools: -- = very poor, - = poor or not available, o = medium, + = good, ++ = excellent, x = available*

	unison	**CVS/subv.**	**rsync**	**mailsync**
Client/Server	equal	C-S/C-S	C-S	equal
Portability	Lin,Un*x,Win	Lin,Un*x,Win	Lin,Un*x,Win	Lin,Un*x
Interactivity	x	x/x	x	-
Speed	-	o/+	+	+
Conflicts	o	++/++	o	+
File Selection	Directory	Selection/file, directory	Directory	Mailbox
History	-	x/x	-	-
Hard Disk Space	o	--	o	+
GUI	+	o/o	-	-
Difficulty	+	o/o	+	o
Attacks	+(ssh)	+/+(ssh)	+(ssh)	+(SSL)
Data Loss	+	++/++	+	+

23.3 Introduction to Unison

Unison is an excellent solution for synchronizing and transferring entire directory trees. The synchronization is performed in both directions and can be controlled by means of an intuitive graphical front-end. A console version can also be used. The synchronization can be automated so interaction with the user is not required, but experience is necessary.

23.3.1 Requirements

Unison must be installed on the client as well as on the server. In this context, the term *server* refers to a second, remote host (unlike CVS, explained in Section 23.1.2 on page 569).

In the following section, Unison is used together with ssh. In this case, an SSH client must be installed on the client and an SSH server must be installed on the server.

23.3.2 Using Unison

The approach used by Unison is the association of two directories (*roots*) with each other. This association is symbolic — it is not an online connection. In this example, the directory layout is as follows:

Client:	`/home/tux/dir1`
Server:	`/home/geeko/dir2`

You want to synchronize these two directories. The user is known as tux on the client and as `geeko` on the server. The first thing to do is to test if the client-server communication works:

```
unison -testserver /home/tux/dir1 ssh://geeko@server//homes/geeko/dir2
```

The most frequently encountered problems are:

- The Unison versions used on the client and server are not compatible.
- The server does not allow SSH connections.

- Neither of the two specified paths exists.

If everything works, omit the option `-testserver`.

During the first synchronization, Unison does not yet know the relationship between the two directories and submits suggestions for the transfer direction of the individual files and directories. The arrows in the 'Action' column indicate the transfer direction. A question mark means that Unison is not able to make a suggestion regarding the transfer direction as both versions were changed or are new.

The arrow keys can be used to set the transfer direction for the individual entries. If the transfer directions are correct for all displayed entries, simply click 'Go'.

The characteristics of Unison (e.g., whether to perform the synchronization automatically in clear cases) can be controlled by means of command-line parameters specified when the program is started. The complete list of all parameters can be viewed with `unison --help`.

For each pair, a synchronization log is maintained in the user directory `~/.unison`. Configuration sets, such as `~/.unison/example.prefs`, can also be stored in this directory.

To start the synchronization, specify this file as the command-line parameter as in `unison example.prefs`.

23.3.3 For More Information

The official documentation of Unison is extremely useful. For this reason, this section merely provides a brief introduction. The complete manual is available at `http://www.cis.upenn.edu/~bcpierce/unison/` and in the SUSE `unison`.

23.4 Introduction to CVS

CVS is suitable for synchronization purposes if individual files are edited frequently and are stored in a loose 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, as 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. In contrast, with the program Unison the data can be passed by host A through the hosts B and C to the server S.

23.4.1 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 CVS_ROOT=tux@server:/serverdir
```

The command `cvsinit` 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
```

23.4.2 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, the newly added files and directories can be committed: `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 (see above).

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.
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 and commited before in the same line, you might get a conflict, indicated with `C`.

In this case look at 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.

23.4.3 For More Information

This section merely offers a brief introduction to the many possibilities of CVS. Extensive documentation is available at the following URLs:

```
http://www.cvshome.org/
http://www.gnu.org/manual/
```

23.5 Introduction to Subversion

Subversion is a free open source versioning control system and is widely regarded as the successor to CVS, meaning that features already introduced for CVS are normally also in subversion. It is especially recommended when the advantages of CVS are sought without having to put up with its disadvantages. Many of these features have already been briefly introduced in Section 23.1.3 on page 569.

23.5.1 Installing a Subversion Server

The installation of a repository database on a server is a relatively simple procedure. Subversion provides a dedicated administration tool for this purpose. The command to enter for creating a new repository is:

```
svnadmin create /path/to/repository
```

Other options can be listed with `svnadmin help`. As opposed to CVS, subversion is not based on RCS, but rather on the Berkeley Database. Make sure not to install a repository on remote file systems, like NFS, AFS, or Windows SMB. The database requires POSIX locking mechanisms, which these file systems do not support.

The command `svnlook` offers a insight into a an existing repository.

```
svnlook info /path/to/repository
```

A server must be configured accordingly in order to allow other users to access the repository. It is possible to resort to the Apache webserver to this end or alternatively make use of svnserve, the server packaged with subversion. Once svnserver is up and running, the repository can be accessed with the schemata `svn://` or `svn+ssh://` in a URL. Those users which are supposed to authenticate themselves when calling `svn` can be set in `/etc/svnserve.conf`.

A decision in favor or against one or the other depends on many factors. It is hence recommended to browse the subversion book (More information about it can be found in section 23.5.3 on page 581

23.5.2 Usage and Operation

Use the command `svn` (similar to `cvs`) to access a subversion repository. The content provided by a correctly configured server fitted with a corresponding repository can be accessed by any client with the following command:

```
svn list http://svn.example.com/path/to/project
```

or

```
svn list svn://svn.example.com/path/to/project
```

Save an existing project in the current directory (check it out) with the command `svn checkout`:

```
svn checkout http://svn.example.com/path/to/project
    nameofproject
```

Checking out creates a new subdirectory `nameofproject/` on the client. Operation (adding, copying, renaming, deleting) can then be performed on it:

```
svn add file
svn copy oldfile newfile
svn move oldfile newfile
svn delete file
```

These commands can also be used on directories. subversion can additionally record properties of a file or directory:

```
svn propset license GPL foo.txt
```

The preceding example sets the value `GPL` for the property `license`. Display properties with `svn proplist`:

```
svn proplist --verbose foo.txt
 Properties on ’foo.txt’:
 license : GPL
```

Save the changes to the server with `svn commit`

Another user can incorporate your changes in his working directory by synchronizing with the server using `svn update`.

Unlike CVS, the status of a working directory in subversion can be displayed *without* accessing the repository with `svn status`. Local changes are displayed in five columns, with the first one being the most important one:

″ No changes.

’A’ Object is marked for addition.

’D’ Object is marked for deletion.

’M’ Object was modified.

’C’ Object is in conflict.

’I’ Object was ignored.

’?’ Object is not being maintained by versioning control.

’!’ Object is reported missing. This flag appears when the object was deleted or moved without the `svn` command.

’ ’ Object was being maintained as a file but has since been replaced by a directory or the opposite has occurred.

The second column shows the status of properties. The meaning of all other columns can be read in the subversion book (see the following section).

Use the command `svn help` to obtain the description of a parameter of a command:

```
svn help proplist
proplist (plist, pl): List all properties on files, dirs, or revisions
usage: 1. proplist [PATH...]
       2. proplist --revprop -r REV [URL]

  1. Lists versioned props in working copy.
  2. Lists unversioned remote props on repos revision.
...
```

23.5.3 For More Information

The first point of reference is the home page of the subversion project at `http://subversion.tigris.org/`. A highly recommendable book can be found in the directory `file:///usr/share/doc/packages/subversion/html/book.html` after installation of the package `subversion-doc` and is also available online at `http://svnbook.red-bean.com/svnbook/index.html`.

23.6 An 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.

23.6.1 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 provided above also involves the creation of a log file listing all connections. It 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.

23.6.2 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 latest news about rsync on the project web site of the project at `http://rsync.samba.org/`.

23.7 Introduction to mailsync

mailsync is mainly suitable for the following three tasks:

- Synchronization of locally stored e-mails with mails stored on a server
- Migration of mailboxes to a different format or to a different server
- Integrity check of a mailbox or search for duplicates

23.7.1 Configuration and Use

mailsync distinguishes between the mailbox itself (the *store*) and the connection between two mailboxes (the *channel*). The definitions of the stores and channels are stored in `~/.mailsync`. The following paragraphs explain a number of store examples.

A simple definition might appear as follows:

```
store saved-messages {
   pat Mail/saved-messages
prefix  Mail/
}
```

`Mail/` is a subdirectory of the user's home directory that contains e-mail folders, including the folder `saved-messages`. If mailsync is started with `mailsync -m saved-messages`, it lists an index of all messages in `saved-messages`. If the following definition is made

```
store localdir {
pat     Mail/*
prefix  Mail/
}
```

the command `mailsync -m localdir` lists all messages stored under `Mail/`. In contrast, the command `mailsync localdir` lists the folder names. The specifications of a store on an IMAP server appear as follows:

```
store imapinbox {
server {mail.edu.harvard.com/user=gulliver}
ref    {mail.edu.harvard.com}
pat    INBOX
}
```

The above example merely addresses the main folder on the IMAP server. A store for the subfolders would appear as follows:

```
store imapdir {
server {mail.edu.harvard.com/user=gulliver}
ref {mail.edu.harvard.com}
pat INBOX.*
prefix  INBOX.
}
```

If the IMAP server supports encrypted connections, the server specification should be changed to

```
server {mail.edu.harvard.com/ssl/user=gulliver}
```

or, if the server certificate is not known, to

```
server {mail.edu.harvard.com/ssl/novalidate-cert/user=gulliver}
```

The prefix is explained later.

Now the folders under `Mail/` should be connected to the subdirectories on the IMAP server:

```
channel folder localdir imapdir {
msinfo .mailsync.info
}
```

mailsync uses the `msinfo` file to keep track of the messages that have already been synchronized.

The command `mailsync folder` does the following:

- Expands the mailbox pattern on both sides.
- Removes the prefix from the resulting folder names.
- Synchronizes the folders in pairs (or creates them if they do not exist).

Accordingly, the folder `INBOX.sent-mail` on the IMAP server is synchronized with the local folder `Mail/sent-mail` (provided the definitions explained above exist). The synchronization between the individual folder is performed as follows:

- If a message already exists on both sides, nothing happens.
- If the message is missing on one side and is new (not listed in the `msinfo` file), it is transmitted there.
- If the message merely exists on one side and is old (already listed in the `msinfo` file), it is deleted there (because the message that had obviously existed on the other side was deleted).

To know in advance which messages will be transmitted and which will be deleted during a synchronization, start mailsync with a channel *and* a store with `mailsync folder localdir`. This command produces a list of all messages that are new on the local host as well as a list of all messages that would be deleted on the IMAP side during a synchronization. Similarly, the command `mailsync folder imapdir` produces a list of all messages that are new on the IMAP side and a list of all messages that would be deleted on the local host during a synchronization.

23.7.2 Possible Problems

In the event of a data loss, the safest method is to delete the relevant channel log file `msinfo`. Accordingly, all messages that only exist on one side are viewed as new and are therefore transmitted during the next synchronization.

Only messages with a message ID are included in the synchronization. Messages lacking a message ID are simply ignored, which means they are not transmitted or deleted. A missing message ID is usually caused by faulty programs when sending or writing a message.

On certain IMAP servers, the main folder is addressed with `INBOX` and subfolders are addressed with a randomly selected name (in contrast to INBOX and INBOX.name). Therefore, for such IMAP servers, it is not possible to specify a pattern exclusively for the subfolders.

After the successful transmission of messages to an IMAP server, the mailbox drivers (c-client) used by mailsync set a special status flag. For this reason, some e-mail programs, like mutt, are not able to recognize these messages as new. Disable the setting of this special status flag with the option `-n`.

23.7.3 For More Information

The `README` in `/usr/share/doc/packages/mailsync/`, which is included in `mailsync`, provides additional information. In this connection, RFC 2076 “Common Internet Message Headers” is of special interest.

Heterogenous Networks

In addition to connecting to other Linux systems, Linux is also able to connect to Windows and Macintosh computers and communicate over Novell networks. This chapter shows the requirements for and configuration of heterogenous networks.

24.1 Samba 588
24.2 Netatalk 599

24.1 Samba

24.1.1 Introduction to Samba

With the program Samba, convert a UNIX machine into a file and print server for DOS, Windows, and OS/2 machines. The Samba Project is run by the Samba Team and was originally developed by the Australian Andrew Tridgell.

Samba has now become a fully-fledged and rather complex product. This section presents an overview of its basic functionality. Samba offers plenty of online 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 is installed for more online documentation and examples. A commented example configuration (`smb.conf.SuSE`) can be found in the `examples` subdirectory.

Beginning from version 9.1, the SUSE LINUX `samba` package provides version 3 of the Samba suite, which brings some important added features:

- Support for Active Directory
- Much improved Unicode support
- The internal authentication mechanisms have been completely revised
- Improved support for the Windows 200x/XP printing system
- Servers can be set up as member servers in Active Directory domains
- Adoption of an NT4 domain, enabling the migration from the latter to a Samba domain.

Samba uses the SMB protocol (Server Message Block) 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.

Note

Migration to Samba3

There are some special points to take into account when migrating from Samba 2.x to Samba 3. A discussion of this topic is included in the Samba HOWTO Collection, where an entire chapter is dedicated to it. After installing the `samba-doc` package, find the HOWTO in `/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf`.

Note

NetBIOS

Note

s390, zseries: NetBIOS Support

IBM S/390 and zSeries merely support SMB over TCP/IP. NetBIOS support is not available on these systems.

Note

NetBIOS is a software interface (API) designed for communication between machines. Here, a name service is provided. It enables machines connected to the net 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, if 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 host names to make administration easier. This is the default used by Samba.

Clients

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 file-system kernel module for SMB that allows the integration of SMB resources on the Linux system level.

SMB servers provide hardware space to their clients by means of shares. A share includes a directory and its 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.

24.1.2 Installing and Configuring the Server

If you intend to use Samba as a server, install `samba`. The services required for Samba can be started with `rcnmb start && rcsmb start` and stopped with `rcsmb stop && rcnmb stop`.

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 net. 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 work group. 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. A misconfigured Samba server can cause serious problems when trying to become LMB for its workgroup. Contact your administrator and subject your configuration to some heavy testing either 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 24.1: *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 net. 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 24.2: *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 identical to the user name.

`valid users = %S` `%S` is replaced by the concrete name of the share as soon as a connection has been successfully established. For a `[homes]` share, this is always identical to the user's name. As a consequence, access rights to a user's share are restricted exclusively to the user.

`browseable = No` This setting enables the share to be 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 `writeable = 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. Therefore, if you want the group to have read or write access, the line `valid users = %S` must be deactivated.

Security Levels

The SMB protocol comes from the DOS and Windows world and directly takes into consideration the problem of 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 registering, the server can grant access to individual exported shares dependent on user names.

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 distinction between share, user, and 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`.

Note

For simple administration tasks with the Samba server, there is also the program `swat`. It provides a simple web interface with which to configure the Samba server conveniently. In a web browser, open `http://localhost:901` and log in as user `root`. However, `swat` must also be activated in the files `/etc/xinetd.d/samba` and `/etc/services`. To do so in `/etc/xinetd.d/samba`, edit the `disable` line so it reads: `disable = no`. More information about `swat` is provided in the man page.

Note

24.1.3 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. This can be brought about with the help of a Samba server. In a Windows-based network, this task is handled by a Windows NT server configured as a primary domain controller (PDC). The entries that must be made in the `[global]` section of `smb.conf` are shown in File 24.3.

Example 24.3: *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. This is done 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 24.4: *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 24.5:** Automated Setup of a Machine Account*

```
add machine script = /usr/sbin/useradd -g machines \
                  -c "NT Machine Account" -d \
                  /dev/null -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`.

24.1.4 Installation and Configuration with YaST

In the 'Start up' menu (Figure 24.1 on the next page), select whether or not to start Samba. If you activate Samba, the service will be started every time you boot the system.

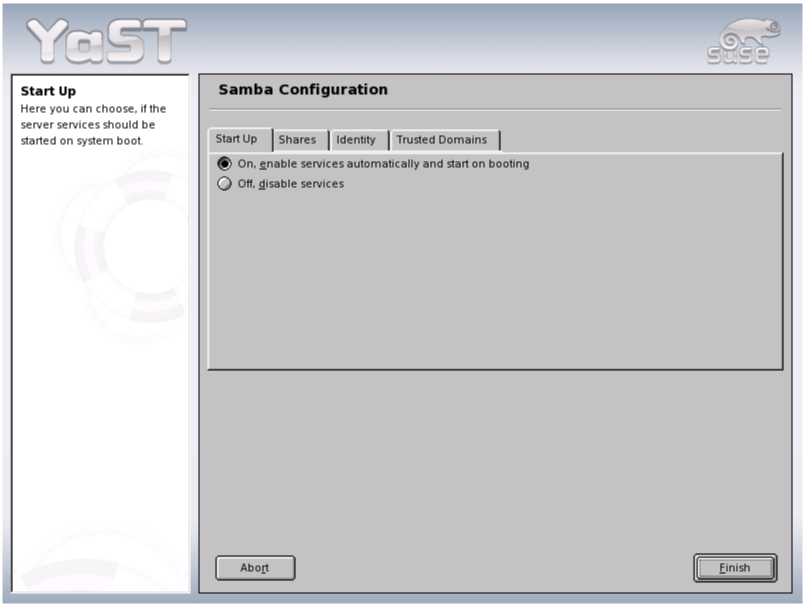
In the 'shares' menu (Figure 24.2 on page 598), determine the Samba shares to be activated. Use the 'Toggle status' button to switch between 'active' and 'inactive'. Click 'Add' to add new shares.

In the 'Identity' menu (Figure 24.3 on page 599), determine the domain the host is associated with ('Base Settings') and whether or not to use an alternative host name in the network ('NetBIOS hostname'). Moreover, you can configure the host as a WINS server. If this is not the case, specify the IP address of the WINS server. If you enter an asterisk (`*`), YaST will automatically find the WINS server.

In the 'Trusted Domains' menu (Figure 24.4 on page 600), determine which domains the host is to trust. This means that you adopt the respective settings of the trusted domain.

24.1.5 Installing Clients

Clients can only access the Samba server via TCP/IP. NetBEUI and NetBIOS via IPX cannot be used with Samba.



Start Up
Here you can choose, if the server services should be started on system boot.

Samba Configuration

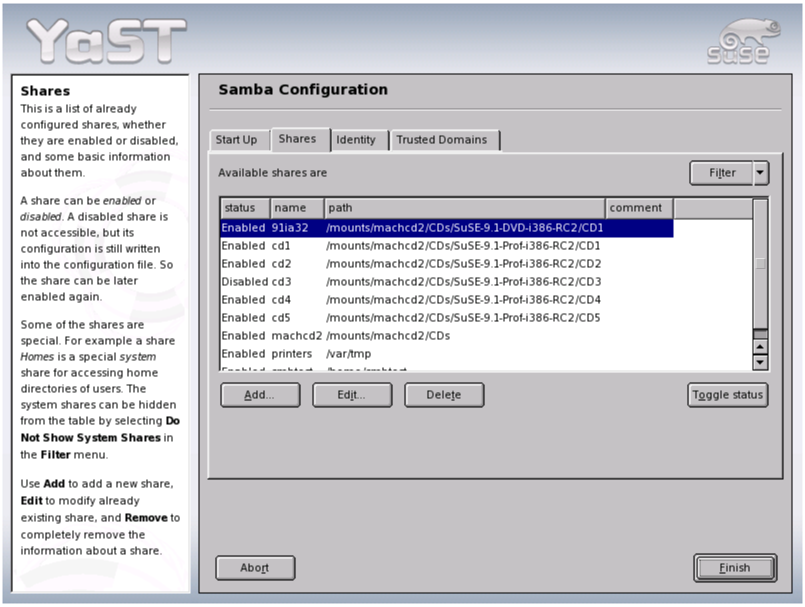
Start Up | Shares | Identity | Trusted Domains

● On, enable services automatically and start on booting
○ Off, disable services

Figure 24.1: Samba Configuration -- Startup

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.



Shares

This is a list of already configured shares, whether they are enabled or disabled, and some basic information about them.

A share can be *enabled* or *disabled*. A disabled share is not accessible, but its configuration is still written into the configuration file. So the share can be later enabled again.

Some of the shares are special. For example a share *Homes* is a special *system* share for accessing home directories of users. The system shares can be hidden from the table by selecting **Do Not Show System Shares** in the **Filter** menu.

Use **Add** to add a new share, **Edit** to modify already existing share, and **Remove** to completely remove the information about a share.

Samba Configuration

Available shares are

status	name	path	comment
Enabled	91ia32	/mounts/machcd2/CDs/SuSE-9.1-DVD-i386-RC2/CD1	
Enabled	cd1	/mounts/machcd2/CDs/SuSE-9.1-Prof-i386-RC2/CD1	
Enabled	cd2	/mounts/machcd2/CDs/SuSE-9.1-Prof-i386-RC2/CD2	
Disabled	cd3	/mounts/machcd2/CDs/SuSE-9.1-Prof-i386-RC2/CD3	
Enabled	cd4	/mounts/machcd2/CDs/SuSE-9.1-Prof-i386-RC2/CD4	
Enabled	cd5	/mounts/machcd2/CDs/SuSE-9.1-Prof-i386-RC2/CD5	
Enabled	machcd2	/mounts/machcd2/CDs	
Enabled	printers	/var/tmp	

Figure 24.2: *Samba Configuration -- Shares*

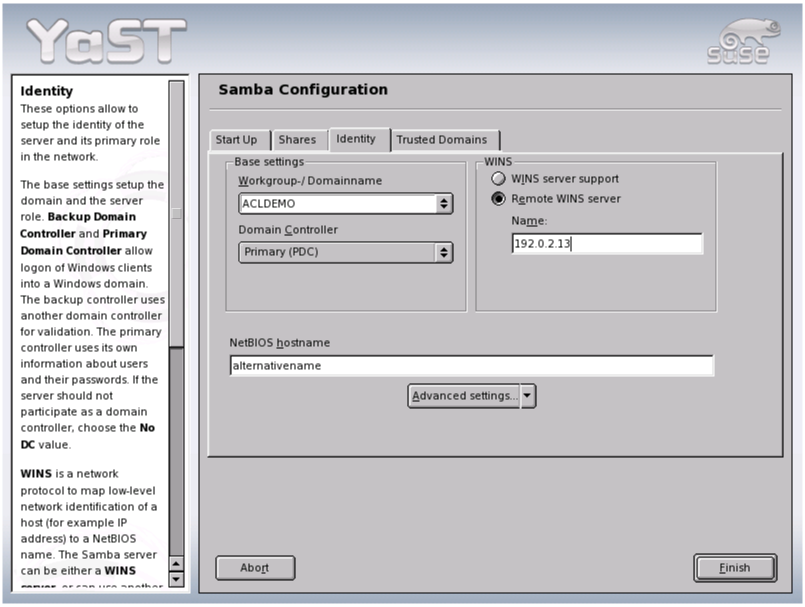
Note

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.

Note

24.1.6 Optimization

`socket options` is one possible optimization provided with the sample configuration that ships with your Samba version. Its default configuration refers to a local ethernet network. For additional information about `socket options`, refer to the section *socket options* in the manual pages of `smb.conf` and to the manual page of `socket(7)`. Further information is provided in the `Samba performance tuning` chapter of the Samba HOWTO Collection.



Identity

These options allow to setup the identity of the server and its primary role in the network.

The base settings setup the domain and the server role. **Backup Domain Controller** and **Primary Domain Controller** allow logon of Windows clients into a Windows domain. The backup controller uses another domain controller for validation. The primary controller uses its own information about users and their passwords. If the server should not participate as a domain controller, choose the **No DC** value.

WINS is a network protocol to map low-level network identification of a host (for example IP address) to a NetBIOS name. The Samba server can be either a **WINS**

Samba Configuration

Workgroup-/ Domainname: ACLDEMO

Domain Controller: Primary (PDC)

○ WINS server support

● Remote WINS server

Name: 192.0.2.13

NetBIOS hostname: alternativename

Figure 24.3: *Samba Configuration -- Identity*

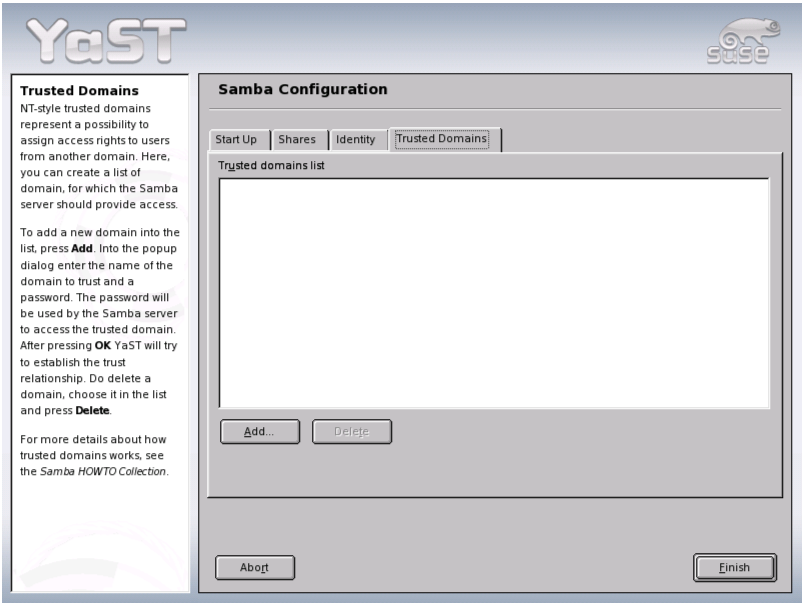
The standard configuration in `/etc/samba/smb.conf` is designed to provide useful settings based on the default settings of the Samba team. However, a ready-to-use configuration is not possible, especially in view of the network configuration and the workgroup name. The commented sample configuration `examples/smb.conf.SuSE` contains information that is helpful for the adaption to local requirements.

Note

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.

Note

24.2 Netatalk



Trusted Domains

NT-style trusted domains represent a possibility to assign access rights to users from another domain. Here, you can create a list of domain, for which the Samba server should provide access.

To add a new domain into the list, press **Add**. Into the popup dialog enter the name of the domain to trust and a password. The password will be used by the Samba server to access the trusted domain. After pressing **OK** YaST will try to establish the trust relationship. Do delete a domain, choose it in the list and press **Delete**.

For more details about how trusted domains works, see the *Samba HOWTO Collection*.

Figure 24.4: *Samba Configuration -- Trusted Domains*

Note

Netatalk on S/390 und zSeries

IBM S/390 and zSeries can use the AppleTalk protocol over TCP/IP. Genuine AppleTalk is not supported.

Note

With `Netatalk`, obtain a high-performance file and print server for MacOS clients. With it, access data on a Linux machine from a Macintosh or print to a connected printer. `Netatalk` is a suite of Unix programs that run on kernel-based DDP (Datagram Delivery Protocol) and implement the AppleTalk protocol family (ADSP, ATP, ASP, RTMP, NBP, ZIP, AEP, and PAP).

AppleTalk is, in effect, an equivalent to the more familiar protocol TCP (Transmission Control Protocol). It has counterparts to many TCP/IP-based services, including services for resolving host names and time synchronization. For example, the command `aecho` (AEP, AppleTalk Echo Protocol) is used instead of `ping` (ICMP ECHO_REQUEST, Internet Control Message

Protocol).

The three daemons described below are normally started on the server:

- atalkd ("AppleTalk Network Manager"), which corresponds to the program ip
- afpd ("AppleTalk Filing Protocol daemon"), which provides an interface for Macintosh clients to Unix file systems.
- papd ("Printer Access Protocol daemon"), which makes printers available in the (AppleTalk) network.

Server directories can be exported with Netatalk at the same time as with Samba for Windows clients (see Section 24.1.1 on page 589) and via NFS (see Section 21.10 on page 518), which is very useful in heterogeneous network environments. This centralizes the management of data backup and user permissions on the Linux server.

There are a number of limitations when working with `Netatalk`:

- Due to Macintosh client restrictions, the user passwords on the server cannot be longer than eight characters.
- Macintosh clients cannot access Unix files with names longer than 31 characters.
- File names may not contain colons (`:`) because they serve as path name separators in MacOS.

24.2.1 Configuring the File Server

In the default configuration, `Netatalk` is already fully functional as a file server for home directories of the Linux system. To use the extended features, define some settings in the configuration files. These are located in the `/etc/netatalk/` directory.

All configuration files are pure text files. Text that follows a hash mark # (comments) and empty lines can be disregarded. The various services (printing, Appletalk broadcast, Appletalk via TCP/IP, time server) can be activated through the file `/etc/netatalk/netatalk.conf`:

```
ATALKD_RUN=yes
PAPD_RUN=yes
AFPD_RUN=yes
TIMELORD_RUN=no
```

Configuring the Network — atalkd.conf

Define, in `/etc/netatalk/atalkd.conf`, over which interfaces services are provided. This is usually `eth0`. In the example file that comes with `Netatalk`, this is the case. Enter additional interfaces to use several network cards at the same time. When the server is started, it searches the network for existing zones and servers and modifies the corresponding lines by entering the set AppleTalk network addresses. You will then find a line such as

```
eth0 -phase 2 -net 0-65534 -addr 65280.57
```

at the end of the file. For more complex configurations, refer to examples in the configuration file. Find documentation about additional options in the manual page of afpd.

► **s390, zseries**
IBM S/390 and zSeries only support the network configuration over the IP protocol. The procedure is not supported under AppleTalk. ◄

Defining File Servers — afpd.conf

The `afpd.conf` file contains definitions for how your file server appears on MacOS machines as an item under the 'Chooser' dialog. As is the case with the other configuration files, these also contain detailed comments explaining the wide variety of options.

If you do not change anything here, the default server is simply started and displayed with the host name in the 'Chooser'. Therefore, you do not necessarily need to enter anything. However, you can give additional file servers a variety of names and options here, for example, to provide a specific guest server on which everybody can save files as "guest".

```
"Guest server" -uamlist uams_guest.so
```

Define a server that denies guests access, but which is only accessible for users who already exist in the Linux system with:

```
"Font server" -uamlist uams_clrtxt.so,uams_dhx.so
```

This behavior is controlled by the option `uamlist` followed by a list of authentication modules to use separated by commas. If you do not provide this option, all procedures are active by default.

An AppleShare server not only provides its services by default via AppleTalk, but also via TCP/IP (*encapsulated*). The default port is 548. Assign dedicated ports to additional AppleShare servers (on the same machine) if these should also run via TCP. The availability of the service via TCP/IP enables access to the server even over non-AppleTalk networks, such as the Internet. In this case, the syntax would read:

```
"Font server" -uamlist uams_clrtxt.so,uams_dhx.so -port 12000
```

The AppleShare server, set to the port 12000, then appears in the network with the name `Font server` and does not allow guest access. In this way, it is also accessible via TCP/IP routers.

The file `AppleVolumes.default` (described in detail below) defines which directories located on the server are made available by each AppleShare server as network *volumes*. By using the `-defaultvol` option for a given AppleShare server, specify another file that defines different directories. The corresponding command (read as one line) is:

```
"Guest server" -uamlist uams_guest.so -defaultvol
/etc/netatalk/AppleVolumes.guest
```

Further options are explained in the `afpd.conf` file itself.

Directories and Access Permissions — AppleVolumes.default

Here, define directories to export. The access permissions are defined with the customary Unix user and group permissions. This is configured in the `AppleVolumes.default` file. Along with `AppleVolumes.default`, additional files can be created, such as `AppleVolumes.guest`, used by some servers (by giving the option `-defaultvol` in the `afpd.conf` file. See the previous section).

Note

Here, the syntax has partially changed. Take this into consideration if you are updating this version from a previous one. For example, it is now `allow:` instead of `access=` (a typical symptom would be if, instead of the drive descriptions, you were to see a display of the drive options on the Mac clients in the 'Chooser'). Because the new files are created with the `.rpmnew` endings during an update, it is possible that your previous settings may no longer function as a result of the modified syntax. Create backups of your configuration files, copy your old configuration into the new files, then rename these files to the proper names. This way, benefit from the current comments contained in the configuration files, which provide a detailed explanation of the options.

Note

The example shown here:

```
/usr/local/psfonts "PostScript Fonts"
```

indicates that the Linux directory `/usr/local/psfonts`, located in the root directory, is available as an AppleShare volume with the name "PostScript Fonts".

Options are separated by a space and attached to the end of a line. A very useful option is the access restriction:

```
/usr/local/psfonts "PostScript Fonts" allow:User1,@group0
```

This restricts access to the volume "PostScript Fonts" to the user "User1" and all members of the group "group0". The users and groups entered here must be known to the Linux system. Likewise, explicitly deny users access with `deny:User2`. These restrictions only apply to access via AppleTalk and not to the normal access rights users have if they can log in to the server itself.

Netatalk maps the customary Resource Fork of MacOS files to `.AppleDouble/` directories in the Linux file system. Using the `noadouble` option, set these directories to be created only when they are actually needed. The syntax is:

```
/usr/local/guests "Guests" options:noadouble
```

Additional options and features can be found in the explanations included in the file itself.

The tilde (~) in this configuration file stands for the home directory for each and every user on the server. This way, every user can easily access his home directory without each one being defined explicitly here. The example file installed already includes a tilde, which is why `Netatalk` makes the home directory available by default as long as you do not modify anything in this file.

afpd also searches for a file `Applevolumes` or `.Applevolumes` in the home directory of a user logged in to the system. Entries in this file supplement the entries in the server files `AppleVolumes.system` and `AppleVolumes.default` to enable individual type and creator file settings and to access specific directories. These entries are extensions and do not allow access for the user for whom access permission is denied from the server side.

The `netatalk.pamd` file is used, via PAM (pluggable authentication modules), for authentication purposes. Using PAM is, however, irrelevant in this context.

File Specifications — AppleVolumes.system

In the `AppleVolumes.System` file, define which customary MacOS type and creator specifications are assigned to certain file endings. An entire series of default values are already predefined. If a file is displayed by a generic white icon, there is not yet an entry for it in this file. If you encounter a problem with a text file belonging to another system, which cannot be opened properly in MacOS or vice versa, check the entries there.

24.2.2 Configuring the Print Server

Make a laserwriter service available by configuring the `papd.conf` file. The printer must be already functioning locally with `lpd`, so configure a printer as described in Chapter 13 on page 289. If you can print a text file locally using the command `lpr file.txt`, the first step has been successfully completed.

You do not necessarily need to enter anything in `papd.conf` if a local printer is configured in Linux, because print jobs can simply be forwarded to the print daemon lpd without additional settings. The printer registers itself in the AppleTalk network as Laserwriter. You can, however, extend your printer entries as follows:

```
Printer_Reception:pr=lp:pd=/etc/netatalk/kyocera.ppd
```

This causes the printer named `Printer_Reception` to appear as a 'Chooser' item. The corresponding printer description file is usually provided by the vendor. Otherwise, refer to the file `Laserwriter` located in the 'System Extensions' folder. However, when using this file you often cannot use all of the printer's features.

24.2.3 Starting the Server

The server can be started at system boot time via its init script or manually with `rcatalk start`. The init script is located at `/etc/init.d/netatalk`. The actual starting of the server takes place in the background. It takes about a minute until the AppleTalk interfaces are set up and responsive. Check for the status as shown in the following (all servers are running if `OK` is reported three times):

```
rcatalk status

Checking for service atalk:OKOKOK
```

From a Mac running MacOS, check for AppleTalk activation, choose 'File-sharing', then double-click 'AppleShare'. The names of the servers should then appear in the window. Double-click a server and log in. It should then be possible to access a shared volume.

The procedure is a bit different for AppleShare servers configured to use TCP only (and no DDP). To connect, press 'Server IP address' and enter the respective IP address. If necessary, append the port number, separated by a colon (`:`).

Note

The S/390 and zSeries are only able to "understand" AppleTalk if the AppleShare server is configured to use TCP/IP encapsulated AppleTalk.

Note

24.2.4 Additional Information

To take full advantage of all the options `Netatalk` offers, read the corresponding manual pages. Find them by entering the command `rpm -qd netatalk`. The `/etc/netatalk/netatalk.conf` file is not used in our `Netatalk`version, so disregard it. Helpful URLs:

- `http://netatalk.sourceforge.net/`
- `http://www.umich.edu/~rsug/netatalk/`
- `http://www.anders.com/projects/netatalk/`

Internet

The Internet has become the number one platform for network communications worldwide. As true network system, Linux can handle a broad range of Internet related tasks — both as a server and as a client system. This chapter discusses some of the topics relevant to the Internet: the configuration of the smpppd (the SUSE Meta PPP Daemon), the manual configuration of ADSL access, and the configuration of the Squid proxy.

► **s390, zseries**
The only relevant paragraph for IBM S/390 and zSeries is the one about Squid. smpppd and ADSL are not supported on these platforms. ◄

25.1 The smpppd as Dial-up Assistant 610
25.2 Configuring an ADSL or T-DSL Connection 612
25.3 Proxy Server: Squid 614

25.1 The smpppd as Dial-up Assistant

25.1.1 Program Components for the Internet Dial-Up

Most home users do not have a dedicated line connecting them to the Internet. Rather, they use dial-up connections. Depending on the dial-up method (ISDN or DSL), the connection is controlled by the ipppd or the 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.

25.1.2 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 as follows:

open-inet-socket = `<yes|no>` To control smpppd via the network, this option must be set to `yes`. The port on which the 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>` If a host has several IP addresses, this parameter can be used to determine by which IP address smpppd should accept connections.

host-range = `<min ip> <max ip>`
The parameter `host-range` can be used to define 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 entitled to access smpppd.

More information about smpppd is available in `man 8 smpppd` and `man 5 smpppd.conf`.

25.1.3 Configuring kinternet and cinternet for Remote Use

The programs kinternet and cinternet can be used locally as well as for controlling a remote smpppd. cinternet is the the command-line counterpart of the graphical kinternet. 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:

server = `<server>` Here, specify the host on which smpppd runs. If this host is the same as the default gateway of the host, it is sufficient to set the `gateway-fallback` to `yes`.

gateway-fallback = `<yes|no>` If no server was specified and there is no local smpppd, this option, which is enabled by default, causes a search for an smpppd on the default gateway.

password = `<password>` Insert the password selected for smpppd.

If the smpppd is active, you can now try to access it. This can be done with the command `cinternet --verbose --interface-list`. If you experience difficulties at this point, refer to `man 5 smpppd-c.conf` and `man 8 cinternet`

25.2 Configuring an ADSL or T-DSL Connection

25.2.1 Default Configuration

Currently, SUSE LINUX supports DSL connections that work with the point-to-point over ethernet protocol (PPPoE) used by most major providers. If you are not sure what protocol is used for your DSL connections, ask your provider. If you have a single-user workstation with a graphical interface, the DSL connection should be set up with the YaST modules ADSL or T-DSL.

1. The `ppp` and `smpppd` packages must be installed. It is best to use YaST for this purpose.
2. Configure your network card with YaST. Do not activate `dhcp`, but set a fixed IP address instead, such as `192.168.2.22`.
3. The parameters set with the DSL module of YaST are saved in the file `/etc/sysconfig/network/providers/provider0`. In addition, there are configuration files for the smpppd (SUSE meta ppp daemon) and its front-ends kinternet and cinternet. For information, refer to `man smpppd`.
4. If necessary, start the network with the command `rcnetwork start` and smpppd with the command `rcsmpppd start`.
5. On a system without a graphical user interface, use the commands `cinternet --start` and `cinternet -stop` to establish or terminate a connection. On a graphical user interface, this can be done with kinternet. This program is started automatically in KDE if you used YaST to set up DSL. Click the gear icon in the control panel. Select 'Communication/Internet' → 'Internet Tools' → 'kinternet'. A plug icon then appears in the control panel. You can now start the connection with a simple click on the icon and terminate the connection later with another click.

25.2.2 DSL Connection by Dial-on-Demand

Dial-on-demand means that the connection is automatically set up when the user goes online, for example, when visiting a web site in a browser or when sending an e-mail. After a certain amount of idle time when no data is sent or received, the connection is automatically be dropped. Because the dial-up connection via PPPoE, the protocol for ADSL, is very fast, it seems as if it were a dedicated line to the Internet.

Using dial-on-demand, however, really only makes sense if you have a flat-rate connection. If you use it but are charged for time online, make sure there are no interval processes, such as a cronjob, periodically establishing a connection. This could get quite expensive.

Although a permanent online connection would also be possible using a DSL flat-rate connection, there are certain advantages to having a connection that only exists for a short amount of time when needed:

- Most providers drop the connection after a certain period of time.
- A permanent connection can be considered as a drain on resources (e.g., IP addresses).
- Being online permanently is a security risk, because hackers may be able to comb the system systematically for vulnerable areas. A system that is only accessible over the Internet when necessary and is always changing IP addresses is significantly more difficult to attack.

Dial-on-demand can be enabled using YaST (also refer to the *User Guide*). Alternatively, set it up manually:

Set the parameter `DEMAND=yes` in the `/etc/sysconfig/network/providers/provider0` file then define an idle time via the variable `IDLETIME=60`. This way, an unused connection is dropped after sixty seconds.

To set up a DSL gateway for private networks, refer to the following article from our support portal: `http://portal.suse.de/sdb/en/2002/07/masq80.html`

25.3 Proxy Server: Squid

Squid is a widely-used proxy cache for Linux and UNIX platforms. The chapter discusses its configuration, the settings required to get it running, how to configure the system to do transparent proxying, how to gather statistics about using the cache with the help of programs like Calamaris and cachemgr, and how to filter web contents with squidGuard.

25.3.1 Squid as Proxy Cache

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. Thus, the clients receive the data much faster than from the Internet. This procedure also reduces the network traffic.

Apart from 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 proxies normally 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.

25.3.2 Some Facts about Proxy Caches

Squid and Security

It is also 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 way of the proxy.

If the firewall configuration includes a DMZ, the proxy should operate within this zone. In this case, it is important that all computers in the DMZ send their log files to hosts inside the secure network. The possibility of implementing a *transparent* proxy is covered in Section 25.3.6 on page 624.

Multiple Caches

Several proxies can be configured in such a way that objects can be exchanged 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 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 (Hyper Text 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.

Note

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.

Note

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.

25.3.3 System Requirements

The most important thing is to determine the maximum 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.

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 will 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 much quicker. Fast SCSI hard disks nowadays have a seek time of under 4 milliseconds.

One possibility to speed up the system is to use a number of disks concurrently or to employ striping RAID arrays.

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 so the less requested objects are replaced by newer ones. On the other hand, if, for example, "1 GB" is available for the cache and the users only surf "10 MB" a 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.

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 25.3.7 on page 627.

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.

25.3.4 Starting Squid

Squid is already preconfigured in SUSE LINUX, so you can start it easily right after installation. A prerequisite for a smooth start is an already configured network, at least one name server, and Internet access. Problems can arise if a dial-up connection is used with a dynamic DNS configuration. In cases such as this, at least the name server should be clearly entered, because Squid does not start if it does not detect a DNS server in `/etc/resolv.conf`.

To start Squid, enter `rcsquid start` at the command line as `root`. For the initial start-up, the directory structure must first be defined in `/var/squid/cache`. This is done by the start script `/etc/init.d/squid` automatically 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 all users to access Squid and, through it, the Internet, change the entry in the configuration file `/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 25.3.5 on page 622.

After modifying the configuration file `/etc/squid/squid.conf`, Squid must reload the configuration file. This can be done with the following command: `rcsquid reload`. Alternatively, do a complete restart of Squid with `rcsquid restart`.

Another important command is `rcsquid status`, which allows you to determine whether the proxy is running. Finally, 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.

Caution

Terminating Squid

Terminating Squid with `kill` or `killall` can destroy the cache. In order to be able to restart Squid, the cache must be deleted.

Caution

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/squid/logs/cache.log`. If Squid should be loaded automatically when the system boots, use the YaST runlevel editor to activate Squid for the desired runlevels.

An uninstall of Squid does not remove the cache or the log files. To remove these, delete the `/var/cache/squid` directory manually.

Local DNS Server

Setting up a local DNS server, such as BIND9, makes sense even if the server does not manage its own domain. It then simply acts as a caching-only DNS and is also able to resolve DNS requests via the root name servers without requiring any special configuration. If you enter the local DNS server in the `/etc/resolv.conf` file with the IP address `127.0.0.1` for `localhost`, Squid should always find a valid name server when it starts. For this to work, it is sufficient just to start the BIND server after installing the corresponding package. The name server of the provider should be entered in the configuration file `/etc/named.conf` under `forwarders` along with its IP address. However, if you have a firewall running, you need to make sure that DNS requests can pass it.

25.3.5 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 the `localhost`. The default port is `3128`. The preinstalled file `/etc/squid/squid.conf` provides detailed information on the options and many examples. Nearly all entries begin with a # sign (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. In this way, easily interpret the default values and the changes.

Note

Update from Version 2.4 to Version 2.5

Following an update of Squid from version 2.4 to version 2.5, the cache of Squid must be deleted, as the directory structure was changed.

Note

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 implement the old `squid.conf`, run a risk that the configuration will no longer function, because options are sometimes modified and new changes added.

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, for example, enter a parent proxy to use the proxy of your ISP. As <*hostname*>, enter the name and IP address of the proxy to use and, as <*type*>, *parent*. For <*proxy-port»*, enter the port number that is also set 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* should 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 the caches. The default is `8 MB`.

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 sixteen 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 fifty to a maximum of eighty percent 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
path for log messages

cache_log /var/log/squid/cache.log
path for log messages

cache_store_log /var/log/squid/store.log
path for log messages

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 the logged IP addresses in the log files to hide the clients' identity. The last digit of the IP address is set to zero if you enter `255.255.255.0` here.

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 can 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 is carried out by a cronjob found 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 the HTTP requests.

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 will make a note of the failed requests then refuse 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.

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 host names, 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 package in and of itself that can be separately installed and configured.

auth_param basic program /usr/sbin/pam_auth

If users must be authenticated on the proxy, a corresponding program, such as pam_auth, can be set here. When accessing pam_auth for the first time, the user sees a login window in which to enter the user name 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 user names 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 `Windows`, there is free software available to download from the Internet. To ensure that only clients with a successful ident lookup are permitted, a corresponding ACL also needs to be defined here:

```
acl identhosts ident REQUIRED

http_access allow identhosts
http_access deny all
```

Here, too, replace *REQUIRED* with a list of permitted user names. Using *ident* can slow down the access time quite a bit, because ident lookups are repeated for each request.

25.3.6 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 or not.
- 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.

Kernel Configuration

First, make sure that the kernel of the proxy server supports a transparent proxy. If not, add these options to the kernel and compile it anew. For more details, refer to Section 9 on page 227. Kernel modules are subject to change from version to version. Check the latest status in `/usr/share/doc/howto/en/html/mini/TransparentProxy-3.html` or on the Internet: `http://www.tldp.org/HOWTO/mini/TransparentProxy-3.html`.

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

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. Its configuration file can be found in `/etc/sysconfig/SuSEfirewall2`. The configuration file consists of well-documented entries. Even to set only a transparent proxy, you must configure some firewall options:

- Device pointing to the Internet: `FW_DEV_EXT`="eth1"
- Device pointing to the network: `FW_DEV_INT`="eth0"

Set ports and services (see `/etc/exports`) on the firewall permitted access from untrusted 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/exports`) on the firewall permitted access from the secure network, both TCP and UDP services:

```
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 the number `15`:

Example 25.1: *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. As Squid supports more protocols 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_TCP="192.168.0.0/16,0/0,80,3128 192.168.0.0/16,0/0,21,3128"
FW_REDIRECT_UDP="192.168.0.0/16,0/0,80,3128 192.168.0.0/16,0/0,21,3128"
```

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 25.3.4 on page 618. 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`.

25.3.7 Squid and Other Programs

In the following, see how other applications interact with Squid. cachemgr.cgi enables the system administrator to check the amount of memory needed for caching objects. squidGuard filters web pages. Calamaris is a report generator for Squid.

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.

Setup

First, a running web server on your system is required. 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 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/
```

Cache Manager ACLs in /etc/squid/squid.conf

There are some default settings in the original file required for the cache manager:

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
```

With the following rules:

```
http_access allow manager localhost
http_access deny manager
```

the first ACL is the most important, as the cache manager tries to communicate with Squid over the cache_object protocol.

The following 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 File 25.2.

Example 25.2: *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 as in File 25.3.

Example 25.3: *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. This can easily be done with `rcsquid reload`.

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`.

squidGuard

This section is not intended to go through 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 be used for 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 info page
- redirect unregistered users to a registration form
- redirect banners to an empty GIF
- have different access rules based on time of day, day of the week, date, etc.
- have different rules for different user groups

Neither squidGuard or "Squid" can be used to:

- Edit, filter, or censor text inside documents
- Edit, filter, or censor HTML-embedded script languages, such as JavaScript or VBscript

Using squidGuard

Install `squidGuard`. Edit a minimal configuration file `/etc/squidguard.conf`. There are plenty of 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.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 lots of requests: on a 500 MHz Pentium with 5,900 domains and 7,880 URLs (totalling 13,780), 100,000 requests can be processed within 10 seconds. Therefore, it is not recommended to set more than four processes, as 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.

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 use of the program is quite easy.

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 of all available reports

-w output as HTML report

-l 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://web.onda.com.br/orso/`.

More Information about Squid

Visit the home page of Squid at `http://www.squid-cache.org/`. Here you can 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 `howtoen` under `/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/`.

Security in the Network

The security of data, services, and money transfers within networks is and will always be an important issue. This chapter provides information about how to prevent unauthorized access to the system, and how guard against attacks from the outside.

The establishment of a CA (Certification Authority) makes it possible to encrypt communications throughout the network, using techniques such as a VPN (Virtual Private Network). Other mechanisms such as masquerading, firewalls, and Kerberos can be used to control the exchange of data and the general data traffic. And the Secure Shell (SSH) allows users to log in to remote hosts by way of an encrypted connection. Apart from these purely technical instructions, this chapter also includes information about the more general security aspects of a Linux network.

26.1 X.509 certification with YaST 634
26.2 VPN with SUSE LINUX 647
26.3 Masquerading and Firewalls 657
26.4 SSH — Secure Shell, the Safe Alternative 665
26.5 Network Authentication — Kerberos 671
26.6 Installing and Administering Kerberos 678
26.7 Security and Confidentiality 695

26.1 X.509 certification with YaST

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 not only used for communication, but can also be found on company ID cards, for example. The generation and administration of certificates is mostly handled by "official" institutions that offer this as a commercial service. In different 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. SUSE LINUX offers two YaST modules for this purpose, which offer elementary management functions for digital X.509 certificates. The following sections offer an insight into the principles of digital certification and explain how to use YaST to create and administer certificates of this type. However, the whole topic of "digital certification" is extremely complex, so that you could fill several books if you were to deal with it exhaustively. Thus, the following descriptions can only offer an overview. For more detailed information please see the related literature list in the print media and on the Internet, e.g. under:

- `http://www.ietf.org/html.charters/pkix-charter.html` (English)
- `http://www.pca.dfn.de/certify/` (German)

26.1.1 The principles of digital certification

Digital certification uses cryptographic processes to encrypt data, thereby protecting it from access by unauthorized persons. 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 public key is circulated by the key owner for use by third parties.

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 should be sure that a public key does indeed belong to the assumed owner. The assignment of users and public keys will be confirmed by trustworthy instances by means of 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 instances are usually part of a certification infrastructure which, in addition to issuing and signing certificates is also responsible for the other aspects of certificate management. This includes 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 "each party" in the "web of trust".

The hierarchically structured *X.509 Public Key Infrastructure* (PKIX) is an alternative model that was defined by the *IETF* (*Internet Engineering Task Force*) and that now acts as an exemplar for almost all publicly used PKIs. In this model authentication is carried out in a hierarchical tree structure by so-called *Certification Authorities* (CA). The root of the tree is formed by the root CA, which certifies all sub-CAs or the next level own to the sub-CAs of the lowest level which issue user certificates. The user certificates become trustworthy through certification by the next highest sub-CAs, which in turn have been certified by the higher levels of the hierarchy. This creates a "certification path" that ends with the root CA. The security of such a PKI therefore stands and falls with the trustworthiness of the CA certificates. In order to make certification practices transparent for PKI customers, the PKI operator defines a *Certification Practice Statement* (CPS) in which the procedures for certificate management are defined. This should ensure that the PKI only issues really trustworthy certificates.

X.509 certificates

An X.509 certificate is a data structure with several fixed fields and (optional) additional extensions. The fixed fields mainly contain the name of the key owner, his/her 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. For this reason a field is also provided for this date. The CA guarantees the validity of the certificate in the specified period. The CPS usually stipulates that the PKI (in other words the CA in the final analysis) should create and distribute a new certificate before validity lapses.

The extensions can contain any additional information. An application does not normally need to be able to evaluate an extension unless it is identified as *critical*. If an application does not recognize a critical extension, it must reject the certificate. Some extensions reduce the use of the certificate to a specific application, e.g. signature or encryption.

The 26.1 table shows the principle underlying a X.509 certificate in version 3.

Table 26.1: *X.509v3 certificate*

Field	Content
Version	The version of the certificate, e.g. v3.
Serial Number	Unique certificate ID (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 (from...to).
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)
Extensions	Optional additional information, e.g. “KeyUsage”, “BasicConstraints” etc.

Blocking X.509 certificates

If a certificate becomes untrustworthy before the validity period has lapsed, then it must be blocked immediately. This can become necessary if, for example, the private key has become public knowledge. This applies in particular 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 (or, more precisely, the responsible CA) must make this information available to all those involved. The instruments currently used for this are the *Certificate Revocation Lists* (CRLs), which are supplied by the CA to public CRL distribution points (CDPs) at regular intervals. As an option, the CDP can also be named as an extension in the certificate, so that the checker can fetch a current CRL from there for validation purposes. One way to do this is the *Online Certificate Status Protocol* (OCSP). The authenticity of the CRLs is ensured by means of the signature of the issuing CA. The 26.2 table shows the principle underlying an X.509-CRL.

Table 26.2: *X.509 certificate revocation list(CRL)*

Field	**Content**
Version	The version of the CRL, e.g. 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.
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.

Repository for certificates and CRLs

In order to be used, the certificates and CRLs for a CA must be made publicly accessible. This involves a so-called *repository*. Because the certificates and CRLs cannot be forged, thanks to the signature, the repository itself does not need to be secured in a special way. On the contrary, the aim should be to achieve the simplest and fastest access possible. For this reason, certificates are often provided by means of an LDAP or HTTP server. You will find explanations about this in 21.8 on page 483. 22 on page 541 contains information about the HTTP server.

Proprietary PKI

YaST contains modules for the elementary management of X.509 certificates. This mainly involves the creation of CAs, sub-CAs and their certificates. At this point it should be noted that the services of a PKI go far beyond simply creating and distributing certificates and CRLs. The operation of a PKI is a service that also requires a well-conceived administrative infrastructure. The continuous updating of certificates and CRLs requires very complex management, which is provided by commercial PKI products and can also be partly automated. YaST functionality for creating and distributing CAs and certificates cannot provide this background information at present. In general, the PKI products currently available under OpenSource are subject to the commercial versions. Thus, if you wish to set up a "small" PKI, you can do this with the YaST modules described below. However, you should use commercial products to set up an "official" — or even commercial — PKI.

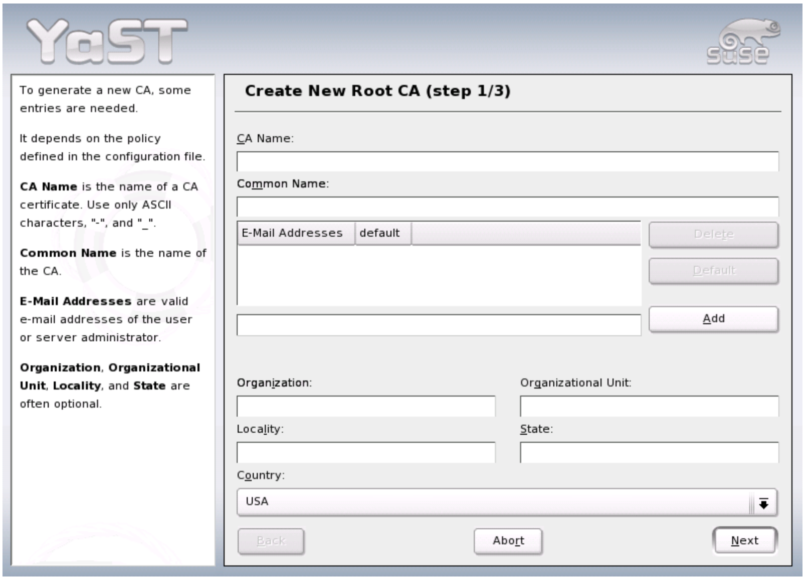
26.1.2 YaST modules for CA management

YaST provides two modules for elementary CA management. The functionality of these two modules is explained below on the basis of the key activities when administering CAs.

Creating a root CA

The first step when setting up a PKI is to create a root CA. This is achieved using the 'CA Management' YaST module in the YaST control center under the 'Security and Users' heading. After the module has been started you will first see a list of all existing CAs. The 'Create Root CA' button takes you to the first of three dialogs for entering CA-related data. You can enter the basic data for the CA in the first dialog (Fig. 26.1 on the next page). The 'Common Name' is the "mnemonic" name of the CA, in other words, the name used to refer to the CA. In contrast, the 'CA Name' field indicates the technical name of the CA. Directory names, among other things, are derived from this name, which is why only the characters specified in the Help can be used. The technical name is also displayed in the overview list when the module is started. Several e-mail addresses can be entered which can be seen by the CA user. This can be helpful for inquiries. Select the country where the CA is to be operated in 'Country'.

After you click on 'Next' you must enter a password in the second dialog. This password is always required when you wish to use the CA, i.e. when


YaST
SUSE
Create New Root CA (step 1/3)
CA Name:
Common Name:
E-Mail Addresses
default
Delete
Default
Add
Organization:
Organizational Unit:
Locality:
State:
Back
Abort
Next

To generate a new CA, some entries are needed.

It depends on the policy defined in the configuration file.

CA Name is the name of a CA certificate. Use only ASCII characters, "-", and "_".

Common Name is the name of the CA.

E-Mail Addresses are valid e-mail addresses of the user or server administrator.

Organization, **Organizational Unit**, **Locality**, and **State** are often optional.

Country: USA

Figure 26.1: *YaST CA module - basic data for a root CA*

you want to create a sub-CA or generate certificates. The 'Key Length' already contains a meaningful default and does not generally need to be changed unless an application cannot deal with this key length. The 'validity period' in the case of a CA is 3650 days, in other words roughly 10 years. This long period makes sense because the replacement of a deleted CA involves an enormous administrative effort. Clicking on the 'Extended' button takes you to a dialog for setting different attributes from the X.509 extensions (Fig. 26.4 on page 643). These values have rational default settings and should only be changed if you are really sure of what you are doing.

In the third and last step, YaST displays the current settings for checking purposes. If you click on 'Create', the root CA will be created and will then appear in the overview list.

Note

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 CA and to create the user certificates from there. This has the advantage that the root CA can be kept isolated and secure, e.g. on an isolated computer on secure premises. This makes it very difficult to attack the root CA.

Note

Creating or revoking a sub-CA

A sub-CA is created in exactly the same way as a root CA, except that it is first necessary to select the CA in which the sub-CA is to be created. After the program starts, select the required CA from the list and click 'Enter CA'. The first time you enter a CA after the program is started, you must key in the password, after which you will be taken to a dialog in which the key CA information is displayed (Fig. 26.2 on the next page). Click on the 'Extended...' button and select 'Create sub-CA'. This will take you to the same dialog as for creating a root CA.

Note

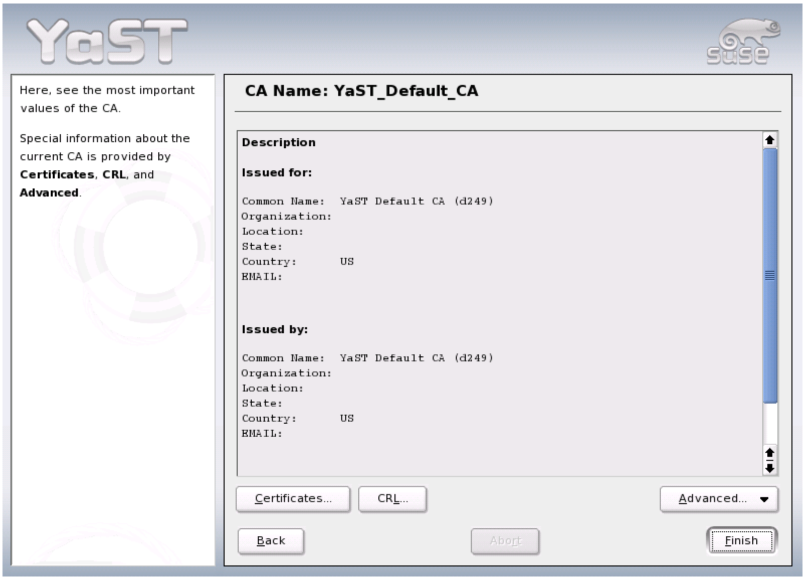
The validity period for a sub-CA must be fully within the validity period of the "parent" CA. Because a sub-CA is always created **after** the "parent" CA, the uncorrected standard value leads to an error message. To avoid this, enter a permissible value for the period of validity.

Note

After you have selected 'Certificates' you will see the dialog for administering CA certificates and sub-CAs. You can reset compromised or otherwise unwanted sub-CAs here using the 'Revoke' button. The revocation is not enough to deactivate a sub-CA on its own. You must also publish revoked sub-CAs in a CRL. The creation of CRLs is described in 26.1.2 on page 644.

Creating or revoking user certificates

In order to create client and server certificates, it is first necessary to enter a CA, as described in 26.1.2. User certificates should only be created in sub-CAs because then the root CA can be kept separate and secure. After you click on 'Certificates...' you will see the dialog for administering certificates (Fig. 26.3 on page 642). The upper part contains a list of existing certificates, while the data for the currently selected certificate appears below.



Here, see the most important values of the CA.

Special information about the current CA is provided by **Certificates**, **CRL**, and **Advanced**.

CA Name: YaST_Default_CA

Description

Issued for:

```
Common Name:  YaST Default CA (d249)
Organization:
Location:
State:
Country:      US
EMAIL:
```

Issued by:

```
Common Name:  YaST Default CA (d249)
Organization:
Location:
State:
Country:      US
EMAIL:
```

Figure 26.2: YaST CA module - using CA

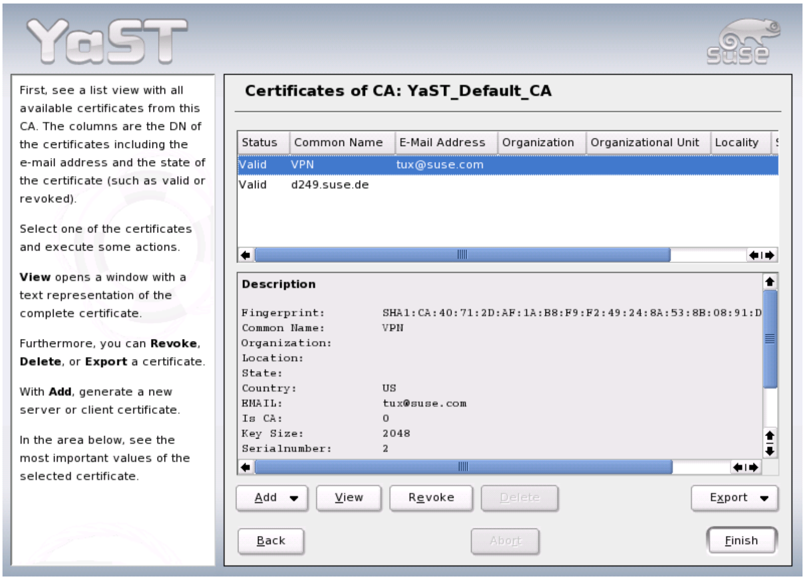
The 'Add' button allows you to create new client and server certificates and to add them to the list of CAs. The dialog for recording data is very similar to the one for creating the CAs and the same principles apply. Additional remarks relate to the e-mail addresses in certificates intended for e-mail signature and encryption. The e-mail address of the sender (in other words of the private key owner) should be contained in the certificate for the signature to enable the e-mail address to assign the correct certificate. For certificate assignment during encryption it is necessary for the e-mail address of the recipient (i.e. the public key owner) to be included in the certificate. In addition, in the case of server certificates, the host name of the server must be entered in the 'Common Name' field. The default validity period for certificates is 365 days.

Note

If certificates for IPsec applications are to be created with Windows-XP, then **client** certificates must be used. There, the "KeyUsage" extension contains the values expected by Windows.

Note

The ‘Revoke’ button also enables you to withdraw compromised or otherwise unwanted certificates. However, the revocation alone is not enough to deactivate a certificate. You must also publish revoked certificates in a CRL. 26.1.2 on page 644 explains how to create CRLs. Revoked certificates can be completely removed **after** publication in a CRL with the ‘Delete’ button.



First, see a list view with all available certificates from this CA. The columns are the DN of the certificates including the e-mail address and the state of the certificate (such as valid or revoked).

Select one of the certificates and execute some actions.

View opens a window with a text representation of the complete certificate.

Furthermore, you can **Revoke**, **Delete**, or **Export** a certificate.

With **Add**, generate a new server or client certificate.

In the area below, see the most important values of the selected certificate.

Certificates of CA: YaST_Default_CA

Status	Common Name	E-Mail Address	Organization	Organizational Unit	Locality
Valid	VPN	tux@suse.com			
Valid	d249.suse.de				

Description

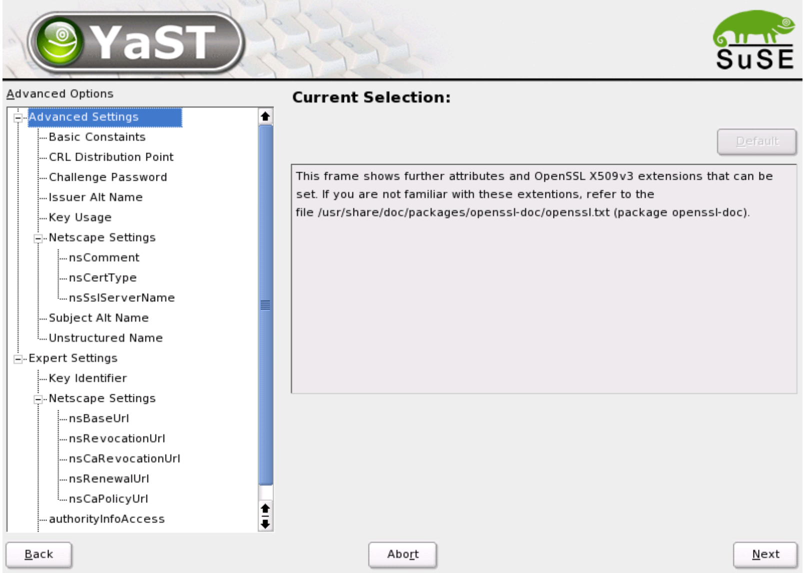
```
Fingerprint:        SHA1:CA:40:71:2D:AF:1A:B8:F9:F2:49:24:8A:53:8B:08:91:D
Common Name:        VPN
Organization:
Location:
State:
Country:            US
EMAIL:              tux@suse.com
Is CA:              0
Key Size:           2048
Serialnumber:       2
```

Figure 26.3: Certificates of a CA

Changing standard 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 a particular application has special requirements in relation to the content of these extensions. If you frequently wish to create certificates for an application of this kind, it may make sense to adjust the defaults. Otherwise you would have to start from scratch every time you create a certificate.

The system manages a set of defaults for every CA for the creation of sub-CAs, client certificates and server certificates. In order to change these defaults, you must enter the required CA, as described in 26.1.2 on page 640. After you click the 'Extended' button, this is where you will find the 'Defaults' option, where you can choose for which type the settings are to be changed. After this you will reach the dialog for changing the defaults (Fig. 26.4).


YaST
SuSE
Advanced Options
Current Selection:
Default
Advanced Settings
Basic Constaints
CRL Distribution Point
Challenge Password
Issuer Alt Name
Key Usage
Netscape Settings
nsComment
nsCertType
nsSslServerName
Subject Alt Name
Unstructured Name
Expert Settings
Key Identifier
Netscape Settings
nsBaseUrl
nsRevocationUrl
nsCaRevocationUrl
nsRenewalUrl
nsCaPolicyUrl
authorityInfoAccess
Back
Abort
Next

This frame shows further attributes and OpenSSL X509v3 extensions that can be set. If you are not familiar with these extentions, refer to the file /usr/share/doc/packages/openssl-doc/openssl.txt (package openssl-doc).

Figure 26.4: *YaST CA module - extended settings*

The tree structure and all extensions known to the system are displayed on the left. If you click on a field here, you can change the associated value on the right hand side and set or delete the "critical" marking with the 'critical' checkbox. After you click on 'Next' you will see a short summary and you can save your changes with 'Save'.

Note

All changes to the defaults only affect objects created after this point. Already existing CAs and certificates remain unchanged.

Note

Creating CRLs

If compromised or otherwise unwanted certificates are to be excluded from further use, they must first be revoked. The procedure for this has already been explained in 26.1.2 on page 640 (for sub-CAs) and 26.1.2 on page 640 (for user certificates). After this, a CRL **must** be created and published with this information.

The system administers precisely one CRL for every CA. In order to create or update this CRL, first enter the required CA, as already described in 26.1.2 on page 640, and click on the 'CRL...' button. The following dialog then displays a summary of the last CRL of this CA. If you have revoked new sub-CAs or certificates in the interim, you must create a new CRL so that this information can be added to the CRL. To create or update the CRL, select 'Create CRL'. You can then specify the period of validity for the new CRL (default: 30 days) and after you click on 'OK' the CRL will be created and displayed. Afterwards you must publish this CRL. This is explained below.

Note

Applications that evaluate CRLs will reject certificates whose CRL is deleted. As a PKI provider, it is your duty always to create and publish a new CRL before a current CRL lapses (period of validity). YaST does not provide a function for automating this procedure at present. You must therefore monitor this yourself.

Note

Exporting CA objects to LDAP

The executing computer should be configured with the YaST LDAPclient for LDAP export because then LDAP server information is available at runtime that can be used when completing dialog fields. Otherwise, although export may be possible, all LDAP data would have to be entered manually. You must always enter several passwords (see table 26.3 on the next page).

Table 26.3: Passwords during LDAP export

Password	Meaning
LDAP password	This password authorizes the user to make entries in the LDAP tree
Certificate password	This 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 CA to LDAP**

  To export a CA, enter the CA as described in 26.1.2 on page 640. Select 'Extended' → 'Export To LDAP' in the subsequent dialog, which will take you to the dialog for entering LDAP data. If your system has been configured with the YaST LDAP client, then the fields are already partly completed, otherwise you will have to 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 you want to export and 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. A corresponding user object is then sought in the LDAP tree and certificate is saved there with the attributes "userCertificate" (PEM format) and "userPKCS12" (PKCS12 format).

- **Exporting CRL to LDAP**

  Enter the CA containing the CRL you want to export and select 'CRL...'. If you wish, you can then create a new CRL and then export this with 'Export' → 'To LDAP'. The LDAP data is also entered here in the same way as with CAs. Entries are then made in the LDAP at the same point as the associated CA, but using the "certificateRevocationList" attribute.

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 to choose from, such as PEM, DER and PKCS12. In the case of PEM it is also possible to choose whether a certificate is to be exported with or without key and whether the key is to be encrypted. In the case of PKCS12, it is also possible to export the certification path.

Export in file format is performed for certificates, CAs and CRLs in just the same way as described for LDAP in 26.1.2 on page 644. You just need to select 'Export As File' instead of 'Export to LDAP'. This will then take you to a dialog for selecting the required output format and for entering the password and file name. The certificate is stored at the required location after you click 'OK'.

Note

You can select any storage location in the file system. Thus, this option can also be used to save CA objects on a USB stack as transport medium for example.

Note

Exporting certificates on floppy

YaST also allows certificates (but not CAs or CRLs) to be exported to floppy. The point of this option is the convenient transport of server certificates from an isolated CA computer to a server which is to use these certificates. This YaST function is the counterpart of a special YaST module which only serves to import certificates exported in this way onto the server (see the next section).

For floppy export, first enter the CA containing the certificates you want to export and select 'Certificates'. Select the required certificate in the list and export it with 'Export' → 'Export To Floppy'. The next dialog will ask you to insert a floppy and you can enter the new PKCS12 password. After you click on 'Next' the certificate will be written onto the floppy.

Importing general server certificates

If you have exported a server certificate to floppy on an isolated CA management computer with YaST, then you can re-import this certificate on a server as a "general server certificate". This can be performed either during installation or at a later point with the YaST module 'Import

General Server Certificate' in the YaST control center under the heading 'Security and Users'. The "general server certificate" is stored in `/etc/ssl/servercerts` and can be used there by any CA-supported service. When this certificate lapses, it can easily be replaced using the same mechanisms. The only remaining administrative effort required is the restart of the participating services.

After the module has been started, you will see the data for the current certificate in the description field. For import, select 'Import' → 'From Floppy' and insert the appropriate floppy. After you have entered the certificate password and clicked 'Next' the certificate will be imported and then displayed in the description field.

Note

If you select 'Import' → 'From Hard Disk' here, you can select the source in the file system. This option can thus also be used to import certificates from a USB stick as transport medium, for example.

Note

26.2 VPN with SUSE LINUX

VPN (*Virtual Private Network*) refers to a technology which is used to implement "secure" data connections via the "non-secure" medium of the Internet. Communication is not **with** the Internet, but **via** the Internet. The data packages are encrypted here for authentication and confidentiality and are packed into a new package (tunneling). This is an economical way to produce a "secure" network between geographically far-flung computers. The standard for this kind of data traffic is IPSEC (*Internet Protocol Security)*, which is implemented under Linux (among others) by means of the FreeS/WAN program.

The establishment of a VPN connection requires the availability of digital certificates from all participating parties which are used to verify the "legality" of the connection. Such certificates can be created with YaST and then used for VPN. 26.1 on page 634 contains a brief explanation of the background of digital certification and outlines how you can create and manage certificates yourself. The next sections explain how you can set up a VPN server and VPN clients under Linux and Windows using YaST.

26.2.1 Setting up Road Warrior servers

A Road Warrior server is a VPN server configuration that accepts connections from any clients with valid and signed CA certificates. Three steps are required in order to set up a Road Warrior server and these are explained below:

- Create a server certificate on the CA management computer
- Import a certificate on the server computer
- Set up a connection on the server.

Creating server certificates

The server certificate is created with the YaST CA Management module (see 26.1.2 on page 640). You should then save the certificate together with the key and all participating CAs in a PKCS12 file (see 26.1.2 on page 646).

Note

If certificates are to be created for IPsec applications with Windows-XP, then **client** certificates must be used for this purpose. The "KeyUsage" extension there contains the values expected by Windows.

Note

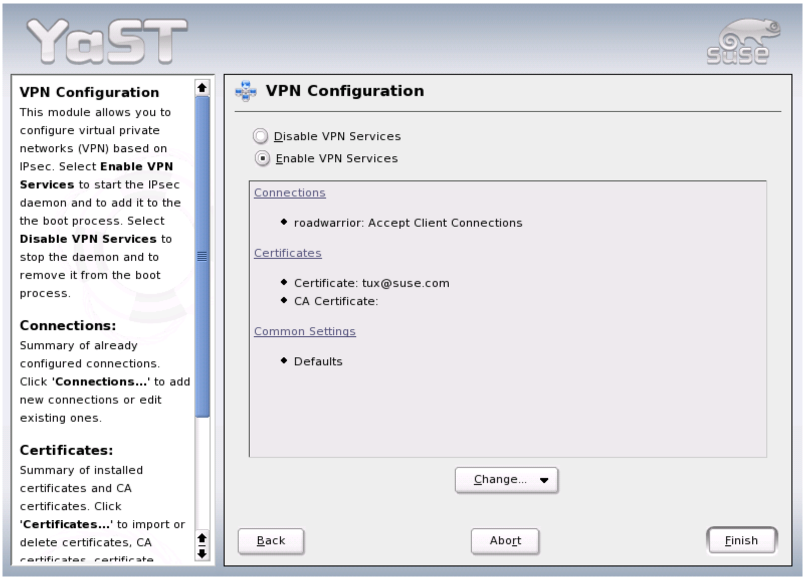
Importing a server certificate on the server

Start the 'VPN' YaST module on the server in the YaST control center under the 'Security and users' heading. In the overview (Fig. 26.5 on the next page) click 'Certificates' → 'Import' and then select your saved PKCS12 file. You must enter the PKCS12 password for the import. After this, the certificate will be displayed in the certificate list and clicking on 'Next' will take you back to the overview.

Note

You should not use the "general server certificate" of the YaST CA Management module here because IPsec manages its own certificates.

Note



VPN Configuration

This module allows you to configure virtual private networks (VPN) based on IPsec. Select **Enable VPN Services** to start the IPsec daemon and to add it to the the boot process. Select **Disable VPN Services** to stop the daemon and to remove it from the boot process.

Connections:

Summary of already configured connections. Click '**Connections...**' to add new connections or edit existing ones.

Certificates:

Summary of installed certificates and CA certificates. Click '**Certificates...**' to import or delete certificates, CA

VPN Configuration

○ Disable VPN Services
● Enable VPN Services

Connections

- roadwarrior: Accept Client Connections

Certificates

- Certificate: tux@suse.com
- CA Certificate:

Common Settings

- Defaults

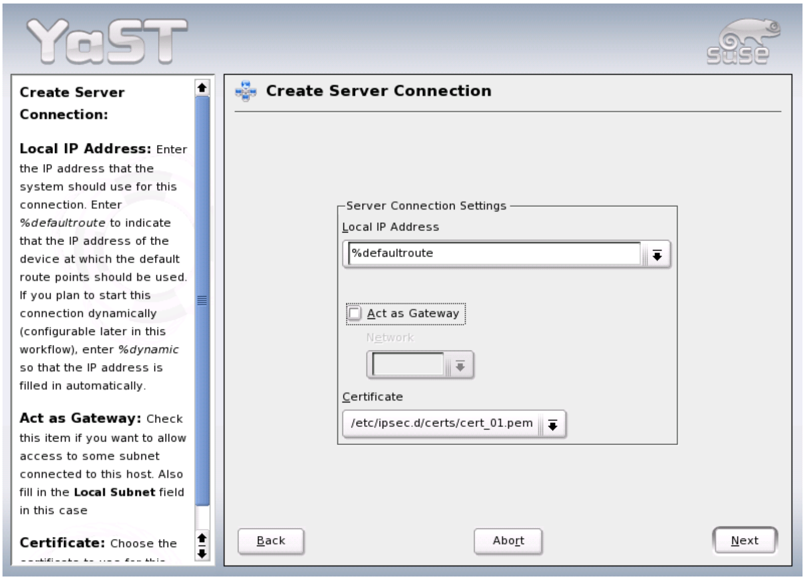
Figure 26.5: *YaST VPN module - overview*

Setting up a VPN connection

Another connection must be set up to ensure that the certificate can be used for IPsec. In the overview (Fig. 26.5) click 'Connections' and then select 'Add' in the connection overview. After you have selected 'Road Warrior Server' a configuration will be created that accepts connections from any client if it has a valid certificate signed by the CA.

The connection settings are selected in the next dialog (Fig. 26.6 on the next page). Enter your own IP address in the 'Local IP Address' field. In the case of Internet dial-up access, this is not usually known prior to the dial-up. However, in the case of Internet access, there is usually a default route. The `%defaultroute` setting instructs the server to use the interface to which the default route points. If the connection is to be set up and cleared dynamically when a network interface without default route is activated and deactivated, then `%dynamic` can be entered instead. The IP addresses of the relevant interface are then used.

If the server is to act as a gateway and clients permit access to a network, then the 'Function As Gateway' field should be checked off. You must then enter this network in the input field, e.g. `10.10.0.0/24`. You can also se-



Create Server Connection:

Local IP Address: Enter the IP address that the system should use for this connection. Enter *%defaultroute* to indicate that the IP address of the device at which the default route points should be used. If you plan to start this connection dynamically (configurable later in this workflow), enter *%dynamic* so that the IP address is filled in automatically.

Act as Gateway: Check this item if you want to allow access to some subnet connected to this host. Also fill in the **Local Subnet** field in this case

Certificate: Choose the

Create Server Connection

Server Connection Settings

Local IP Address: %defaultroute

☐ Act as Gateway

Network:

Certificate: /etc/ipsec.d/certs/cert_01.pem

Figure 26.6: *YaST VPN module - connection settings*

lect the required certificate here. The first, and perhaps only, certificate is already preselected.

Note

Either the first `Subject Alternative Name` (if any) or the `Distinguished Name` from the certificate is used in this simplified Road Warrior configuration workflow.

Note

After you click 'Next' you can choose how the connection is to be handled at system startup in the next dialog. A connection can either be "prepared" or "ignored". In the case of a prepared connection, the server waits for connection inquiries from clients. This is only possible if the local IP address is already known. This means, in the case of `%defaultroute`, that the default route must already be set and the computer must already have dialed into the Internet (if the line is a dialup connection). If the connection to the Internet is not already established when the system is started, you can also specify that the connection is to be set up and cleared dynamically with a network interface, e.g. with a DSL interface. If the interface selected here is not the interface of the default route, it makes no sense to enter `%defaultroute` as the local IP address either. When you click once on 'OK' you will see the new connection in the connection overview list. Click 'OK' again and then 'Exit' to conclude the configuration.

26.2.2 Setting up a VPN Linux client with FreeS/WAN

Three key steps are required in order to set up a VPN Linux client. These are explained below:

- Create a client certificate on the CA administration computer
- Export a FreeS/WAN configuration file
- Import files on the client computer

Creating a client certificate

The client certificate is created with the YaST CA Management module (see 26.1.2 on page 640). The finished certificate is then saved together with the key and all participating CAs in a PKCS12 file (see 26.1.2 on page 646).

Exporting a FreeS/WAN configuration file

On the server, start the 'VPN' YaST module in the YaST control center under the heading 'Security and Users'. In the overview (Fig. 26.5 on page 649) click 'Connections' and then select the required server connection in the connection overview. After you select 'Experts...' → 'Export' → 'FreeS/WAN' you can decide on the storage location for the `freeswan_ipsec.conf` file, which must be transferred to the Linux client. This file is a suggestion for a FreeS/WAN client and its details may need to be adapted. The file is tailored to FreeS/WAN Version 2. Older versions require additional parameters.

Importing files on the client

Next, the certificates and the configuration file is to be transferred to the client by means of a secure medium. The IPsec configuration file must be saved on the client as `/etc/ipsec.conf`.

To import the certificate, start the 'VPN' YaST module in the YaST control center under the heading 'Security and Users' on the client. In the overview (Fig. 26.5 on page 649) click 'Certificates' → 'Import' and then select your saved client certificate. For import purposes you must enter the password for the certificate. The certificate is then displayed in the certificate list and clicking 'Next' takes you back to the overview.

Note

The connection may need to be adapted to local circumstances (e.g. change certificate and ID).

Note

Manual client configuration

If the client computer does not have a YaST VPN module, then you can import the certificates manually:

- The client certificate must be copied to `/etc/ipsec.d/certs`.
- The CA certificate must be copied to `/etc/ipsec.d/cacerts`.
- The key must be copied to `/etc/ipsec.d/private`. Only the `root` user should have access to this file, i.e. the authorizations must be adapted accordingly.
- The password for the key must be entered in `/etc/ipsec.secrets`. This file should also only be accessible from `root`.

The `openssl` command line program can be used to extract the certificate from the PKCS#12 file:

```
# openssl pkcs12 -clcerts -nokeys -in DATEI.p12 -out \
          /etc/ipsec.d/certs/cert_01.pem
```

The same applies to the CA certificate:

```
# openssl pkcs12 -cacerts -nokeys -in DATEI.p12 -out \
          /etc/ipsec.d/cacerts/cacert_01.pem
```

and also to the keys:

```
# openssl pkcs12 -nocerts -nodes -in USER.p12 -out \
          /etc/ipsec.d/private/key_01.pem

# chmod 600 /etc/ipsec.d/private/key_01.pem
```

The `-nodes` option ensures that the key is stored without a password. That is no harm in this case because the file can only be read by `root` in any case. Another entry is required in `/etc/ipsec.secrets` so that FreeS/WAN will recognize the key:

```
# echo ': RSA /etc/ipsec.d/private/key_01.pem ""' \
         >> /etc/ipsec.secrets

# chmod 600 /etc/ipsec.secrets
```

The configuration file can now be copied to `/etc/ipsec.conf`. Under certain circumstances the file name at `leftcert` may need to be adapted. However, `/etc/ipsec.d/certs/cert_01.pem` is normally already entered. The value following `right` must be identical with the DNS host name or the IP address of the server.

The `rcipsec start` command starts the IPsec and establishes the connection (if `auto=start` has been configured). `ipsec auto --status` or `setkey-D` and an inspection of `/var/log/messages` enable you to check that everything has worked. `rcipsec stop` ends IPsec and all connections are cleared.

26.2.3 IPsec clients on Windows XP and Windows 2000

You can also set up IPsec connections to SUSE LINUX from Windows XP and Windows 2000 clients. The various steps are described below.

- Create client certificate on the CA management computer
- Export the Windows configuration file
- Prepare Windows
- Configure the Windows snap-ins
- Import a client certificate
- Make a note of important certificate data
- Configure the IPsec connection
- Create desktop links

Creating a client certificate

The client certificate is created using the YaST CA Management module (see 26.1.2 on page 640). The completed certificate should then be saved together with the key and all associated CAs in a PKCS12 file (see 26.1.2 on page 646).

Exporting a Windows configuration file

On the server, start the 'VPN' YaST module in the YaST control center under the heading 'Security and Users'. In the overview (Fig. 26.5 on page 649) click 'Connections' and then select the required server connection in the connection overview. After you select 'Experts...' → 'Export' → 'Windows' you can decide on the storage location for the `windows_ipsec.conf` file, which must be transferred to the Windows client. This file is a suggestion for a Windows client and its details may need to be adapted.

Preparing Windows

You can set up the IPsec connection manually, which requires `ipseccmd.exe` (Windows XP) or `ipsecpol.exe` (Windows 2000). These should be included in your Windows installation. In the case of Windows XP, you must execute `support\tools\setup.exe` on the installation CD (complete installation). However, these programs are command line-based, making them quite difficult to use. You can also configure the connection by means of the so-called MMC (*Microsoft Management Console*), however this is not particularly intuitive. Instead, we recommend using the `ipsec.exe` tool, which does the main work of configuring the IPsec connection under Windows XP or Windows 2000 for you.

Download this tool onto your computer under URL `http://vpn.ebootis.de/package.zip` and decompress the contents, for example under `C:\Programs\IPsec\`. At this point we would like to thank the author, `marcus@ebootis.de`.

If you use Windows 2000, you should first load at least ServicePack2, so that Windows 2000 can also handle the so-called 3DES encryption. Otherwise a connection to Windows 2000 cannot be made. ServicePack2 is available under the following URL:

`http://www.microsoft.com/windows2000/downloads/servicepacks/sp2/sp2lang.asp`

In the case of Windows 2000 you also need `ipsecpol.exe`, which can be found in the so-called resource kit under the following URL:

`http://agent.microsoft.com/windows2000/techinfo/reskit/tools/existing/ipsecpol-o.asp`

Note

This program normally installs to `C:/Programs/ResourceKit`. However, it is not much use at this point because it is a command line-based program and therefore needs to be copied into a directory in which executable files are stored. We recommend copying `ipsecpol.exe` to `C:/WINNT` and the corresponding DLLs to `C:/WINNT/System`. `ipsecpol` must be executed as administrator.

Note

Configuring the required snap-ins

Open the MMC on the Windows client. In the start menu, go to 'Run' → 'MMC'. In the MMC click 'File' → 'Add/ Remove Snap-In'. A dialog opens up in which you may see active snap-ins. Click on 'Add'. A selection window opens to display all available snap-ins. 'Certificates' → 'Add' takes you to the configuration wizard. Here you should select 'Computer Account' and click on 'Next'. Select 'Local Computer' → 'Finish' and then the 'IP Security Guidelines Management' → 'Add' snap-in. A configuration wizard opens up in which you should select 'Local Computer' → 'Finish'. Click 'Close' and then 'OK'.

Importing a client certificate

The two snap-ins that have been added can be seen in the MMC. Open the 'Certificates' directory. Right click on 'Own Certificates'. In the dropdown menu, select 'All Tasks' → 'Import'. The certificate wizard opens. Select 'Next' → 'Search'. Under 'File Type' enter 'Private Information Exchange' (*.pfx,*.p12). Select the exported PKCS12 file and click 'Next'. You must now enter the password you have used in the YaST CA Management module to export the certificates. Click 'Next'. Now select 'Save All Certificates In Following Storage' → 'auto' and then 'Next' → 'Finish'. A dialog box will indicate if the import procedure has succeeded. Click 'OK'.

Taking note of important certificate data

The prepared IPsec sample configuration normally already contains the correct DN of the CA ('Issuer').

In the MMC now click 'File' → 'Save'. Save your configuration with the suggested name at the suggested location. To establish whether the certificate data is correct, open the 'Own Certificates' directory in the MMC again and open the 'Certificates' Right click on the certificate and select 'Open' from the dropdown menu and then the 'Details' tab. When you click on 'Issuer' you will see entries similar to those below, which you should note down:

```
        E=bsupport@suse.de
        CN=mainca
        OU=bu
        O=SuSE
        L=Nuremberg
        S=Franconia
        C=DE
```

Close the certificate view with 'OK' and the MMC with 'File' → 'Exit' → 'Save' → 'Yes'.

Configuring an IPsec connection

Install the `ipsec.exe` tool by decompressing `package.zip` to `C:\Programs\IPsec\`. In the next step, replace the standard version of the `ipsec.conf` file contained there with the exported `windows_ipsec.conf` from the VPN server (rename). Then change to the `C:\Programs\IPsec` directory and open the file with an editor to enable the configuration data to be checked. The following listing shows the standard values.

```
conn <Name of the connection>
        left=%any
        right=<IP of the SuSE Linux standard server>
        rightca=<the previously noted values in reverse order,
        separated by commas>
        network=auto
        auto=start
        pfs=yes
```

The first line must be left justified. All other lines must be indented. Here is a specific example for `ipsec.conf`:

```
conn me_to_servername
        left=%any
        right=10.10.254.181
        rightca="C=DE,S=Franconia,L=Nuremberg,O=SuSE,OU=bu,
                 CN=mainca,E=bsupport@suse.de"
        network=auto
        auto=start
        pfs=yes
```

Creating desktop links

Finally, the desktop link is created and the connection is activated. Create a link to the `C:\Programs\IPsec\IPSEC.exe` file on the desktop. Now establish the connection to the Internet and click on the first link. A window will open and the IPsec filters will be configured for your current connection. The best way to test the tunnel is with `ping <client IP behind the tunnel`. The message “Negotiating IP Security” appears once or twice, after which you will see the normal `ping` responses: the tunnel is active. In the case of Windows 2000, this takes two `ping` commands, so start `ping` again.

Closing a connection

To deactivate the IPsec filter and the tunnel, first call `IPSEC.exe -off` and then `IPSEC.exe -delete`. It is best to create a desktop link for this too.

26.3 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.

26.3.1 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 instance.

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).

The above-mentioned 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 26.7 on page 707 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.

26.3.2 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 21.1.2 on page 415) 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.

Note

Using the Correct Network Mask

When configuring your network, make sure both the broadcast address and the network mask are the same for all local hosts. Failing to do so results in a broken network because packets cannot be routed properly.

Note

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 still needs to be configured before it can forward such packets. For security reasons, SUSE LINUX does not enable this 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 packets coming back in reply. The router must identify these incoming packets and translate their target addresses, so packets can be forwarded back 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). Netscape, as well as the standard FTP program and many others, use the PASV mode. This passive mode is much less problematic as far as packet filtering and masquerading is concerned.

26.3.3 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 instance, 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 will be 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. For more information and links,

read the Firewall HOWTO included in `howto`. If this package is installed, you can read the HOWTO with `less /usr/share/doc/howto/en/Firewall-HOWTO.gz`.

26.3.4 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 Network 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 Network 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 21.1.2 on page 415), 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 web or mail servers.

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 that are allowed. Note that the ruleset is only applied to packets originating from external hosts. Locally generated packets are not captured by the firewall.

The configuration can be performed with YaST (see Section 26.3.4 on page 663). It can also be made manually in the file `/etc/sysconfig/SuSEfirewall2`, which is well commented.

Manual Configuration

The following paragraphs provide step-by-step instructions for a successful configuration. Each configuration item is marked as to whether it is relevant to firewalling or masquerading. Aspects related to the DMZ (demilitarized zone) as mentioned in the configuration file are not covered here. They are applicable only to a more complex network infrastructure as found in larger organizations (corporate networks), which require extensive configuration as well as in-depth knowledge about the subject.

First, use the YaST runlevel editor 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 or DSL connection, enter `ppp0`. For an ISDN link, use `ippp0`. 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 IPs 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. It is more secure to have a proxy server between the hosts of the internal network and the Internet.

`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_INTERNAL` (firewall)
Set this to `yes` to protect your firewall host from attacks originating in your internal network. Services are only be available to the internal network if explicitly enabled. Also see `FW_SERVICES_INT_TCP` and `FW_SERVICES_INT_UDP`.

`FW_AUTOPROTECT_SERVICES` (firewall)
Normally, set this to `yes` to enable the automatic generation of explicit rules for running services.

`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 name server and want to make it available to the outside. 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_INTERNAL` is set to `yes`.

`FW_SERVICES_INT_UDP` (firewall)
See above.

`FW_STOP_KEEP_ROUTING_STATE` (firewall)
Insert `yes` if you have configured your dial-up procedure to work automatically via diald or ISDN (dial-on-demand).

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)
```

Configuration with YaST

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 four sections:

Basic Settings Specify the interfaces to protect. To protect an individual host to which no internal network is connected, just specify the interface facing the Internet. If an internal network is connected to your system, the interface facing the network must also be specified. Exit this dialog with 'Next'.

Services You only need this option to use your system to offer services accessible from the Internet (web server, mail server, etc.). Activate the respective check boxes or use 'Expert...' to enable services by way of their port numbers (listed in `/etc/services`). If you are not going to use your host as a server, press 'Next' to exit this dialog without making any changes.

Features Here, select the main features of your firewall:

'Forward Traffic and Do Masquerading' Protects hosts in the internal network from the Internet — all Internet services appear to be used by your firewall, while the internal hosts remain invisible.

'Protect from Internal Network' If enabled, *internal* hosts can only use the services explicitly made available to them. Given that services cannot be made available from these dialogs, you should disable this option if you want internal hosts to access the firewall.

'Protect All Running Services' Enable this to completely deny access to the TCP and UDP services of the firewall from the outside. This does not affect the services explicitly made available in the preceding step.

'Allow traceroute' This assists in checking the routing to your firewall.

'Treat IPsec Data Traffic as Internal' This tells the firewall to deal with successfully decrypted IPsec packets as if they were packets coming from the internal network.

When completed the feature configuration, exit this dialog with 'Next'.

Logging Determine the scope of logging for your firewall. Before activating the 'Logging options', consider that these log files produce a large amount of output. The configuration of the logging function is the final step of the firewall configuration. Exit the dialog with 'Next' and confirm the following message to activate the firewall.

26.3.5 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.

26.4 SSH — Secure Shell, the Safe Alternative

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 is OpenSSH.

26.4.1 The OpenSSH Package

SUSE LINUX installs the package OpenSSH by default. The programs ssh, scp, and sftp are then available as alternatives to telnet, rlogin, rsh, rcp, and ftp.

26.4.2 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, you can 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 user name is different from the remote user name, 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"
tux@otherplanet’s 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 likewise executed on sun.

26.4.3 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 user name on earth is different from the user name 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. All output can be suppressed 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.

26.4.4 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 familiar commands as known from ftp. The sftp program may be a better choice than scp, especially when transferring data for which the file names are unknown beforehand.

26.4.5 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. A newly installed SUSE LINUX system defaults to version 2. If you want to keep using version 1 after an 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, as well as 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 over 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 instead 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.

Version 2 of the SSH protocol is used by default. Override this to use version 1 of the protocol with the `-1` switch. 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 backup 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 regarding this system must be removed from `~/.ssh/known_hosts`.

26.4.6 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. As 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 file name in which to store the keys:

```
Enter file in which to save the key (/home/tux/.ssh/id_rsa):
```

Confirm the default setting and answer the request for a passphrase. Even if the software suggests an empty passphrase, a text from ten to thirty 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`.

```
Enter same passphrase again:
Your identification has been saved in /home/tux/.ssh/id_rsa
Your public key has been saved in /home/tux/.ssh/id_rsa.pub.
The key fingerprint is:
79:c1:79:b2:e1:c8:20:c1:89:0f:99:94:a8:4e:da:e8 tux@sun
```

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`.

26.4.7 X, Authentication, and Other 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 be 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 system-wide 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`.

26.5 Network Authentication — Kerberos

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 must register once and is then 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 no one can take the identity of someone else.
- Make sure each network server also proves its identity. If you do not, 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 the above 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.

Note

The original Kerberos was designed at the MIT. Besides the MIT Kerberos, several other implementations of Kerberos exist. SUSE LINUX ships with a free implementation of Kerberos 5, the Heimdal Kerberos 5 from KTH. Because the following text covers features common to all versions, the program itself is referred to as Kerberos as long as no Heimdal-specific information is presented.

Note

26.5.1 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 user name 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*.

26.5.2 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 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.

First contact Your first contact with Kerberos is quite similar to any login procedure at a normal networking system. Enter your user name. 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 will generate 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.

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 him 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 will 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 can be ignored.

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.

Ticket-granting — getting into contact with 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.

Compatibility to Windows 2000 Windows 2000 contains a Microsoft implementation of Kerberos 5. As SUSE LINUX makes use of the Heimdal implementation of Kerberos 5, find useful information and guidance in the Heimdal documentation. See Section 26.5.4 on the next page.

26.5.3 Users' View of Kerberos

Ideally, a user's one and only contact with Kerberos happens during login at his workstation. The login process includes obtaining a ticket-granting ticket. At logout, a user's Kerberos tickets are automatically destroyed, which hinders anyone else from impersonating this user when not logged in. The automatic destruction 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 ten 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. Those interested in a list of all the tickets silently acquired for them by Kerberos should run klist.

Here is a short list of some applications that use Kerberos authentication. These applications can be found under `/usr/lib/heimdal/bin`. 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
- popper, push
- ftp, ftpd
- su

- imapd
- pine

You no longer have to type 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 the encrypted contents of the tickets are adjusted to the new situation. Simply copying tickets between workstations is not sufficient as the ticket contains workstation-specific information (the IP address). XDM and KDM offer Kerberos support, too. Read more about the Kerberos network applications in the *Kerberos V5 UNIX User's Guide* at `http://web.mit.edu/kerberos/www/krb5-1.3/krb5-1.3/doc/krb5-user.html`.

26.5.4 For More Information

SUSE LINUX contains a free implementation of Kerberos called Heimdal. Its documentation is installed along with the package `heimdal` under `/usr/share/doc/packages/heimdal/doc/heimdal.info`. It is also available at the project's home page at `http://www.pdc.kth.se/heimdal/`.

The official site of the MIT Kerberos is `http://web.mit.edu/kerberos/www/`. There, find links to any other relevant resource concerning Kerberos. A "classical" dialog pointing out the principles of Kerberos is available at `http://web.mit.edu/kerberos/www/dialogue.html`. It is a less technical but still a comprehensive read.

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.

These links provide a short introduction to Kerberos and answer many questions regarding Kerberos installation, configuration, and administration:

`http://web.mit.edu/kerberos/www/krb5-1.3/krb5-1.3/doc/krb5-user.html`,

`http://web.mit.edu/kerberos/www/krb5-1.3/krb5-1.3/doc/krb5-install.html`,

`http://web.mit.edu/kerberos/www/krb5-1.3/krb5-1.3/doc/krb5-admin.html`.

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.

26.6 Installing and Administering Kerberos

This section covers the installation of the Heimdal Kerberos implementation as well as some aspects of administration. This section assumes you are familiar with the basic concepts of Kerberos (see also Section 26.5 on page 671).

26.6.1 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. Setting up crossrealm authentication is described in [15], for instance. For the remainder of this section, the realm name `SAMPLE.COM` is used in all examples.

26.6.2 Setting up the KDC Hardware

The first thing required to use Kerberos is a machine that will act 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:

- Put the server machine into a physically secured location, such as a locked server room to which only a very few people have access.
- Do not run any network applications on it except the KDC. This includes servers and clients — for instance, the KDC should not import any file systems via NFS or use DHCP to retrieve its network configuration.
- It is probably a good approach to 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 X11-based. Even installing an SSH server should be considered a potential security risk.
- No graphical login is provided on this machine as an X server is a potential security risk. Kerberos provides its own administration interface.
- 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).

  Also consider disabling DNS lookups, because there is a potential risk involved. If there is a security bug in the DNS resolver library, an attacker might be able to trick the KDC into performing a DNS query that triggers this bug. To disable DNS lookups, simply remove `/etc/resolv.conf`.

- Disable all user accounts except root's account by editing `/etc/shadow` and replacing the hashed passwords with `*` or `!` characters.

26.6.3 Clock Synchronization

To use Kerberos successfully, make sure 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 will probably work 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. NTP configuration itself is beyond the scope of this section. For more information, refer to the NTP documentation included in your installed system under `/usr/share/doc/packages/xntp-doc/`.

It is also possible to adjust the maximum deviation Kerberos allows when checking time stamps. This value (called "clock skew") can be set via the `krb5.conf` file as described in Section 26.6.6 on page 686.

26.6.4 Log Configuration

By default, the Kerberos daemons running on the KDC host log information to the syslog daemon. To keep an eye on what your KDC is doing, process these log files regularly, scanning for unusual events or potential problems. Either do this by running a log scanner script on the KDC host itself or by copying these files from the KDC to another host with `rsync`.

Forwarding all log output via syslogd's log forwarding mechanisms is not recommended, because information traverses the network unencrypted.

26.6.5 Installing the KDC

This section covers the initial installation of the KDC, including creation of an administrative principal.

Installing the RPMs

Before you can start, install the Kerberos software. On the KDC, install the packages `heimdal`, `heimdal-lib`, and `heimdal-tools` with `rpm -ivh heimdal-*.rpm heimdal-lib-*.rpm heimdal-tools*.rpm`.

Setting the Master Key

Your next step is to initialize the database where Kerberos keeps all information about principals. First, set 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 type in the password every time the KDC is restarted. Make sure 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/heimdal/heimdal.db`), do not back up the stash file (which is in `/var/heimdal/m-key`). 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 will need it when restoring your database from backup tape after a crash.

To set the master key, run kstash without arguments and enter the pass phrase twice:

```
kstash

Master key:<enter pass phrase>
Verifying password - Master key:<enter pass phrase again>
```

Creating the Realm

Finally, create entries for your realm in the Kerberos database. Run `kadmin` with the -l option as shown. This option tells kadmin to access the database locally. By default, it tries to contact the Kerberos admin service over the network. At this stage, this will not work because it is not running yet.

Now, tell kadmin to initialize your realm. It will ask you a number of questions in return. It is best to accept the default settings offered by kadmin initially:

```
kadmin -l

kadmin> init SAMPLE.COM
Realm max ticket life [unlimited]: <press return>
Realm max renewable ticket life [unlimited]: <press return>
```

To verify that it did anything, use the list command:

```
kadmin> list *
default@SAMPLE.COM
kadmin/admin@SAMPLE.COM
kadmin/hprop@SAMPLE.COM
kadmin/changepw@SAMPLE.COM
krbtgt/SAMPLE.COM@SAMPLE.COM
changepw/kerberos@SAMPLE.COM
```

This shows that there are now a number of principals in the database. All of these are for internal use by Kerberos.

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 -l

kadmin> add newbie
Max ticket life [1 day]: <press return>
Max renewable life [1 week]: <press return>
Principal expiration time [never]: <press return>
Password expiration time [never]: <press return>
Attributes []: <press return>
newbie@SAMPLE.COM’s Password: <type password here>
Verifying password: <re-type password here>
```

Accepting the defaults by pressing Enter is okay. Choose a good password, however.

Next, create another principal named `newbie/admin` by typing `add newbie/admin` at the `kadmin` prompt. The `admin` suffixed to your user name 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.

Starting the KDC

Start the KDC daemons. This includes kdc itself (the daemon handling user authentication and ticket requests), kadmind (the server performing remote administration), and kpasswddd (handling user's password change requests). To start the daemon manually, enter `rckdc start`. Also make sure KDC is started by default when the server machine is rebooted with the command `insserv kdc`.

26.6.6 Configuring Kerberos Clients

When configuring Kerberos, there are basically two approaches you can take — static configuration via the `/etc/krb5.conf` file or dynamic configuration via DNS. With DNS configuration, Kerberos applications try to locate the KDC services via DNS records. With static configuration, add the host names 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, by 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 host names.

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.sample.com` is the host name of the KDC):

```
[libdefaults]
        default_realm = SAMPLE.COM

[realms]
        SAMPLE.COM = {
                kdc = kdc.sample.com
                kpasswd_server = kdc.sample.com
                admin_server = kdc.sample.com
        }
```

The `default_realm` line sets the default realm for Kerberos applications. If you have several realms, just add another statement to the `[realms]` section.

Also add a statement to this file that tells applications how to map host names to a realm. For instance, 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]
        .sample.com = SAMPLE.COM
        www.foobar.com = SAMPLE.COM
```

This tells the library that all hosts in the `sample.com` DNS domains are in the `SAMPLE.COM` Kerberos realm. In addition, one external host named `www.foobar.com` should also be considered a member of the `SAMPLE.COM` realm.

DNS-Based Configuration

DNS-based Kerberos configuration makes heavy use of SRV records (see *(RFC2052) A DNS RR for specifying the location of services* at `http://www.ietf.org`). These records are not supported in earlier implementations of the BIND name server. At least BIND version 8 is required for this.

The name of an SRV record, as far as Kerberos is concerned, is always in the format `_service._proto.realm`, where realm is the Kerberos realm. Domain names in DNS are case insensitive, so case-sensitive Kerberos realms would break when using this configuration method. `_service` is a service name (different names are used when trying to contact the KDC or the password service, for example). `_proto` can be either `_udp` or `_tcp`, but not all services support both protocols.

The data portion of SRV resource records consists of a priority value, a weight, a port number, and a host name. 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 will probably never need any of this, so it is okay to set these to zero.

Heimdal 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.SAMPLE.COM.  IN  SRV    0 0 88 kdc.sample.com.
_kerberos._tcp.SAMPLE.COM.  IN  SRV    0 0 88 kdc.sample.com.
```

_kpasswd This describes the location of the password changing server. Typical records look like this:

```
_kpasswd._udp.SAMPLE.COM.   IN  SRV    0 0 464 kdc.sample.com.
```

Because kpasswdd does not support TCP, there should be no `_tcp` record.

_kerberos-adm This describes the location of the remote administration service. Typical records look like this:

```
_kerberos-adm._tcp.SAMPLE.COM. IN  SRV    0 0 749 kdc.sample.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 `SAMPLE.COM` realm, even if it is not part of the sample.com DNS domain. This can be done by attaching a TXT record to `_keberos.hostname`, as shown here:

```
_keberos.www.foobar.com.  IN TXT "SAMPLE.COM"
```

Adjusting the Clock Skew

The *clock skew* is the tolerance for accepting tickets with time stamps that do not exactly match the host's system clock. Usually, the clock skew is set to 300 seconds, or 5 minutes. This means a ticket can have a time stamp somewhere between 5 minutes ago and 5 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
```

The YaST Kerberos Client

As an alternative to the manual configuration as described above, you can also use YaST to configure a Kerberos client. To do so, in the YaST Control Center select 'Network Services' → 'Kerberos Client'. When the dialog has opened, select the checkbox 'Use Kerberos'. If you want to set up a DNS based client, it is sufficient to just confirm the 'Basic Kerberos Settings' as displayed. If your domain does not support this kind of configuration, you need to provide the correct values for the 'Default Domain', the 'Default Realm' and the 'KDC Server Address' yourself. Selecting the button 'Advanced Settings' takes you to another YaST dialog, where you can tweak options related to tickets, OpenSSH support and time synchronization.

The dialog opened with 'Advanced Settings' includes all the settings related to ticket attributes. If you want to forward your complete identity in order to use your tickets on other hosts, select the checkbox labeled 'Tickets are Forwardable'. To enable the transfer of certain tickets only, select 'Tickets are Proxiable'. Tickets can be kept available by a PAM module even after a session has ended, by enabling 'Retain Tickets'. The 'Default Ticket Lifetime' can be specified 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 enable Kerberos authentication support for your OpenSSH client, select the corresponding checkbox. The client will then use Kerberos tickets to authenticate with the SSH server. You can exlude 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`). Lastly, you can 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...'. After finishing the configuration, YaST will perform all the necessary changes, and the Kerberos client is ready for use.

26.6.7 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/heimdal/kadmind.acl` (ACL is an abbreviation for access control list). The ACL 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              all
```

Replace the user name `newbie` with your own. Restart the KDC for the change to take effect.

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:

```
kinit newbie/admin

newbie/admin@SAMPLE.COM’s Password: <enter password>
/usr/sbin/kadmin
kadmin> privs
change-password, list, delete, modify, add, get
```

Using the privs command, verify which privileges you have. The list shown above is the full set of privileges.

As an example, modify the principal `newbie`:

```
kadmin> mod newbie

Max ticket life [1 day]:2 days
Max renewable life [1 week]:
Principal expiration time [never]:2005-01-01
Password expiration time [never]:
Attributes []:
```

This changes the maximum ticket life time to two days and sets the expiration date for the account to January 1, 2005..

Basic kadmin Commands

Here is a brief list of kadmin commands. For more information, refer to the manual page of `kadmin`.

add `principal` add a new principal

modify `principal` edit various attributes of a principal, such as maximum ticket life time and account expiration date

delete `principal` remove a principal from the database

rename `principal` `newname` renames a principal to `newname`

list `pattern` list all principals matching the given pattern. Patterns work much like the shell globbing patterns: `list newbie*` would list `newbie` and `newbie/admin` in our example.

get `principal` display detailed information about the principal

passwd `principal` changes a principal's password

At all stages, help is available by typing ? and Enter. This even works in prompt environments generated by `modify` and `add`.

The `init` command used when initially creating the realm (as well as a few others) is not available in remote mode. To create a new realm, go to the KDC's console and use kadmin in local mode (using the `-l` command line option). The same is true for dumping and restoring the KDC database using the `dump`, `load`, and `merge` commands.

26.6.8 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, Keberos-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 host name. 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. Obviously, 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 `machine.sample.com`, enter the following commands during your kadmin session:

```
kinit newbie/admin

newbie/admin@SAMPLE.COM’s Password: <type password>
kadmin add -r host/machine.sample.com
Max ticket life [1 day]:
Max renewable life [1 week]:
Principal expiration time [never]:
Password expiration time [never]:
Attributes []:
```

Instead of setting a password for the new principal, the -r 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:

```
ktutil get host/machine.sample.com
```

When completed, make sure you destroy the admin ticket obtained via kinit above with `kdestroy`.

26.6.9 Enabling PAM Support for Kerberos

SUSE LINUX 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_unix` module, too, supports Kerberos authentication and password updating. To enable Kerberos support in `pam_unix`, 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_unix` 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.

26.6.10 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 in SUSE LINUX 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
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 system-wide 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 you have a valid ticket then connect to the SSH server. To force SSH protocol version 1, specify option `-1` on the command line.

Note

Additional Information

The file `/usr/share/doc/packages/openssh/README.kerberos` discusses the interaction of OpenSSH and Kerberos in more detail.

Note

26.6.11 Using LDAP and Kerberos

When using Kerberos, one way to distribute the user information (such as user ID, groups, home directory, etc.) 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 used in SUSE LINUX 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 binding to the OpenLDAP server, create a principal `ldap/earth.sample.com` and add that to the keytab:

```
kadmin add -r ldap/earth.sample.com

ktutil get ldap/earth.sample.com
```

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 group `ldap`.

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
```

Neither solution is perfect. However, at the moment it is not possible to configure OpenLDAP to make it use a separate keytab file. Finally, restart the LDAP server using `rcldap restart`.

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=suse,dc=de '(uid=newbie)'

SASL/GSSAPI authentication started
SASL SSF: 56
SASL installing layers
[...]

# newbie, People, suse.de
dn: uid=newbie,ou=People,dc=suse,dc=de
uid: newbie
cn: Olaf Kirch
[...]
```

As you can see, ldapsearch prints a message that it started GSSAPI authentication. The next message is admittedly 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 of 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.

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=suse,dc=de`, 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=suse,dc=de" 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, which is how the LDAP server can find out that the Kerberos user `joe@SAMPLE.COM` corresponds to the LDAP distinguished name `uid=joe,ou=people,dc=suse,dc=de`. This sort of mapping must be configured manually using the `saslExpr` directive. In our example, add the following to `slapd.conf`:

```
saslRegexp
    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 `saslRegexp` 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 `saslRegexp` 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 user name is not part of the DN, you can even use search expressions to map the SASL DN to the user DN.

26.7 Security and Confidentiality

26.7.1 Basic Considerations

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 chapter 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.

26.7.2 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.

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.

Passwords

On a Linux system, passwords are, of course, 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 “tantalise” into “t@nt@1ls3”.

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.

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. 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. Prevent others from using such parameters during boot by setting an additional password in `/boot/grub/menu.lst` (see Chapter 8 on page 199).

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 the more than 200,000 files included in a SUSE 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 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's configuration programs to set permissions accordingly, select 'Security' in YaST. To learn more about the topic, read the comments in `/etc/permissions` or consult the manual page of `chmod` (`man chmod`).

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 them into memory areas that are too small to hold them. 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 that a program executes 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, in particular if the program is being executed with special privileges (see Section 26.7.2 on the preceding page).

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 26.7.2 on the facing page).

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.

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 our 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'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.

Network Security

Network security is important for protecting from an attack that is started outside. The typical login procedure requiring a user name 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.

X Window System and X11 Authentication

As mentioned at the beginning, network transparency is one of the central characteristics of a UNIX system. X11, the windowing system of UNIX operating systems, can make use of this feature in an impressive way. With X11, it is basically no problem to log in at a remote host and start a graphical program that is then be 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 (i.e., 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 Section 26.4 on page 665.

Caution

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.

Caution

Buffer Overflows and Format String Bugs

As discussed on Section *Buffer Overflows and Format String Bugs*, 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 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.

DoS — Denial of Service

The purpose of this kind of 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 done 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.

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 himself. 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 will 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, mostly 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.

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 host names. The attacker needs a good understanding of the actual structure of the trust relationships between 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.

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. Rather, 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.

26.7.3 Some General Security Tips and Tricks

Information: To handle security competently, it is important to keep up with new developments and to 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://www.suse.de/security`. The list `suse-security-announce@suse.de` 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 `suse-security@suse.de` is a good place to discuss any security issues of interest. Subscribe to it under the URL as given above for `suse-security-announce@suse.de`.

`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, sendmail, 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 will make 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 `tripwire`, available on the SUSE LINUX distribution. Encrypt the database created by `tripwire` 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 will work, 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_wrappers`, 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.

26.7.4 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.suse.de/security`.

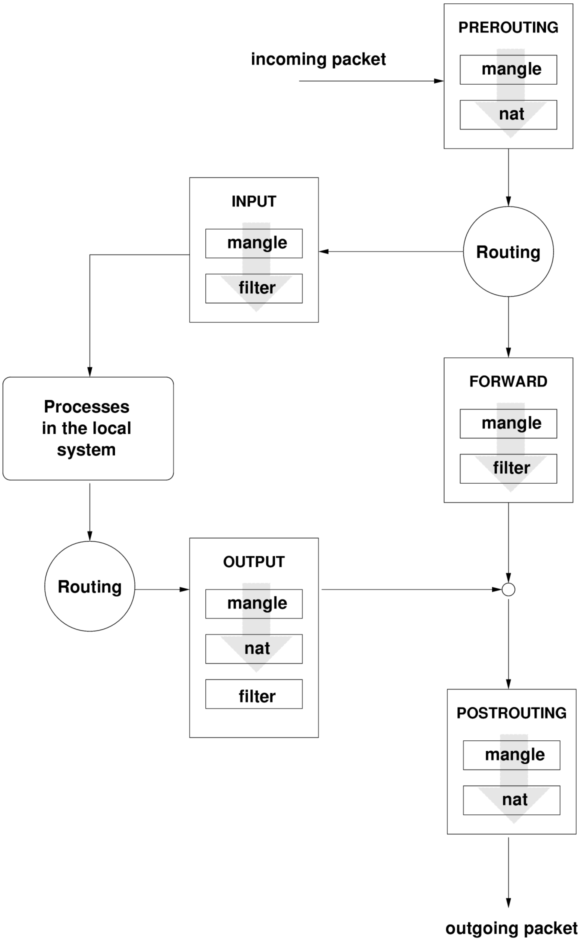

PREROUTING
incoming packet
mangle
nat
INPUT
mangle
filter
Routing
Processes
in the local
system
FORWARD
mangle
filter
Routing
OUTPUT
mangle
nat
filter
POSTROUTING
mangle
nat
outgoing packet

Figure 26.7: *iptables: A Packet's Possible Paths*

Part IV

Administration

27

Access Control Lists in Linux

This chapter provides a brief summary of the background and functions of POSIX ACLs for Linux file systems. Learn how the traditional permission concept for file system objects can be expanded with the help of ACLs (*Access Control Lists*) and which advantages this concept provides.

27.1 Advantages of ACLs 712
27.2 Definitions 713
27.3 Handling ACLs 714
27.4 Support by Applications 723

27.1 Advantages of ACLs

Note

POSIX ACLs

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/`.

Note

Traditionally, three sets of permissions 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. A more detailed discussion of this topic can be found in the *User Guide*.

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 for situations that require 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 clearly evident in situations like the replacement of a Windows server by 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.

Given than Samba supports 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, it is even possible to assign permissions to users that only exist in the Windows domain without any account on the Linux server. On the server side, edit the access control lists using getfacl and setfacl.

27.2 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`). An introduction to the user concept in Linux is provided in the *User Guide*.

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 (see Table 27.1 on the following page), 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.

27.3 Handling ACLs

This section explains the basic structure of an ACL and its various characteristics. The interrelation between ACLs and the traditional permission concept in the Linux file system is briefly demonstrated by means of several figures. Two examples show how to create your own ACLs using the correct syntax. In conclusion, find information about the way ACLs are interpreted by the operating system.

27.3.1 Structure of ACL Entries

There are two basic classes of ACLs: A *minimum* ACL merely comprises 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 27.1 provides a summary of the various types of ACL entries that are possible.

Table 27.1: *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`

The permissions defined in the entries *owner* and *other* are always effective. Except for the *mask* entry, all other entries (*named user*, *owning group*, and *named group*) can be either effective or masked. If permissions exist in one of the above-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. The example in Table 27.2 on the facing page demonstrates this mechanism.

Table 27.2: Masking Access Permissions

Entry Type	Text Form	Permissions
named user	`user:jane:r-x`	`r-x`
mask	`mask::rw-`	`rw-`
	effective permissions:	`r--`

27.3.2 ACL Entries and File Mode Permission Bits

Figure 27.1 and Figure 27.2 on the following page 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 as displayed by `ls -l`, for instance. In both cases, the *owner class* permissions are mapped to the ACL entry *owner*. Equally, *other class* permissions are mapped to the respective ACL entry. However, the mapping of the *group class* permissions is different in the two cases.

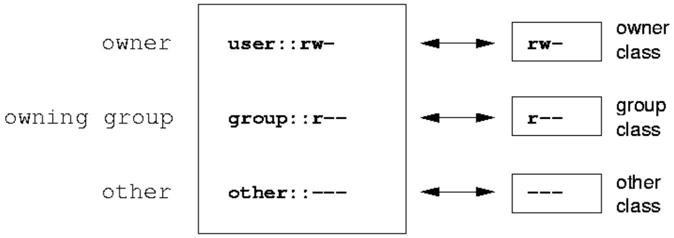


Figure 27.1: *Minimum ACL: ACL Entries Compared to Permission Bits*

- 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 27.1.
- In the case of an extended ACL — with *mask* — the *group class* permissions are mapped to the *mask* entry. This is shown in Figure 27.2 on the following page.

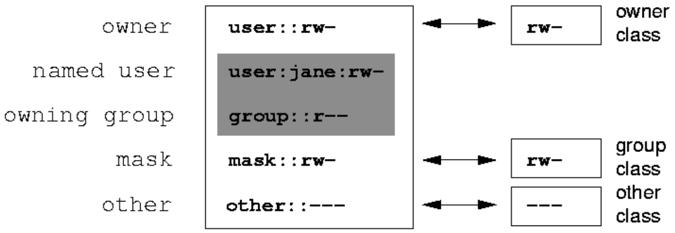


Figure 27.2: *Extended ACL: ACL Entries Compared to Permission Bits*

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 by means of ACLs. Any permissions not reflected here were either not set in the ACL or are not effective. Changes made to the permission bits are reflected by the ACL and vice versa.

27.3.3 A Directory with Access ACL

The handling of access ACLs is demonstrated in three steps by means of the following example:

- Creating a file system object (a directory in this case)
- Modifying the ACL
- Using masks

1. Before you create 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 corresponding man page (`man umask`).

   `mkdir mydir` should create the `mydir` directory with the default permissions as set by `umask`. Use the following command to check if all permissions were assigned correctly:

```
ls -dl mydir

drwxr-x--- ... tux project3 ... mydir
```

2. Check the initial state of the ACL and insert a new user entry and a new group entry with `getfacl mydir`. This gives information like:

   ```
   # file: mydir
   # owner: tux
   # group: project3
   user::rwx
   group::r-x
   other::---
   ```

   The output of `getfacl` precisely reflects the mapping of permission bits and ACL entries as described in Section 27.3.2 on page 715. 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`.

   Your first modification of the ACL is the assignment of read, write, and execute permissions to an additional user `jane` and an additional group `djungle`.

   ```
   setfacl -m user:jane:rwx,group:djungle:rwx mydir
   ```

   The option `-m` prompts `setfacl` to modify the existing ACL. The following argument indicates the ACL entries to modify (several 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:jane:rwx
   group::r-x
   group:djungle:rwx
   mask::rwx
   other::---
   ```

   In addition to the entries initiated for the user `jane` and the group `djungle`, a *mask* entry has been generated. This *mask* entry is set automatically to reduce all entries in the *group class* to a common denominator. Furthermore, `setfacl` automatically adapts existing *mask* entries to the settings modified, provided you do not 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 that would be displayed by `ls -dl mydir` now correspond to the `mask` entry.

```
drwxrwx---+ ... tux project3 ... mydir
```

The first column of the output now contains an additional `+` to indicate that there is an *extended* ACL for this item.

3. 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 27.2 on page 715). As far as the effective permissions of the *owning group* are concerned, nothing has changed even after the addition of the ACL entries.

   Edit the *mask* entry with `setfacl` or `chmod`.

```
chmod g-w mydir

ls -dl mydir

drwxr-x---+ ... tux project3 ... mydir
```

```
getfacl mydir

# file: mydir
# owner: tux
# group: project3
user::rwx
user:jane:rwx           # effective: r-x
group::r-x
group:djungle: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`:

```
chmod g+w mydir

ls -dl mydir

drwxrwx---+ ... tux project3 ... mydir


getfacl mydir

# file: mydir
# owner: tux
# group: project3
user::rwx
user:jane:rwx
group::r-x
group:djungle:rwx
mask::rwx
other::---
```

27.3.4 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 under the directory inherit when they are created. A default ACL affects subdirectories as well as files.

Effects of a Default ACL

There are two different ways in which the permissions of a directory's default ACL are passed to the files and subdirectories in it:

- A subdirectory inherits the default ACL of the parent directory both as its own default ACL and as an access ACL.
- A file inherits the default ACL as its own 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:

- Creating a default ACL for an existing directory
- Creating a subdirectory in a directory with default ACL
- Creating a file in a directory with default ACL

1. Add a default ACL to the existing directory `mydir/` with:

   ```
   setfacl -d -m group:djungle: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:jane:rwx
   group::r-x
   group:djungle:rwx
   mask::rwx
   other::---
   default:user::rwx
   default:group::r-x
   default:group:djungle: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 `djungle` 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 will “inherit” the default ACL.

```
mkdir mydir/mysubdir

getfacl mydir/mysubdir

# file: mydir/mysubdir
# owner: tux
# group: project3
user::rwx
group::r-x
group:djungle:r-x
mask::r-x
other::---
default:user::rwx
default:group::r-x
default:group:djungle: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/`, just as the default ACL that this directory will hand down to its subordinate objects.

3. Use `touch` to create a file in the `mydir/` directory:

```
touch mydir/myfile

ls -l mydir/myfile

-rw-r-----+ ... tux project3 ... mydir/myfile


getfacl mydir/myfile
```

```
# file: mydir/myfile
# owner: tux
# group: project3
user::rw-
group::r-x          # effective:r--
group:djungle:r-x   # effective:r--
mask::r--
other::---
```

Important in this example: `touch` passes on `mode` with the value `0666`, which means that new 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 27.3.4 on page 719).

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 via `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.

27.3.5 The ACL Check Algorithm

A check algorithm is applied to all processes or applications before they are 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 correct permissions, a randomly selected entry triggers the final result “access denied”.

27.4 Support by Applications

As described in the preceding sections, 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. However, some important applications still lack ACL support. Except for the star archiver, there are currently no backup applications that guarantee the full preservation of ACLs.

The basic file commands (`cp`, `mv`, `ls`, and so on) do support ACLs, but many editors and file managers (such as Konqueror) do not. When copying files with Konqueror, for instance, the ACLs of these files are lost. When modifying files with an editor, the ACLs of files are sometimes preserved, sometimes not, depending on the backup mode of the editor used:

- If the editor writes the changes to the original file, the access ACL will be preserved.
- If the editor saves the updated contents to a new file that is subsequently renamed to the old file name, the ACLs may be lost, unless the editor supports ACLs.

Note

Additional Information

Detailed information about ACLs is available at the following URLs:
`http://sdb.suse.de/en/sdb/html/81_acl.html` `http://acl.bestbits.at/`
and in the manual pages of `getfacl`, `acl(5)`, and `setfacl(1)`.

Note

28

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.

28.1 List of Open Files: lsof 726
28.2 User Accessing Files: fuser 727
28.3 File Properties: stat 728
28.4 Processes: top 728
28.5 Process List: ps 729
28.6 Process Tree: pstree 730
28.7 Who Is Doing What: w 731
28.8 Memory Usage: free 731
28.9 Kernel Ring Buffer: dmesg 732
28.10 File Systems and Their Usage: mount, df, and du . . 733
28.11 The /proc File System 734
28.12 procinfo 735
28.13 PCI Resources: lspci 736
28.14 System Calls of a Program Run: strace 737
28.15 Library Calls of a Program Run: ltrace 738
28.16 Specifying the Required Library: ldd 739
28.17 Interprocess Communication: ipcs 739

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 dollar sign prompt). Remarks

are indicated by the use of 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.

28.1 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:

```
$ lsof -p $$
COMMAND  PID USER   FD   TYPE DEVICE    SIZE     NODE NAME
zsh     4694   jj  cwd    DIR   0,18     144 25487368 /suse/jj/t (totan:/real-home/jj)
zsh     4694   jj  rtd    DIR    3,2     608        2 /
zsh     4694   jj  txt    REG    3,2  441296    20414 /bin/zsh
zsh     4694   jj  mem    REG    3,2  104484    10882 /lib/ld-2.3.3.so
zsh     4694   jj  mem    REG    3,2   11648    20610 /usr/lib/zsh/4.2.0/zsh/rlimits.so
[...]
zsh     4694   jj  mem    REG    3,2   13647    10891 /lib/libdl.so.2
zsh     4694   jj  mem    REG    3,2   88036    10894 /lib/libnsl.so.1
zsh     4694   jj  mem    REG    3,2  316410   147725 /lib/libncurses.so.5.4
zsh     4694   jj  mem    REG    3,2  170563    10909 /lib/tls/libm.so.6
zsh     4694   jj  mem    REG    3,2 1349081    10908 /lib/tls/libc.so.6
zsh     4694   jj  mem    REG    3,2      56    12410 /usr/lib/locale/de_DE.utf8/LC_TELEPHONE
[...]
zsh     4694   jj  mem    REG    3,2      59    14393 /usr/lib/locale/en_US/LC_NUMERIC
zsh     4694   jj  mem    REG    3,2  178476    14565 /usr/lib/locale/en_US/LC_CTYPE
zsh     4694   jj  mem    REG    3,2   56444    20598 /usr/lib/zsh/4.2.0/zsh/computil.so
zsh     4694   jj    0u   CHR 136,48               50 /dev/pts/48
zsh     4694   jj    1u   CHR 136,48               50 /dev/pts/48
zsh     4694   jj    2u   CHR 136,48               50 /dev/pts/48
zsh     4694   jj   10u   CHR 136,48               50 /dev/pts/48
```

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. Usually a huge number of files will be open. To find out how many files are open, enter the following:

```
$ lsof | wc -l
3749
```

List all the character devices used with:

```
$ lsof | grep CHR
sshd       4685    root  mem    CHR    1,5             45833 /dev/zero
sshd       4685    root  mem    CHR    1,5             45833 /dev/zero
sshd       4693      jj  mem    CHR    1,5             45833 /dev/zero
sshd       4693      jj  mem    CHR    1,5             45833 /dev/zero
zsh        4694      jj    0u   CHR 136,48                50 /dev/pts/48
zsh        4694      jj    1u   CHR 136,48                50 /dev/pts/48
zsh        4694      jj    2u   CHR 136,48                50 /dev/pts/48
zsh        4694      jj   10u   CHR 136,48                50 /dev/pts/48
X          6476    root  mem    CHR    1,1             38042 /dev/mem
lsof      13478      jj    0u   CHR 136,48                50 /dev/pts/48
lsof      13478      jj    2u   CHR 136,48                50 /dev/pts/48
grep      13480      jj    1u   CHR 136,48                50 /dev/pts/48
grep      13480      jj    2u   CHR 136,48                50 /dev/pts/48
```

28.2 User Accessing Files: fuser

Suppose that we want to unmount a file system under `/mnt`:

```
$ mount -l | grep /mnt
/dev/sda on /mnt type ext2 (rw,noexec,nosuid,nodev,noatime,user=jj)
```

The attempt to unmount it fails:

```
$ umount /mnt
umount: /mnt: device is busy
```

To find out which processes are accessing the files in the `/mnt` directory, enter:

```
$ fuser -v /mnt/*

                        USER          PID ACCESS COMMAND
/mnt/notes.txt
                        jj          26597 f....  less
```

Following termination of the `less` process, which was running on another terminal, the file system can successfully be unmounted.

28.3 File Properties: stat

The command `stat` displays details of the properties of a file:

```
$ stat xml-doc.txt
  File: ‘xml-doc.txt’
  Size: 632             Blocks: 8          IO Block: 4096   regular file
Device: eh/14d  Inode: 5938009     Links: 1
Access: (0644/-rw-r--r--)  Uid: (11994/      jj)   Gid: (   50/    suse)
Access: 2004-04-27 20:08:58.000000000 +0200
Modify: 2003-06-03 15:29:34.000000000 +0200
Change: 2003-07-23 17:48:27.000000000 +0200
```

The parameter `--filesystem` produces details of the properties of the file system in which the specified file is located:

```
 $ /usr/bin/stat  . --filesystem
  File: "."
    ID: 0        Namelen: 255     Type: ext2/ext3
Blocks: Total: 19347388   Free: 17831731   Available: 16848938   Size: 4096
Inodes: Total: 9830400    Free: 9663967
```

28.4 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:

```
$ top -n 1
top - 14:19:53 up 62 days,  3:35, 14 users,  load average: 0.01, 0.02, 0.00
Tasks: 102 total,   7 running,  93 sleeping,   0 stopped,   2 zombie
Cpu(s):   0.3% user,   0.1% system,   0.0% nice,  99.6% idle
Mem:    514736k total,   497232k used,    17504k free,    56024k buffers
Swap:  1794736k total,   104544k used,  1690192k free,   235872k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  Command
 1426 root      15   0  116m  41m  18m S  1.0  8.2  82:30.34 X
20836 jj        15   0   820  820  612 R  1.0  0.2   0:00.03 top
    1 root      15   0   100   96   72 S  0.0  0.0   0:08.43 init
    2 root      15   0     0    0    0 S  0.0  0.0   0:04.96 keventd
    3 root      34  19     0    0    0 S  0.0  0.0   0:00.99 ksoftirqd_CPU0
    4 root      15   0     0    0    0 S  0.0  0.0   0:33.63 kswapd
    5 root      15   0     0    0    0 S  0.0  0.0   0:00.71 bdflush
        [...]
 1362 root      15   0   488  452  404 S  0.0  0.1   0:00.02 nscd
 1363 root      15   0   488  452  404 S  0.0  0.1   0:00.04 nscd
 1377 root      17   0    56    4    4 S  0.0  0.0   0:00.00 mingetty
 1379 root      18   0    56    4    4 S  0.0  0.0   0:00.01 mingetty
 1380 root      18   0    56    4    4 S  0.0  0.0   0:00.01 mingetty
```

If you press Ⓕ 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. Here, ⟨*UID*⟩ is the user ID of the user. The following variant is useful:

```
$ top -U $(id -u <username>)
```

28.5 Process List: ps

The command `ps` produces a list of processes. If the parameter `r` is added, only those processes really running are shown:

```
$ ps r
  PID TTY      STAT   TIME COMMAND
22163 pts/7    R      0:01 -zsh
 3396 pts/3    R      0:03 emacs new-makedoc.txt
20027 pts/7    R      0:25 emacs xml/common/utilities.xml
20974 pts/7    R      0:01 emacs jj.xml
27454 pts/7    R      0:00 ps r
```

This parameter must be written *without* a minus sign. The various parameters are written sometimes with and sometimes without the minus sign. The man page could easily frighten off potential users, but fortunately, the `ps --help` command produces a brief page of help.

To check how many `emacs` processes are running, use:

```
$ ps x | grep emacs
 1288 ?        S      0:07 emacs
 3396 pts/3    S      0:04 emacs new-makedoc.txt
 3475 ?        S      0:03 emacs .Xresources
20027 pts/7    S      0:40 emacs xml/common/utilities.xml
20974 pts/7    S      0:02 emacs jj.xml

$ pidof emacs
20974 20027 3475 3396 1288
```

The parameter `-p` selects processes via the process ID:

```
$ ps www -p $(pidof xterm)
  PID TTY      STAT   TIME COMMAND
 9025 ?        S      0:01 xterm -g 100x45+0+200
 9176 ?        S      0:00 xterm -g 100x45+0+200
```

```
29854 ?        S      0:21 xterm -g 100x75+20+0 -fn \
  -B&H-LucidaTypewriter-Medium-R-Normal-Sans-12-120-75-75-M-70-iso10646-1
 4378 ?        S      0:01 xterm -bg MistyRose1 -T root -n root -e su -l
25543 ?        S      0:02 xterm -g 100x45+0+200
22161 ?        R      0:14 xterm -g 100x45+0+200
16832 ?        S      0:01 xterm -bg MistyRose1 -T root -n root -e su -l
16912 ?        S      0:00 xterm -g 100x45+0+200
17861 ?        S      0:00 xterm -bg DarkSeaGreen1 -g 120x45+40+300
19930 ?        S      0:13 xterm -bg LightCyan
21686 ?        S      0:04 xterm -g 100x45+0+200 -fn lucidasanstypewriter-12
23104 ?        S      0:00 xterm -g 100x45+0+200
26547 ?        S      0:00 xterm -g 100x45+0+200
```

28.6 Process Tree: pstree

The command `pstree` produces a list of processes in the form of a tree:

```
$ pstree
init-+-atd
     |-3*[automount]
     |-bdflush
     |-cron
  [...]
     |-usb-storage-1
     |-usb-storage-2
     |-10*[xterm---zsh]
     |-xterm---zsh---mutt
     |-2*[xterm---su---zsh]
     |-xterm---zsh---ssh
     |-xterm---zsh---pstree
     |-ypbind---ypbind---2*[ypbind]
     ‘-zsh---startx---xinit4-+-X
                             ‘-ctwm-+-xclock
                                    |-xload
                                    ‘-xosview.bin
```

The parameter `-p` adds the process ID to a given name. To have the command lines displayed as well, use the `-a` parameter:

```
$ pstree -pa
init,1
  |-atd,1255
[...]
  ‘-zsh,1404
      ‘-startx,1407 /usr/X11R6/bin/startx
          ‘-xinit4,1419 /suse/jj/.xinitrc [...]
              |-X,1426 :0 -auth /suse/jj/.Xauthority
```

```
                '-ctwm,1440
                    |-xclock,1449 -d -geometry -0+0 -bg grey
                    |-xload,1450 -scale 2
                    '-xosview.bin,1451 +net -bat +net
```

28.7 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:

```
$ w
 15:17:26 up 62 days,  4:33, 14 users,  load average: 0.00, 0.04, 0.01
USER     TTY        LOGIN@   IDLE   JCPU   PCPU WHAT
jj       pts/0     30Mar04  4days  0.50s  0.54s xterm -bg MistyRose1 -e su -l
jj       pts/1     23Mar04  5days  0.20s  0.20s -zsh
jj       pts/2     23Mar04  5days  1.28s  1.28s -zsh
jj       pts/3     23Mar04  3:28m  3.21s  0.50s -zsh
[...]
jj       pts/7     07Apr04  0.00s  9.02s  0.01s w
jj       pts/9     25Mar04  3:24m  7.70s  7.38s mutt
[...]
jj       pts/14    12:49   37:34   0.20s  0.13s ssh totan
```

The last line shows that user jj has established a secure shell (`ssh`) connection to the computer `totan`.

If any users of other systems have logged in remotely, the parameter `-f` will show the computers from which they have established the connection.

28.8 Memory Usage: free

The utility `free` examines RAM usage. Details of both free and used memory (and swap areas) are shown:

```
$ free
             total       used       free     shared    buffers     cached
Mem:        514736     273964     240772          0      35920      42328
-/+ buffers/cache:     195716     319020
Swap:      1794736     104096    1690640
```

With `-m`, have all sizes expressed in megabytes:

```
$ free -m
             total       used       free     shared    buffers     cached
Mem:           502        267        235          0         35         41
-/+ buffers/cache:        191        311
Swap:         1752        101       1651
```

The really interesting information is contained in this line:

```
-/+ buffers/cache:        191        311
```

It calculates the amount of memory taken up with buffers and caches.

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.

28.9 Kernel Ring Buffer: dmesg

The Linux kernel keeps certain messages in a ring buffer. To view these messages, enter the command `dmesg`:

```
$ dmesg
[...]
sdc : READ CAPACITY failed.
sdc : status = 1, message = 00, host = 0, driver = 08
Info fld=0xa00 (nonstd), Current sd00:00: sense key Not Ready
sdc : block size assumed to be 512 bytes, disk size 1GB.
sdc: test WP failed, assume Write Enabled
 sdc: I/O error: dev 08:20, sector 0
 I/O error: dev 08:20, sector 0
 I/O error: dev 08:20, sector 2097144
 I/O error: dev 08:20, sector 2097144
 I/O error: dev 08:20, sector 0
 I/O error: dev 08:20, sector 0
 unable to read partition table
 I/O error: dev 08:20, sector 0
nfs: server totan not responding, still trying
nfs: server totan OK
```

The last line indicates that there is a temporary problem in the NFS server `totan`. The lines up to that point are triggered by the insertion of a USB memory stick.

Older events are logged in the files `/var/log/messages` and `/var/log/warn`.

28.10 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:

```
$ mount
/dev/hdb2 on / type ext2 (rw)
proc on /proc type proc (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/hda1 on /data type ext2 (rw)
shmfs on /dev/shm type shm (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
automount(pid1012) on /suse type autofs \
    (rw,fd=5,pgrp=1012,minproto=2,maxproto=3)
totan:/real-home/jj on /suse/jj type nfs \
    (rw,nosuid,rsize=8192,wsize=8192,hard,intr,nolock,addr=10.10.0.1)
```

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.

```
$ df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/hdb2             7.4G  5.1G  2.0G  73% /
/dev/hda1              74G  5.8G   65G   9% /data
shmfs                 252M     0  252M   0% /dev/shm
totan:/real-home/jj   350G  324G   27G  93% /suse/jj
```

Users of the NFS file server `totan` should clear their home directory immediately.

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 form that ordinary people can understand. With this command:

```
$ du -sh ~
361M    /suse/jj
```

see how much space your own home directory occupies.

28.11 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, the CPU type can be ascertained with this command:

```
$ 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
[...]
```

Allocation and use of interrupts:

```
$ cat /proc/interrupts
           CPU0
  0:  537544462          XT-PIC  timer
  1:     820082          XT-PIC  keyboard
  2:          0          XT-PIC  cascade
  8:          2          XT-PIC  rtc
  9:          0          XT-PIC  acpi
 10:      13970          XT-PIC  usb-uhci, usb-uhci
 11:  146467509          XT-PIC  ehci_hcd, usb-uhci, eth0
 12:    8061393          XT-PIC  PS/2 Mouse
 14:    2465743          XT-PIC  ide0
 15:       1355          XT-PIC  ide1
NMI:          0
LOC:          0
ERR:          0
MIS:          0
```

Some of the important files and their contents are as follows:

`/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`. Query Memory usage with the command `vmstat`.

Information about processes currently running can be found in the `/proc/`⟨*NNN*⟩ directories, where ⟨*NNN*⟩ is the process ID (PID) of the relevant process. A process and its particular characteristics can be seen with `/proc/self/`:

```
$ ls -l /proc/self
lrwxrwxrwx  1 root root 64 Apr 29 13:52 /proc/self -> 27585

$ ls -l /proc/self/
total 0
dr-xr-xr-x  2 jj suse 0 Apr 29 13:52 attr
-r--------  1 jj suse 0 Apr 29 13:52 auxv
-r--r--r--  1 jj suse 0 Apr 29 13:52 cmdline
lrwxrwxrwx  1 jj suse 0 Apr 29 13:52 cwd -> /suse/jj/t
-r--r--r--  1 jj suse 0 Apr 29 13:52 delay
-r--------  1 jj suse 0 Apr 29 13:52 environ
lrwxrwxrwx  1 jj suse 0 Apr 29 13:52 exe -> /bin/ls
dr-x------  2 jj suse 0 Apr 29 13:52 fd
-rw-------  1 jj suse 0 Apr 29 13:52 mapped_base
-r--r--r--  1 jj suse 0 Apr 29 13:52 maps
-rw-------  1 jj suse 0 Apr 29 13:52 mem
-r--r--r--  1 jj suse 0 Apr 29 13:52 mounts
lrwxrwxrwx  1 jj suse 0 Apr 29 13:52 root -> /
-r--r--r--  1 jj suse 0 Apr 29 13:52 stat
-r--r--r--  1 jj suse 0 Apr 29 13:52 statm
-r--r--r--  1 jj suse 0 Apr 29 13:52 status
dr-xr-xr-x  3 jj suse 0 Apr 29 13:52 task
-r--r--r--  1 jj suse 0 Apr 29 13:52 wchan
```

The address assignment of executables and libraries is contained in the `maps` file:

```
$ cat /proc/self/maps
08048000-0804c000 r-xp 00000000 03:02 22890      /bin/cat
0804c000-0804d000 rw-p 00003000 03:02 22890      /bin/cat
0804d000-0806e000 rwxp 0804d000 00:00 0
40000000-40016000 r-xp 00000000 03:02 10882      /lib/ld-2.3.3.so
40016000-40017000 rw-p 00015000 03:02 10882      /lib/ld-2.3.3.so
40017000-40018000 rw-p 40017000 00:00 0
4002b000-40135000 r-xp 00000000 03:02 10908      /lib/tls/libc.so.6
40135000-4013d000 rw-p 0010a000 03:02 10908      /lib/tls/libc.so.6
4013d000-40141000 rw-p 4013d000 00:00 0
bfffe000-c0000000 rw-p bfffe000 00:00 0
ffffe000-fffff000 ---p 00000000 00:00 0
```

28.12 procinfo

Important information from the `/proc` file system is summarized by the command `procinfo`:

```
$ procinfo
Linux 2.4.21-144-athlon (root@i386.suse.de) (gcc 3.3.1 ) #1 \
   Fri Nov 28 01:14:40 UTC 2003 1CPU [nunez.suse.de]

Memory:      Total        Used        Free      Shared     Buffers      Cached
Mem:        514736      496088       18648           0       56128      224656
Swap:      1794736      104488     1690248

Bootup: Wed Feb 25 09:44:25 2004    Load average: 0.00 0.01 0.00 1/104 21285

user  :       4:34:59.40   0.3%  page in : 11320141  disk 1:   474842r  358260w
nice  :       0:07:12.64   0.0%  page out: 14495036  disk 2:   649679r  989842w
system:       1:15:00.55   0.1%  swap in :    58942  disk 3:     6547r     610w
idle  :  61d 22:40:40.26  99.6%  swap out:   149085  disk 4:     1169r      23w
uptime:  62d  4:37:52.84         context :767431068

irq  0: 537347285 timer                 irq 10:     13970 usb-uhci, usb-uhci
irq  1:    814562 keyboard              irq 11: 146415669 ehci_hcd, usb-uhci,
irq  2:         0 cascade [4]           irq 12:   8008998 PS/2 Mouse
irq  6:         2                       irq 14:   2463408 ide0
irq  8:         2 rtc                   irq 15:      1355 ide1
```

To see all the information, use the parameter `-a`. The parameter `-n`⟨*N*⟩ 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:

```
Memory:      Total        Used        Free      Shared     Buffers      Cached
Mem:             0           2          -2           0           0           0
Swap:            0           0           0

Bootup: Wed Feb 25 09:44:17 2004    Load average: 0.00 0.00 0.00 1/106 31902

user  :       0:00:00.02   0.4%  page in :        0  disk 1:        0r       0w
nice  :       0:00:00.00   0.0%  page out:        0  disk 2:        0r       0w
system:       0:00:00.00   0.0%  swap in :        0  disk 3:        0r       0w
idle  :       0:00:04.99  99.6%  swap out:        0  disk 4:        0r       0w
uptime:  64d  3:59:12.62         context :     1087

irq  0:       501 timer                 irq 10:         0 usb-uhci, usb-uhci
irq  1:         1 keyboard              irq 11:        32 ehci_hcd, usb-uhci,
irq  2:         0 cascade [4]           irq 12:       132 PS/2 Mouse
irq  6:         0                       irq 14:         0 ide0
irq  8:         0 rtc                   irq 15:         0 ide1
irq  9:         0 acpi
```

28.13 PCI Resources: lspci

The command `lspci` lists the PCI resources:

```
$ lspci
00:00.0 Host bridge: VIA Technologies, Inc. \
    VT8366/A/7 [Apollo KT266/A/333]
00:01.0 PCI bridge: VIA Technologies, Inc. \
    VT8366/A/7 [Apollo KT266/A/333 AGP]
00:0b.0 Ethernet controller: Digital Equipment Corporation \
    DECchip 21140 [FasterNet] (rev 22)
00:10.0 USB Controller: VIA Technologies, Inc. USB (rev 80)
00:10.1 USB Controller: VIA Technologies, Inc. USB (rev 80)
00:10.2 USB Controller: VIA Technologies, Inc. USB (rev 80)
00:10.3 USB Controller: VIA Technologies, Inc. USB 2.0 (rev 82)
00:11.0 ISA bridge: VIA Technologies, Inc. VT8235 ISA Bridge
00:11.1 IDE interface: VIA Technologies, Inc. VT82C586/B/686A/B \
    PIPC Bus Master IDE (rev 06)
00:11.5 Multimedia audio controller: VIA Technologies, Inc. \
    VT8233 AC97 Audio Controller (rev 50)
01:00.0 VGA compatible controller: Matrox Graphics, Inc. \
    MGA G550 AGP (rev 01)
```

Using `-v` results in a more detailed listing:

```
$ lspci -v
[...]
01:00.0 \
 VGA compatible controller: Matrox Graphics, Inc. MGA G550 AGP (rev 01) \
    (prog-if 00 [VGA])
 Subsystem: Matrox Graphics, Inc. Millennium G550 Dual Head DDR 32Mb
 Flags: bus master, medium devsel, latency 32, IRQ 10
 Memory at d8000000 (32-bit, prefetchable) [size=32M]
 Memory at da000000 (32-bit, non-prefetchable) [size=16K]
 Memory at db000000 (32-bit, non-prefetchable) [size=8M]
 Expansion ROM at <unassigned> [disabled] [size=128K]
 Capabilities: <available only to root>
```

Information about device name resolution is obtained from 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, you should use the parameter `-n`.

28.14 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:

```
$ strace ls

execve("/bin/ls", ["ls"], [/* 88 vars */]) = 0
uname({sys="Linux", node="edison", ...}) = 0
brk(0)                                  = 0x805b000
old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) \
          = 0x40017000
open("/etc/ld.so.preload", O_RDONLY)    = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=76333, ...}) = 0
old_mmap(NULL, 76333, PROT_READ, MAP_PRIVATE, 3, 0) = 0x40018000
[...]
ioctl(1, SNDCTL_TMR_TIMEBASE or TCGETS, {B38400 opost isig icanon echo ...}) = 0
ioctl(1, TIOCGWINSZ, {ws_row=53, ws_col=110, ws_xpixel=897, ws_ypixel=693}) = 0
open(".", O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY) = 3
fstat64(3, {st_mode=S_IFDIR|0755, st_size=144, ...}) = 0
fcntl64(3, F_SETFD, FD_CLOEXEC)         = 0
getdents64(3, /* 5 entries */, 4096)    = 160
getdents64(3, /* 0 entries */, 4096)    = 0
close(3)                                = 0
fstat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 48), ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) \
          = 0x40018000
write(1, "ltrace-ls.txt  myfile.txt  strac"..., 41) = 41
munmap(0x40018000, 4096)                = 0
exit_group(0)                           = ?
```

For example, to trace all attempts to open a particular file, use the following:

```
$ strace ls myfile.txt 2>&1 | grep open
open("/etc/ld.so.preload", O_RDONLY)    = -1 ENOENT (No such file or directory)
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`.

28.15 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:

```
$ strace -c find /usr/share/doc
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 86.27    1.071814          30     35327           write
 10.15    0.126092          38      3297           getdents64
  2.33    0.028931           3     10208           lstat64
  0.55    0.006861           2      3122         1 chdir
  0.39    0.004890           3      1567         2 open
[...]
  0.00    0.000003           3         1           uname
  0.00    0.000001           1         1           time
------ ----------- ----------- --------- --------- ----------------
100.00    1.242403                 58269         3 total
```

28.16 Specifying the Required Library: ldd

With the command `ldd`, find out which libraries the dynamic executable specified as argument would be subsequently loaded:

```
$ ldd /bin/ls
linux-gate.so.1 =>  (0xffffe000)
librt.so.1 => /lib/tls/librt.so.1 (0x4002b000)
libacl.so.1 => /lib/libacl.so.1 (0x40033000)
libselinux.so.1 => /lib/libselinux.so.1 (0x40039000)
libc.so.6 => /lib/tls/libc.so.6 (0x40048000)
libpthread.so.0 => /lib/tls/libpthread.so.0 (0x4015d000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
libattr.so.1 => /lib/libattr.so.1 (0x4016d000)
```

Static binaries do not need any dynamic libraries:

```
$ ldd /bin/rpm
        not a dynamic executable
$ file /bin/rpm
/bin/rpm: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), \
      statically linked, stripped
```

28.17 Interprocess Communication: ipcs

The command `ipcs` produces a list of the IPC resources currently in use:

```
$ ipcs
------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x000027d9 5734403    toms      660        64528      2
0x00000000 5767172    toms      666        37044      2
0x00000000 5799941    toms      666        37044      2

------ Semaphore Arrays --------
key        semid      owner      perms      nsems
0x000027d9 0          toms      660        1

------ Message Queues --------
key        msqid      owner      perms      used-bytes   messages
```

Part V

Appendix

Information sources and documentation

This section explains where you can find information and documentation about your system.

SUSE documentation

You will find detailed information in our books in HTML or PDF format in the RPM packages `suselinux-adminguide_en` and `suselinux-adminguide_en-pdf`).

The books are installed in the `/usr/share/doc/manual/` directory in a standard installation. The SUSEHelpCenters give you access to this information.

The Linux Documentation Project (TLDP)

The Linux Documentation Project (see `http://www.tldp.org/`) is a team of volunteers who produce documentation about Linux. TLDP contains HOWTOs, FAQs and guides, all of which have been published under a free license.

HOWTOs are step-by-step instructions and are intended for end users, system administrators and programmers. For example, the creation of a DHCP server is described in a HOWTO, as well as the points to be noted, but not how Linux as such is installed. As a rule, documentation of this kind is kept quite general so that it can be applied to every distribution. The `howto` package contains HOWTOs in ASCII format. Users who prefer HTML should install `howtoenh`.

FAQs (*Frequently Asked Questions*) are collections of questions and answers relating to certain problem areas that frequently arise in mailing lists. For example "What is LDAP?", "What is a RAID?" etc. Texts in this category are generally quite short.

Guides are books that can deal with a topic in much greater detail than HOWTOs and FAQs. Examples are kernel programming, network administration etc. The underlying idea is to provide the reader with detailed information.

Some TLDP documentation is also available in other formats, such as PDF, single and multiple HTML pages, PostScript and as SGML/XML sources. In some cases there are also translations into different languages.

Man pages and Info pages

A man page (*Manual page*) is a help text for a command, system call, file format, etc. A man page is normally divided into various sections, such as name, syntax, description, options, files, etc.

To display a man page, enter:

```
man ls
```

The previous entry shows a help text for the `ls` command. The cursor keys allow you to move the visible area, while ⓠ exits `man`. To print a man page (for example for the command `ls`), enter:

```
card ls
```

For more help for the `card` (package `a2ps`) command, use the `--help` option.

Some documentation is also available in Info format, for example `grep`. The call is:

```
info grep
```

Info pages are more detailed than man pages and are divided into different "nodes". A node is a page that can be read similarly to an HTML browser with an Info reader. The Ⓟ (previous page) and Ⓝ (next page) keys are used to navigate in an info page. Ⓠ is used to exit `info`. Other keys are listed in the `info` documentation (call `info info`).

Both man pages and info pages can be called in Konqueror by entering `man:`⟨*command*⟩ or `info:`⟨*command*⟩ in the URL line.

Standards and specifications

There are various information options if you require information about standards or specifications:

`http://www.w3.org` The *World Wide Web Consortium* (W3C) is certainly one of the best-known organizations and was founded in October 1994 by TIM BERNERS-LEE and concentrates on the standardizing of web technologies. It promotes the dissemination of open, license-free and manufacturer-independent specifications, such as HTML, XHTML, XML and others. These "web standards" are developed in a 4-stage process in so-called *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,* see `http://www.ietf.org/rfc.html`) and is 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 also the summary under `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, etc. IEEE standards are liable to a charge.

`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 liable to a charge.

`http://www.din.de`, `http://www.din.com`

The Deutsches Institut fÃ¼r Normung (DIN) is a registered technical/scientific association and 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 liable to a charge and can be ordered using the DIN homepage.

B

Manual Page of e2fsck

```
E2FSCK(8)                                                    E2FSCK(8)


NAME
       e2fsck - check a Linux second extended file system

SYNOPSIS
       e2fsck  [ -pacnyrdfvstDFSV ] [ -b superblock ] [ -B block
       size ] [ -l|-L bad_blocks_file ] [ -C fd ] [ -j  external-
       journal ] [ -E extended_options ] device

DESCRIPTION
       e2fsck  is used to check a Linux second extended file sys
       tem (ext2fs).  E2fsck also supports ext2 filesystems coun
       taining  a journal, which are also sometimes known as ext3
       filesystems, by first applying the journal to the filesys
       tem  before  continuing  with  normal  e2fsck  processing.
       After the journal has been applied, a filesystem will nor
       mally  be  marked  as clean.  Hence, for ext3 filesystems,
       e2fsck will normally run the journal and exit, unless  its
       superblock indicates that further checking is required.

       device  is  the device file where the filesystem is stored
       (e.g.  /dev/hdc1).

OPTIONS
       -a     This option does the same thing as the  -p  option.
              It is provided for backwards compatibility only; it
              is suggested that people  use  -p  option  whenever
              possible.

       -b superblock
              Instead  of  using  the  normal  superblock, use an
              alternative  superblock  specified  by  superblock.
              This  option  is  normally  used  when  the primary
              superblock has been corrupted.  The location of the
```

backup superblock is dependent on the filesystem's blocksize. For filesystems with 1k blocksizes, a backup superblock can be found at block 8193; for filesystems with 2k blocksizes, at block 16384; and for 4k blocksizes, at block 32768.

Additional backup superblocks can be determined by using the mke2fs program using the -n option to print out where the superblocks were created. The -b option to mke2fs, which specifies blocksize of the filesystem must be specified in order for the superblock locations that are printed out to be accurate.

If an alternative superblock is specified and the filesystem is not opened read-only, e2fsck will make sure that the primary superblock is updated appropriately upon completion of the filesystem check.

-B blocksize

Normally, e2fsck will search for the superblock at various different block sizes in an attempt to find the appropriate block size. This search can be fooled in some cases. This option forces e2fsck to only try locating the superblock at a particular blocksize. If the superblock is not found, e2fsck will terminate with a fatal error.

-c

This option causes e2fsck to run the badblocks(8) program to find any blocks which are bad on the filesystem, and then marks them as bad by adding them to the bad block inode. If this option is specified twice, then the bad block scan will be done using a non-destructive read-write test.

-C fd

This option causes e2fsck to write completion information to the specified file descriptor so that the progress of the filesystem check can be monitored. This option is typically used by programs which are running e2fsck. If the file descriptor specified is 0, e2fsck will print a completion bar as it goes about its business. This requires that e2fsck is running on a video console or terminal.

-d

Print debugging output (useless unless you are debugging e2fsck).

-D

Optimize directories in filesystem. This option causes e2fsck to try to optimize all directories, either by reindexing them if the filesystem supports directory indexing, or by sorting and com

```
              pressing  directories  for  smaller directories, or
              for filesystems using traditional  linear  directo
              ries.

       -E extended_options
              Set  e2fsck extended options.  Extended options are
              comma separated, and may take an argument using the
              equals  (’=’) sign.  The following options are sup
              ported:

                   ea_ver=extended_attribute_version
                          Assume  the  format  of  the   extended
                          attribute  blocks  in the filesystem is
                          the specified version number.  The ver
                          sion number may be 1 or 2.  The default
                          extended attribute version format is 2.

       -f     Force checking even if the file system seems clean.

       -F     Flush the filesystem device’s buffer caches  before
              beginning.   Only  really  useful  for doing e2fsck
              time trials.

       -j external-journal
              Set the pathname  where  the  external-journal  for
              this filesystem can be found.

       -l filename
              Add  the block numbers listed in the file specified
              by filename to the list of bad blocks.  The  format
              of  this  file  is the same as the one generated by
              the badblocks(8) program.  Note that the block num
              bers  are based on the blocksize of the filesystem.
              Hence, badblocks(8) must be given the blocksize  of
              the  filesystem in order to obtain correct results.
              As a result, it is much simpler and  safer  to  use
              the  -c option to e2fsck, since it will assure that
              the correct parameters are passed to the  badblocks
              program.

       -L filename
              Set  the  bad  blocks list to be the list of blocks
              specified by filename.  (This option is the same as
              the  -l  option,  except  the  bad  blocks  list is
              cleared before the blocks listed in  the  file  are
              added to the bad blocks list.)

       -n     Open the filesystem read-only, and assume an answer
              of ‘no’ to all questions.  Allows e2fsck to be used
              non-interactively.   (Note:  if  the  -c, -l, or -L
              options are specified in addition to the -n option,
              then  the  filesystem will be opened read-write, to
              permit the bad-blocks list to be updated.  However,
              no other changes will be made to the filesystem.)

       -p     Automatically  repair  ("preen")  the  file  system
              without any questions.

       -r     This option does nothing at  all;  it  is  provided
              only for backwards compatibility.

       -s     This  option  will byte-swap the filesystem so that
              it is using  the  normalized,  standard  byte-order
              (which  is i386 or little endian).  If the filesys
              tem is already in the standard  byte-order,  e2fsck
              will take no action.

       -S     This  option will byte-swap the filesystem, regard
              less of its current byte-order.

       -t     Print timing statistics for e2fsck.  If this option
              is  used  twice,  additional  timing statistics are
              printed on a pass by pass basis.

       -v     Verbose mode.

       -V     Print version information and exit.

       -y     Assume an answer of ‘yes’ to all questions;  allows
              e2fsck to be used non-interactively.

EXIT CODE
       The exit code returned by e2fsck is the sum of the follow
       ing conditions:
            0    - No errors
            1    - File system errors corrected
            2    - File system errors corrected, system should
                   be rebooted
            4    - File system errors left uncorrected
            8    - Operational error
            16   - Usage or syntax error
            32   - E2fsck canceled by user request
            128  - Shared library error

SIGNALS
       The following signals have the following effect when  sent
       to e2fsck.

       SIGUSR1
              This  signal  causes  e2fsck  to start displaying a
              completion bar.  (See discussion of the -C option.)

       SIGUSR2
              This signal causes e2fsck to stop displaying a com
              pletion bar.
```

```
REPORTING BUGS
       Almost any piece of  software  will  have  bugs.   If  you
       manage  to find a filesystem which causes e2fsck to crash,
       or which e2fsck is unable to repair, please report  it  to
       the author.

       Please include as much information as possible in your bug
       report.  Ideally, include a  complete  transcript  of  the
       e2fsck  run,  so I can see exactly what error messages are
       displayed.  If you have a writeable filesystem  where  the
       transcript can be stored, the script(1) program is a handy
       way to save the output of e2fsck to a file.

       It is also useful to send the output of dumpe2fs(8).  If a
       specific  inode  or inodes seems to be giving e2fsck trou
       ble, try running the debugfs(8) command and send the  out
       put  of the stat(1u) command run on the relevant inode(s).
       If the inode is a directory, the debugfs dump command will
       allow  you to extract the contents of the directory inode,
       which can sent to me after being first run  through  uuen
       code(1).

       Always  include  the full version string which e2fsck dis
       plays when it is run, so I know which version you are run
       ning.

AUTHOR
       This  version  of  e2fsck  was  written  by  Theodore Ts'o
       <tytso@mit.edu>.

SEE ALSO
       mke2fs(8), tune2fs(8), dumpe2fs(8), debugfs(8)



E2fsprogs version 1.34      July 2003                   E2FSCK(8)
```

C

Manual Page of reiserfsck

```
REISERFSCK(8)                                       REISERFSCK(8)

NAME
       reiserfsck - check a Linux Reiserfs file system

SYNOPSIS
       reiserfsck  [  -afprVy ] [ --rebuild-sb | --check | --fix-
       fixable | --rebuild-tree | --clean-attributes  ]  [  -j  |
       --journal device ] [ -z | --adjust-size ] [ -n | --nolog ]
       [ -l | --logfile file ] [ -q | --quiet ] [ -y | --yes ]  [
       -S  |  --scan-whole-partition ] [ --no-journal-available ]
       device

DESCRIPTION
       Reiserfsck searches for a Reiserfs filesystem on a device,
       replays  any  necessary transactions, and either checks or
       repairs the file system.

       device is the special file corresponding to the device  or
              partition  (e.g /dev/hdXX for IDE disk partition or
              /dev/sdXX for SCSI disk partition).

OPTIONS
       --rebuild-sb
              This option recovers the superblock on  a  Reiserfs
              partition.   Normally  you only need this option if
              mount reports "read_super_block: can’t find a reis
              erfs  file system" and you are sure that a Reiserfs
              file system is there.

       --check
              This default action checks file system  consistency
              and reports but does not repair any corruption that
              it finds. This option may be used  on  a  read-only
              file system mount.

       --fix-fixable
```

```
              This  option  recovers  certain kinds of corruption
              that do not require rebuilding the entire file sys
              tem  tree  (--rebuild-tree). Normally you only need
              this option if the --check option reports  "corrup
              tion  that  can  be fixed with --fix-fixable". This
              includes: zeroing invalid data-block pointers, cor
              recting  st_size and st_blocks for directories, and
              deleting invalid directory entries.

       --rebuild-tree
              This option rebuilds the entire  file  system  tree
              using leaf nodes found on the device.  Normally you
              only need this option if the --check option reports
              "corruption   that   can   be   fixed  only  during
              --rebuild-tree".  You are  strongly  encouraged  to
              make  a  backup  copy of the whole partition before
              attempting the --rebuild-tree option.

       --clean-attributes
              This option cleans  reserved  fields  of  Stat-Data
              items.

       --journal device , -j device
              This option supplies the device name of the current
              file system journal.  This option is required  when
              the  journal  resides on a separate device from the
              main data device (although it can be  avoided  with
              the expert option --no-journal-available).

       --adjust-size, -z
              This option causes reiserfsck to correct file sizes
              that are larger than the offset of the last discov
              ered byte.  This implies that holes at the end of a
              file will be removed.  File sizes that are  smaller
              than  the  offset  of  the last discovered byte are
              corrected by --fix-fixable.

       --logfile file, -l  file
              This option causes reiserfsck to report any corrup
              tion it finds to the specified log file rather than
              stderr.

       --nolog, -n
              This option prevents reiserfsck from reporting  any
              kinds of corruption.

       --quiet, -q
              This  option prevents reiserfsck from reporting its
              rate of progress.

       --yes, -y
              This option inhibits reiserfsck from asking you for
              confirmation  after telling you what it is going to
              do, assuming yes. For safety, it does not work with
              the --rebuild-tree option.

       -a, -p These  options are usually passed by fsck -A during
              the automatic checking of those  partitions  listed
              in  /etc/fstab.  These  options cause reiserfsck to
              print some information  about  the  specified  file
              system,  check if error flags in the superblock are
              set and  do  some  light-weight  checks.  If  these
              checks reveal a corruption or the flag indicating a
              (possibly fixable) corruption is found set  in  the
              superblock,  then  reiserfsck  switches to the fix-
              fixable mode. If the flag indicating a  fatal  cor
              ruption  is found set in the superblock, then reis
              erfsck finishes with an error.

       -V     This option prints the  reiserfsprogs  version  and
              exit.

       -r, -f These options are ignored.

EXPERT OPTIONS
       DO  NOT  USE  THESE  OPTIONS  UNLESS YOU KNOW WHAT YOU ARE
       DOING.  WE ARE NOT RESPONSIBLE  IF  YOU  LOSE  DATA  AS  A
       RESULT OF THESE OPTIONS.

       --no-journal-available
              This  option  allows reiserfsck to proceed when the
              journal device is not available. This option has no
              effect when the journal is located on the main data
              device. NOTE: after this  operation  you  must  use
              reiserfstune to specify a new journal device.

       --scan-whole-partition, -S
              This option causes --rebuild-tree to scan the whole
              partition, not only used space on the partition.

EXAMPLE OF USING
       1. You think something may be wrong with a reiserfs parti
       tion  on  /dev/hda1  or  you  would just like to perform a
       periodic disk check.

       2. Run reiserfsck --check --logfile  check.log  /dev/hda1.
       If  reiserfsck  --check  exits  with  status 0 it means no
       errors were discovered.

       3. If reiserfsck --check exits with status 1 (and  reports
       about  fixable  corruptions)  it means that you should run
       reiserfsck --fix-fixable --logfile fixable.log  /dev/hda1.

       4.  If reiserfsck --check exits with status 2 (and reports
       about fatal corruptions) it means that  you  need  to  run
       reiserfsck --rebuild-tree.  If reiserfsck --check fails in
       some way you should also  run  reiserfsck  --rebuild-tree,
       but  we also encourage you to submit this as a bug report.

       5. Before running reiserfsck --rebuild-tree, please make a
       backup  of the whole partition before proceeding. Then run
       reiserfsck --rebuild-tree --logfile rebuild.log /dev/hda1.

       6.  If  the  --rebuild-tree step fails or does not recover
       what you expected, please submit this as a bug report. Try
       to provide as much information as possible and we will try
       to help solve the problem.

EXIT CODES
       reiserfsck uses the following exit codes:
          0 - No errors.
          1 - File system errors corrected.
          4 - File system fatal errors left uncorrected,
              reiserfsck --rebuild-tree needs to be launched.
          6 - File system fixable errors left uncorrected,
              reiserfsck --fix-fixable needs to be launched.
          8 - Operational error.
          16 - Usage or syntax error.



AUTHOR
       This version of reiserfsck  has  been  written  by  Vitaly
       Fertman <vitaly@namesys.com>.

BUGS
       There  are  likely  to be some bugs. Please report bugs to
       the ReiserFS mail-list <reiserfs-list@namesys.com>.

TODO
       Faster recovering, signal handling,  i/o  error  handling,
       etc.

SEE ALSO
       mkreiserfs(8),  reiserfstune(8)  resize_reiserfs(8), debu
       greiserfs(8),

Reiserfsprogs-3.6.9         April 2003              REISERFSCK(8)
```

D

The GNU General Public License

GNU General Public License

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Foreword

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the *GNU General Public License* is intended to guarantee your freedom to share and change free software — to make sure the software is free for all its users. This *General Public License* applies to most of the *Free Software Foundation's* software and to any other program whose authors commit to using it. (Some other *Free Software Foundation* software is covered by the *GNU Library General Public License* instead.) You can apply it to your programs, too.

When we speak of *"free" software*, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU General, Public License

Terms and Conditions for Copying, Distribution and Modification

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this *General Public License*. The "Program", below, refers to any such program or work, and a *work based on the Program* means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

1. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
2. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
3. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this

License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

1. Accompany it with the complete corresponding machine--readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

2. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

3. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, “complete source code” means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty--free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The *Free Software Foundation* may publish revised and/or new versions of the *General Public License* from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the *Free Software Foundation*. If the Program does not specify a version number of this License, you may choose any version ever published by the *Free Software Foundation*.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the *Free Software Foundation*, write to the *Free Software Foundation*; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

No Warranty

11. Because the program is licensed free of charge, there is no warranty for the program, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holders and/or other parties provide the program "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.

12. In no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute the program as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.

End of Terms and Conditions

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program’s name and a brief
idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public
License along with this program; if not, write to the Free
Software Foundation, Inc., 59 Temple Place, Suite 330, Boston,
MA 02111-1307, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) <year> <name of author>

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
’show w’. This is free software, and you  are welcome to
redistribute it under certain conditions; type ’show c’ for
details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program ‘Gnomovision’ (which makes passes at compilers) written
by James Hacker.

signature of Ty Coon, 1st April 1989 Ty Coon, President of Vice
```

This *General Public License* does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the *GNU Library General Public License* instead of this License.

Bibliography

[1] *SUSE LINUX* (*User Guide*). SUSE, 9. Edition ©2004 .

[2] EDWARD C. BAILEY. *Maximum RPM*. ©1997 . ISBN 1-888172-78-9.

[3] BRYAN COSTALES, ERIC ALLMAN, NEIL RICKERT. *sendmail*. ©1993 . ISBN 1-56592-056-2.

[4] WERNER ALMESBERGER. *LILO User's guide*. `file:///usr/share/doc/lilo/user.dvi`.

[5] OLAF KIRCH. *LINUX Network Administrator's Guide*. ©1995 . ISBN 1-56592-087-2.

[6] SIMON GARFINKEL, GENE SPAFFORD. *Practical UNIX Security*. ©1993 . ISBN 0-937175-72-2.

[7] CRAIG HUNT. *TCP/IP Netwerk Administration*. ©1995 . ISBN 3-930673-02-9.

[8] TIM O'REILLY, GRACE TODINO. *Managing UUCP and Usenet*. ©1992 . ISBN 0-937175-93-5.

[9] MATT WELSH. *Linux Installation and Getting Started*. 2. Edition ©1994 . ISBN 3-930419-03-3.

[10] LINDA LAMB. *Learning the vi Editor*. ©1990 . ISBN 0-937175-67-6.

[11] MATT WELSH, LARS KAUFMAN. *Running Linux*. ©1995 O'Reilly. ISBN 1-56592-100-3.

[12] WILLIAM R. CHESWICK, STEVEN M. BELLOVIN. *Firewalls and Internet Security* (*Repelling the Wily Hacker*). 2. Edition ©2003 Addison-Wesley Pub Co. ISBN 0-201-63466-X.

[13] BRENT CHAPMAN, ELISABETH D. ZWICKY. *Building Internet Firewalls*. ©1995 O'Reilly and Associates. ISBN 1-565-92124-0.

[14] CLIFFORD STOLL. *Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage* . ©2000 Pocket Books. ISBN 0-743-41146-3.

[15] BRIAN TUNG. *Kerberos: A Network Authentication System*. ©1999 Fischer-TB. Verlag. ISBN 0-201-37924-4.

Index

symbols

.local as top-level domain 163
64-bit Linux 191
  - Kernel specifications 195
  - Runtime support 192
  - Software development 193

A

ACLs
  - access 707, 710
  - check algorithm 716
  - default 707, 713
  - masks 711
  - structure 708
ACLs (Access Control Lists) 705–717
ACPI
  - disabling 10
addresses
  - IP 410
  - MAC 410
ADSL
  - configuring 608–609
  - dial-on-demand 609
Apache 237, 537–561
  - apxs 543
  - CGI 551
  - configuring 544–549
  - content negotiation 540
  - default page 539
  - DocumentRoot 545
  - flags 545
  - installing 542–544
  - logging 548, 549
  - modules 540
    · activating 544
    · loading 546
    · mod_perl 552
    · mod_php4 555
    · mod_python 555
    · mod_ruby 555
  - permissions 546, 559
  - security 559
  - Squid 623
  - SSI 548, 551
  - starting 543
  - threads 542
  - troubleshooting 560
  - virtual hosts 540, 556–558
AppleTalk . . . . *see* Netatalk
authentication
  - PAM 395–402

B

backups 64
  - creating with YaST 93
  - restoring 93
Bash
  - .bashrc 237
  - .profile 237
  - profile 237
BIND 451–460
BIOS 8, 198
  - virus protection 118
  - Wake on LAN (WOL) 373
Bluetooth 344
  - hciconfig 346
  - hcitool 346
  - opd 348
  - pand 347
  - sdptool 347
boot disks 199
  - creating

    · DOS 122
  - creating with dd 124
  - creating with rawrite 123
boot loader
  - ZIPL 222
booting 197–747
  - BIOS 8, 198
  - boot loaders 201, 245
  - boot managers 197–199
  - boot sectors 198–199
  - CD, from 8
  - CD 2, from 125
  - CMOS 198
  - concepts 199
  - configuring 32
    · YaST 216–219
  - DOS 199
  - floppy disks, from 122, 125
  - graphic 118
    · disabling 118
  - GRUB 118, 201–209
  - initial ramdisk 243–247
  - initrd
    · creating 246
  - LILO 118, 209
  - linuxrc 244
  - loader 218
    · locations for 218
  - log 100
  - map files 200
  - methods 118
  - PCMCIA, from 328
  - securing 214
  - system freezes 118
  - Windows 199
  - Windows NT 199

C

cable modem 436
cardmgr 446
cards
  - graphics 73
    · drivers 276
  - ISDN 441
  - network 434
    · testing 433
  - PCMCIA 324–334, 445–446
    · removing 326
  - sound 82
CD-ROM drives
  - ATAPI 126
  - supported 126
CDs
  - booting 8
chown 163
CID-keyed fonts 282
CJK 252
CMOS 198
Cold Standby 388
coldplug 315
commands
  - apm 356
  - cardctl 333
  - chown 163
  - e2fsck 741
  - ether-wake 374
  - fonts-config 278
  - free 241
  - getfacl 711
  - grub 201
  - head 163
  - hotplug 313
  - hwinfo 314
  - ifport 332
  - kadmin 678
  - kinit 684
  - kstash 677
  - ktutil 686
  - ldapadd 491
  - ldapdelete 494
  - ldapmodify 493
  - ldapsearch 494, 689
  - mii_tool 332
  - nice 163
  - scp 662
  - setfacl 711
  - sftp 663
  - smbpasswd 591
  - sort 163
  - ssh 661
  - ssh-agent 666
  - ssh-keygen 665
  - tail 163
  - udev 317
configuration
  - cable modem 436
  - DSL 439
  - ISDN 441
  - mail server 86
  - modem 437
  - network 434
  - T-DSL 441
configuration files 426
  - .bashrc 237, 241
  - .emacs 242
  - .mailsync 579

- .profile 237
- .xsession 666
- /etc/group 156
- /etc/hotplug 312
- /etc/netatalk/netatalk.conf 597
- /etc/passwd 156
- acpi 357
- afpd.conf 598
- apache2 544
- AppleVolumes.default 599
- AppleVolumes.system 602
- asound.conf 83
- atalkd.conf 598
- boot/grub/menu.lst 202
- config 227
- crontab 238
- csh.cshrc 254
- dhclient.conf 519
- dhcp 427
- dhcpd.conf 520
- exports 516, 517, 621
- fstab 29, 131, 250
- grub.conf 207
- gshadow 164
- host.conf 428
  - · alert 429
  - · multi 429
  - · nospoof 429
  - · order 429
  - · trim 429
- HOSTNAME 432
- hosts 413, 428
- hotplug 315
- httpd.conf 237, 544, 545
- hwinfo 314
- hwup 313
- ifcfg-* 426
- inittab 252, 258, 260
- irda 342
- kernel 246
- krb5.conf 679, 682, 687
- krb5.keytab 685
- language 253, 254
- lilo.conf 211–245
- logrotate.conf 239
- menu.lst 233, 245
- modprobe.conf 161, 229–230
- modules.conf 83, 161
- modules.dep 229
- named.conf 452–460, 615
- Netatalk 597
- netatalk.conf 603
- network 332, 427
  - · providers 608
- networks 428
- nscd.conf 432
- nsswitch.conf 429, 499
- openldap 688
- pam_unix2.conf 498, 686
- papd.conf 602
- pcmcia 325–327, 330, 331
- permissions 700
- powermanagement 355, 357
- profile 237, 240, 254
- rc.config 98, 266
- resolv.conf 242, 427, 452, 614
- routes 446
- samba 590
- services 590
- slapd.conf 485, 689, 690
- smb.conf 584, 586
- smppd.conf 606
- smpppd-c.conf 607
- squid.conf 614, 615, 618, 621, 624, 626
- squidguard.conf 626
- ssh_config 687
- sshd_config 666, 687
- suseconfig 268
- sysconfig 98, 266–268
- syslinux.cfg 245
- wireless 427
- XF86Config 272
  - · Device 275
  - · Monitor 276
  - · Screen 274

configuring 266
- Apache 544–549
- booting 201
- CD-ROM 65
- DNS 84, 451
- e-mail 85
- firewalls 92
- graphics cards 73
- groups 88
- hard disk controller 69
- hard disks
  - · DMA 79
- hardware 65
- IPv6 446
- IrDA 342
- keyboard 99
- language 99
- mouse 80
- Netatalk 597
- network 83–85

- networks
  - · manually 424–447
- NFS 84
- NTP
  - · clients 85
- printing 66
- routing 85, 446
- Samba 586–592
  - · clients 85
  - · servers 84
- scanner 80
- security 88–92
- software 49–63
- sound cards 82
- Squid 615
- SSH 661
- system 47–101
- system services 87
- time zone 99
- users 88
- X 70

consoles
- assigning 252
- graphical
  - · disabling 118
- switching 252

core files 240

cpuspeed 363

crashes 741, 747

cron 238

CVS 565, 571–574

D

depmod 229

device nodes
- udev 317

DHCP 518–531
- Configuration with YaST 524
- dhcpd 520–521
- packages 519
- server 520–521
- static address assignment 522

disks
- boot 94
  - · creating 221
- booting from 199
- floppy
  - · formatting 124
- module 94
- rescue 94

Distributed Replicated Block Device *see* DRBD

DNS 413
- BIND 451–460
- configuring 84, 451
- domains 427
- forwarding 452
- logging 455
- mail exchanger 414
- name servers 427
- NIC 413
- options 454
- reverse lookup 459
- security and 699
- Squid and 615
- starting 452
- top level domain 413
- troubleshooting 452
- zones
  - · files 457

domain name system *see* DNS

DOS
- sharing files 584

DRBD 392

E

e-mail
- configuring 85
- synchronizing 565
  - · mailsync 579–582

e2fsck 741

editors
- Emacs 242–243

Emacs 242–243
- .emacs 242
- default.el 242

encoding
- UTF-8 163

error messages
- bad interpreter 29
- permission denied 29

ESA Native installation
- IPLing 34

F

FAT file system 27

fdisk 220
- mbr 220

FHS *see* file systems, FHS

file systems 376–385
- access control lists 706–717
- e2fsck 741
- Ext2 377
- Ext3 378–379
- FAT 27
- FHS 236

- JFS 380–381
- LFS 383
- limitations 383
- NTFS 27, 28
- ReiserFS 379–380
- reiserfsck 747
- repairing 251
- selecting 376
- supported 382–383
- sysfs 312
- terms 376
- TeX and 236
- XFS 381–382

files
- sharing 84
- synchronize
  · rsync 566
  · subversion 565
- synchronizing 563–582
  · CVS 565, 571–574
  · mailsync 565, 579–582
  · Unison 564, 570–571

firewalls 92, 653
- packet filters 653, 656
- Squid and 621
- SUSEfirewall2 653
- SuSEfirewall2 657

font systems 278
- CID-keyed fonts 282
- X11 core fonts 281
- Xft 278

fonts 278
- CID-keyed 282
- X11 core 281
- Xft 278

FTP
- servers 236

G

GPL 751

graphical user interface 70–79

graphics
- 3D 283–285
  · 3Ddiag 285
  · diagnosis 284
  · SaX2 284
  · testing 284
  · troubleshooting 285
- cards
  · drivers 276
- GLIDE 283–285
- OpenGL 283–285
  · drivers 283
  · testing 284

graphics cards
- 3D 283–285
  · drivers 283
  · installation support for 285
  · support for 283

groups
- administering 88

GRUB 201
- boot menu 202
- boot password 208
- device names 203
- GRUB shell 207
- grub.conf 207
- partition names 203
- problems 209
- uninstalling 219

H

HA
- Cold Standby 388
- DRBD 392
- Failover 388
- heartbeat 391
- High availability clusters 393
- Hot Standby 389
- Linux Virtual Server 393
- Load Balancing 389
- RAID 392
- rsync 392
- SPoF 388
- STONITH 389
- Warm Standby 388

HA (High Availability) 388–394

hard disks
- DMA 79
- parallel use of 129

hardware
- CD-ROM 65
- CD-ROM drives
  · ATAPI 126
- hard disk controller 69
- information 79
- Promise controller 157
- SCSI devices 127

hciconfig 346

hcitool 346

head 163

heartbeat 391

help
- info pages 240
- man pages 240
- X 277

High Availability . . . . *see* HA, *see* HA
High availability clusters 393
host names 84
Hot Standby 389
hotplug
    - agent 313
        · devices 313
        · interfaces 313
        · PCI 314
        · USB 314
    - blacklist 314
    - device names 315
    - error analysis 316
    - event recorder 316
    - events 312
    - log files 316
    - map files 314
    - modules
        · automatic loading of 314
    - network devices 315
    - PCI 315
    - whitelist 314
hwinfo 314

I

I18N 253
inetd 87
info pages 240
init 258
    - adding scripts 263
    - inittab 258
    - scripts 261–264
input method
    - CJK 252
insmod 229
installation
    - rescue system 183
    - VNC 115
installation support
    - 3D graphics cards and 285
installing
    - boot loader 118
    - GRUB 201
    - network, from 120–122
    - packages 167
    - PCMCIA 332
    - text mode 116–117
    - YaST 7–46
Internet
    - cinternet 607
    - dial-up 606–607
    - DSL 439
    - ISDN 441
    - kinternet 607
    - smpppd 606–607
    - TDSL 441
    - web servers . . . . *see* Apache
IP addresses
    - classes 410
    - dynamic assignment 518
    - IPv6 414
        · configuring 446
    - masquerading 655
    - private 412
IPsec
    - YaST 643
IrDA 341–343
    - configuring 342
    - starting 342
    - stopping 342
    - troubleshooting 343
iSCSI 140

K

Kerberos 667–674
    - administering 674–690
    - authenticators 668
    - clients
        · configuring 679–682
    - clock skew 682
    - clock synchronization 676
    - configuring
        · clients 679–682
    - credentials 668
    - installing 674–690
    - kadmin 684
    - KDC 675–679
        · administering 683
        · nsswitch.conf 675
        · resolv.conf 675
        · starting 679
    - keytab 685
    - LDAP and 688–690
    - logging 676
    - master key 677
    - PAM support 686
    - principals 668
        · creating 678
        · host 685
    - realms 674
        · creating 678
    - session key 669
    - SSH configuration 687
    - ticket-granting service 671
    - tickets 668, 671
kernels 226–233

- caches 241
- compiling 226, 231
- configuring 227–230
- daemon 230
- error messages 231
- installing 232–233
- limits 384
- modprobe.conf 230
- module loader 230
- modules 228–230
  · compiling 231
  · modprobe.conf 161
  · network cards 433
  · PCMCIA 324
- pcmcia 329
- problems 247
- sources 227
- System.map 233
- version 2.6 161

keyboard
- configuring 99
- layout 252
- mapping 252
  · compose 252
  · multikey 252

Kmod *see* kernels, module loader

L

L10N 253

language 99

laptops 323–334
- IrDA 341–343
- power management 351–363
- SCPM 334

LDAP 479–508
- access control 488
- ACLs 486
- adding data 489
- administering groups 505
- administering users 505
- configuration with YaST 494
- deleting data 494
- directory tree 482
- Kerberos and 688–690
- ldapadd 489
- ldapdelete 494
- ldapmodify 492
- ldapsearch 493
- modifying data 492
- searching data 493
- server
  · configuration with YaST 494
- server configuration 485
- YaST
  · modules 500
  · templates 500
- YaST LDAP client 498

LFS 383

license *see* GPL

Lightweight Directory Access Protocol . *see* LDAP

LILO 209
- boot sector, in 211
- configuring 118, 211
- floppy disk, on 210
- installing 215
- lilo.conf 211
- map files 210
- MBR 211
- memory test 215
- message files 210
- other systems 214
- parameters 213
- uninstalling 219
- updating 215–216

Linux
- networks and 405
- sharing files with another OS 584, 595

Linux Standard Base *see* LSB

Linux Virtual Server 393

linuxthreads 161

Load Balancing 389

Local Area Networks .. *see* networks, LANs

locale
- UTF-8 163

log files 238
- apache2 549, 560
- boot.msg 100, 356
- httpd 547, 549, 560
- installation 160
- log 91
- messages 100, 326, 329, 452, 659
- Squid 614, 616, 623
- Unison 571
- XFree86 285

logging
- login attempts 91
- logrotate 239
  · configuring 239

Logical Volume Manager *see* LVM

logrotate 238

LPAR installation
- IPL 34

LSB 236
- installing packages 166

lsmod 230
LVM
  - YaST 132

M

MacOS
  - sharing files 595
mail server
  - configuration 86
man pages 240
masquerading 655
  - configuring with SuSEfirewall2 . 657
Master Boot Record .... *see* MBR
MBR 198
  - LILO 211
memory
  - RAM 241
modems
  - YaST 437
modinfo 230
modprobe 229
modules
  - loading 109
  - parameters 109
monitor settings 70
mountd 517
mouse
  - configuring 80
multicast DNS 163

N

name servers .... *see* DNS
NAT .... *see* masquerading
Netatalk 595
  - AppleDouble 601
  - configuring 597
  - guest servers 598
  - permissions 599
  - printing 602
  - restrictions 597
  - starting 602
  - TCP/IP and 598
Network
  - SLP 448
network
  - configuring 83–85
  - routing 85
  - YaST 434
Network File System .... *see* NFS
Network Information Service ..... *see* NIS
networks 405
  - authentication
    · Kerberos 667–674
  - base network address 412
  - broadcast address 412
  - configuration files 426–432
  - configuring 424–447
    · IPv6 446
  - DHCP 518
  - DNS 413
  - integrating 433–447
  - LANs 433–447
  - localhost 412
  - netmasks 411
  - routing 410, 411
  - TCP/IP 406
NFS 514
  - clients 84, 514
  - exporting 515
  - importing 515
  - mounting 515
  - permissions 516
  - servers 84, 515
nfsd 517
NGPT 161
nice 163
NIS 509–513
  - clients 512
  - masters 509–511
  - slaves 509–511
notebooks .... *see* laptops
NPTL 161, 162
NSS 429
  - databases 430
NTFS file system 27
NTP 85

O

opd 348
OpenSSH .... *see* SSH
OS/2
  - sharing files 584

P

packages
  - compiling 173
  - compiling with build 175
  - installing 167
  - LSB 166
  - package manager 166
  - RPMs 166
  - uninstalling 167
  - verifying 167
packet filters .... *see* firewalls
PAM 395–402
pand 347

partitioning
- expert 128
- fdisk 220
- partition table 198
- swap 128
- YaST 131
partitions
- creating 17, 23, 24
- fstab 29
- LVM 25
- parameters 24
- RAID 25
- resizing Windows 25
- swap 25
- types 17
PCMCIA 324–334
- booting from 328
- card manager 325
- cardctl 333
- configuring 326–328
- driver assignment 331
- IDE 328
- installing with 332
- IrDA 341–343
- ISDN 327
- modems 327
- modules 325
- network cards 327, 445–446
- problems 328
- removing cards 326
- restarting 325
- SCSI 328
- starting
  · preventing 329
- systems 324
- troubleshooting 328
- utilities 333
permissions
- ACLs 706–717
- file permissions 240
phone exchange 443
Pluggable Authentication Modules *see* PAM
ports
- scanning 623
PostgreSQL
- updating 157
power management 351–363
- ACPI 351, 356–361, 364
- APM 351, 354–356, 364
- battery monitor 353
- charge level 365
- cpufreqd 363
- cpuspeed 363
- hibernation 353
- powersave 363–369
- standby 352
- suspend 352
- YaST 369
powersave 363
- configuration 364
Print system
- Update 288
print system *see* spool system
printer configuration
- update 166
printing 66–287
- configuring with YaST 66
- connection 67
- GDI printers 302
- Ghostscript driver 67
- IrDA 342
- network
  · troubleshooting 304
- port 67
- printer driver 67
- problems 69
- queues 67
- Samba 586
- switching the printing system 66
- troubleshooting 69
  · network 304
protocols
- SMB 584
proxies *see* Squid
- advantages 610
- caches 610
- transparent 620

R

RAID 392
- soft 138
rccoldplug 315
reiserfsck 747
removable media
- subfs 166
repairing systems 177
rescue system 11, 183, 247
- starting 248
- using 249
resolver library
- .local as top-level domain 163
RFCs 406
rmmod 229
routing 85, 410, 446–447
- masquerading 655

  - netmasks 411
  - routes 446
  - static 447

RPM 166–176
  - database
    · rebuilding 169, 173
  - dependencies 167
  - patches 169
  - queries 171
  - rpmnew 167
  - rpmorig 167
  - rpmsave 167
  - security 701
  - SRPMS 174
  - tools 176
  - uninstalling 169
  - updating 168
  - verify 172
  - verifying 167

rpmbuild 166

rsync 392, 566, 577

runlevels 98, 258–260
  - changing 260
  - default 98
  - editing in YaST 264
  - switching 98

S

Samba 584–595
  - clients 85, 585–586, 592–594
  - configuring 586–592
  - help 595
  - installing 586
  - login 590
  - names 585
  - NetBIOS 585
  - optimizing 594
  - permissions 589
  - printers 586
  - printing 594
  - security 589–590
  - servers 84, 586–592
  - shares 586, 587
  - SMB 584
  - starting 586
  - stopping 586
  - swat 590
  - TCP/IP and 584

SaX 70

SaX2
  - multihead 76

scanning
  - configuring 80
  - troubleshooting 81

SCPM 97, 334
  - configuring 336
  - managing profiles 336

screen
  - resolution 275

scripts
  - acpid_proxy 360
  - apmd_proxy 356
  - hotplug 326
  - init.d 258, 261–264, 432
    · boot 262
    · boot.local 262
    · boot.setup 262
    · halt 262
    · Netatalk 602
    · network 432
    · nfsserver 432, 516
    · portmap 432, 516
    · rc 260, 261, 263
    · sendmail 432
    · squid 614
    · xinetd 432
    · ypbind 432
    · ypserv 432
  - irda 342
  - mkinitrd 246
  - modify_resolvconf 242, 427
  - pcmcia 326, 446
    · network 332
  - SuSEconfig 266–268
    · disabling 268

SCSI devices
  - configuring 127
  - file names, assigning 127

sdptool 347

security 691–702
  - attacks 698–699
  - booting 692, 694
  - bugs and 695, 697
  - configuring 88–92
  - DNS 699
  - engineering 691
  - firewalls 92, 653
  - local 692–696
  - network 696–699
  - passwords 693
  - permissions 694–695
  - reporting problems 702
  - RPM signatures 701
  - Samba 589
  - serial terminals 692

- Squid 610
- SSH 661–667
- tcpd 702
- tips and tricks 700
- viruses 695
- worms 699
- X and 696

Server
- YOU 148

Service Location Protocol *see* SLP
Setting up installation servers 144
SGML
- directories 166

SLP 448
- Konqueror 450
- Registering services 448
- SLP browser 450
- slptool 450

SLP browser 450
slptool 450
SMB *see* Samba
software
- compiling 173
- installing 53–59
- removing 53–59

sort 163
sound
- configuring in YaST 82
- fonts 83

source
- compiling 173

spm 173
SPoF (Single Point of Failure) 388
spool system 287
Squid 610
- access controls 624
- ACLs 618
- Apache 623
- cachemgr.cgi 623, 625
- caches 610, 611
  - · size 613
- Calamaris 626, 627
- configuring 615
- CPU and 613
- directories 614
- DNS 615
- features 610
- firewalls and 621
- log files 614, 616, 623
- object status 611
- permissions 614, 618
- RAM and 613
- reports 626, 627
- security 610
- squidGuard 625
- starting 614
- statistics 623, 625
- stopping 614
- system requirements 612
- transparent proxies 620, 623
- troubleshooting 614
- uninstalling 615

SSH 661–667
- authentication mechanisms 665
- daemon 663
- key pairs 663, 665
- scp 662
- sftp 663
- ssh 661
- ssh-agent 666
- ssh-keygen 665
- sshd 663
- X and 666

start script
- boot.udev 321

start scripts
- rccoldplug 315

STONITH 389
subfs
- removable media 166

Subversion 574
subversion 565
support 100
system
- configuring 47–101
- freezes 118
- information 107
- language 99
- limiting resource use 240
- localizing 253
- optimizing 129
- rescuing 247
- resources
  - · viewing 331
- security 89
- services 87
- updating 61, 155–176

T

T-DSL *see* ADSL
tail 163
TCP/IP 406
- ICMP 407
- IGMP 407
- layer model 407
- packets 407, 409

  - services 406
  - TCP 406
  - UDP 407

thread packages
  - NPTL 162

time zones 99

TrueType *see* X, TrueType fonts

U

udev 317
  - automatisms 319
  - keys 320
  - mass storage 321
  - regular expressions 319
  - rules 318
  - start script 321
  - sysfs 320
  - udevinfo 320
  - YaST 322

ulimit 240
  - options 240

uninstalling
  - GRUB 219
  - LILO 219

Update
  - Print system 288
  - YOU server 148

updating 155–176
  - base system 158
  - checking passwd and group 156
  - log files 160
  - manually 158
  - online 50–53
  - patch CD 53
  - problems 156, 160
  - YaST 158

USB
  - booting from 199, 200

users
  - administering with YaST 88

UTF-8
  - encoding 163

V

virtual consoles
  - switching 98

virtual memory 25

VNC
  - installation 115

VPN
  - FreeS/WAN 643
  - FreeS/WAN client 646
  - FreeS/WAN-Server 643
  - Windows client 649
  - YaST 643

W

Wake on LAN (WOL) 371

Warm Standby 388

web servers
  - Apache *see* Apache
  - setting up 237

whois 414

Windows
  - NT boot manager 199
  - sharing files 584

wireless connections
  - Bluetooth 344

X

X 271
  - character sets 277
  - CID-keyed fonts 282
  - configuring 70
  - drivers 276
  - font systems 278
  - fonts 277
  - help 277
  - multihead 76
  - optimizing 272
  - SaX2 272
  - security 696
  - SSH and 666
  - TrueType fonts 277
  - virtual screen 275
  - X11 core fonts 281
  - xf86config 272
  - Xft 278
  - xft 277

X Window System *see* X

X.509 certificate 631

X.509 certification
  - Principles 630
  - Repository 633
  - revocation list 632
  - YaST 630

X.509 certification
  - Repository 634

XF86Config
  - color depth 275
  - Depth 275
  - Device 275
  - Display 275
  - Files 273
  - InputDevice 273
  - Modeline 275

- modelines 273
- Modes 273, 275
- Monitor 273, 275
- ServerFlags 273

Xft 278

XML
- directories 166

Y

YaST
- 3D 283
- backups 64, 93
- boot configuration 216
- boot mode 32
- CA module 634
- cable modem 436
- CD-ROM 65
- configuring 47–101
- Control Center 48
- DHCP 524
- disk creation 94
- disk space 18
- DMA 79
- DNS 84
- driver CDs 101
- DSL 439
- e-mail 85
- EVMS 97
- firewall 92
- graphical user interface 70–79
- graphics card 70
- graphics cards 73
- group administration 88
- hard disk controller 69
- hardware 65
- hardware information 79
- host name 84
- installation mode 14
- installation scope 30
- Installation server 144
- installation sources 49
- installation suggestion 15
- installing with 7–46
- IPsec 643
- ISDN 441
- keyboard layout 15, 99
- language 99
- language selection 12
- LDAP client 498
- LDAP server 494
- LVM 96, 132
- mail server 86
- manual installation 11
- memory test 11
- modem 437
- monitor settings 70
- mouse 16, 80
- ncurses 101
- network card 434
- network configuration 36, 83–85
- NFS client 84
- NFS server 84
- NIS client 42
- NIS clients 512
- NTP
  - · clients 85
- online update 50–53
- package dependencies 31
- package manager 54
- partitioning 17, 23, 131
- power management 369
- printing 66
- profile manager 97
- rc.config 98
- repairing systems 177
- rescue system 11
- root password 36
- routing 85
- runlevels 98, 264
- safe settings 11
- Samba
  - · clients 85
  - · server 84
- scanner 80
- SCPM 97
- security 88–92
- sendmail 85
- SLP browser 450
- soft RAID 138
- software 49–63
- software updates 39
- sound cards 82
- starting 8, 48
- support request 100
- sysconfig editor 98, 268
- system security 89
- system start-up 8
- T-DSL 441
- text mode $hyperpage116, 101 - -117$
  - · modules 104
  - · troubleshooting 117
- time zone 99
- updating 53, 61, 158
- user administration 88
- VPN 643

- Wake on LAN (WOL) 374
- X.509 certification 630
    · Changing standard values . . . 638
    · Client and server certificates . 636
    · Creating CRLs 640
    · Exporting CA objects as a file 641
    · Exporting CA objects to LDAP . . . 640
    · Exporting certificates on floppy . . 642
    · Importing general server certificates 642
    · Root CA 634
    · Sub-CA 636
- YOU 50–53
- YOU server 148
YOU 148
YP . . . . *see* NIS

Z

z/VM Installation
- IPL 34
ZIPL 222, 223