



SUSE LINUX

ADMINISTRATION GUIDE

10. Edition 2004

Copyright ©

This publication is intellectual property of Novell Inc.

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 GmbH, 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 trademarks. SUSE LINUX GmbH 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: Daniel Pisano, Tino Tanner

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, L^AT_EX

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

Contents

I	Installation	5
1	Installation with YaST	7
1.1	System Start-up for Installation	8
1.1.1	Possible Problems when Booting the System	8
1.1.2	Boot Options	9
1.2	The Boot Screen	10
1.3	Language Selection	12
1.4	Installation Mode	12
1.5	Installation Suggestion	13
1.5.1	Installation Mode	14
1.5.2	Keyboard Layout	14
1.5.3	Mouse	14
1.5.4	Partitioning	15
1.5.5	Expert Partitioning with YaST	19
1.5.6	Software	26
1.5.7	Boot Configuration	29
1.5.8	Time Zone	29
1.5.9	Language	29
1.5.10	Launching the Installation	30
1.6	Finishing the Installation	31

1.6.1	root Password	31
1.6.2	Network Configuration	32
1.6.3	Firewall Configuration	32
1.6.4	Testing the Internet Connection	33
1.6.5	Loading Software Updates	34
1.6.6	User Authentication	35
1.6.7	Configuring the Host as a NIS Client	35
1.6.8	Creating Local User Accounts	36
1.6.9	Release Notes	39
1.7	Hardware Configuration	39
1.8	Graphical Login	40
2	System Configuration with YaST	43
2.1	Starting YaST	44
2.2	The YaST Control Center	45
2.3	Software	46
2.3.1	Change Installation Source	46
2.3.2	YaST Online Update	46
2.3.3	Installing and Removing Software	48
2.3.4	System Update	56
2.4	Hardware	58
2.4.1	CD-ROM and DVD Drives	59
2.4.2	Printer	59
2.4.3	Hard Disk Controller	64
2.4.4	Graphics Card and Monitor (SaX2)	64
2.4.5	Hardware Information	74
2.4.6	IDE DMA Mode	74
2.4.7	Joystick	75
2.4.8	Keyboard Layout Selection	75
2.4.9	Mouse	75
2.4.10	Scanner	75

2.4.11	Sound	77
2.4.12	TV and Radio Cards	78
2.5	Network Devices	79
2.6	Network Services	79
2.6.1	Administration from a Remote Host	80
2.6.2	DHCP Server	80
2.6.3	Host Name and DNS	80
2.6.4	DNS Server	80
2.6.5	HTTP Server	80
2.6.6	LDAP Client	80
2.6.7	Mail Transfer Agent	81
2.6.8	NFS Client and NFS Server	81
2.6.9	NIS Client and NIS Server	82
2.6.10	NTP Client	82
2.6.11	Network Services (inetd)	82
2.6.12	Routing	83
2.6.13	Configuring Samba Servers and Clients	83
2.7	Security and Users	83
2.7.1	User Administration	83
2.7.2	Group Administration	84
2.7.3	Security Settings	85
2.7.4	Firewall	87
2.8	System	88
2.8.1	Backup Copy of the System Areas	88
2.8.2	Restoring the System	88
2.8.3	Creating a Boot, Rescue, or Module Disk	89
2.8.4	LVM	91
2.8.5	Partitioning	92
2.8.6	Profile Manager (SCPM)	92
2.8.7	Runlevel Editor	92

2.8.8	Sysconfig Editor	93
2.8.9	Time Zone Selection	93
2.8.10	Language Selection	93
2.9	Miscellaneous	93
2.9.1	Submitting a Support Request	93
2.9.2	Boot Log	94
2.9.3	System Log	94
2.9.4	Loading a Vendor's Driver CD	95
2.10	YaST in Text Mode (ncurses)	95
2.10.1	Navigation in Modules	96
2.10.2	Restriction of Key Combinations	98
2.10.3	Starting the Individual Modules	98
2.10.4	The YOU Module	98
2.11	Online Update from the Command Line	99
3	Special Installation Procedures	101
3.1	linuxrc	102
3.1.1	Main Menu	102
3.1.2	System Information	103
3.1.3	Loading Modules	105
3.1.4	Entering Parameters	105
3.1.5	Start Installation or System	107
3.1.6	Potential Problems	108
3.1.7	Passing Parameters to linuxrc	109
3.2	Installation with VNC	111
3.2.1	Preparing for the VNC Installation	111
3.2.2	Clients for the VNC Installation	112
3.3	Text-Based Installation with YaST	112
3.4	Starting SUSE LINUX	114
3.4.1	The Graphical SUSE Screen	114
3.4.2	Disabling the SUSE Screen	115

3.5	Special Installation Procedures	116
3.5.1	Installation without CD-ROM Support	116
3.5.2	Installation from a Network Source	116
3.6	Tips and Tricks	116
3.6.1	Creating a Boot Disk in DOS	116
3.6.2	Creating a Boot Disk in a UNIX-Type System	118
3.6.3	Booting from a Floppy Disk (SYSLINUX)	119
3.6.4	Supported CD-ROM Drives	120
3.7	ATAPI CD-ROM Hangs while Reading	120
3.8	Permanent Device Names for SCSI Devices	121
3.9	Partitioning for Experts	122
3.9.1	Size of the Swap Partition	122
3.9.2	Partitioning Proposals for Special Purposes	123
3.9.3	Optimization	123
3.10	LVM Configuration	126
3.10.1	Logical Volume Manager (LVM)	127
3.10.2	LVM Configuration with YaST	128
3.10.3	LVM — Partitioning	128
3.10.4	LVM — Configuring Physical Volumes	130
3.10.5	Logical Volumes	132
3.11	Soft RAID	135
3.11.1	Common RAID Levels	135
3.11.2	Soft RAID Configuration with YaST	136
3.11.3	Troubleshooting	136
3.11.4	For More Information	137

4	Updating the System and Package Management	139
4.1	Updating SUSE LINUX	140
4.1.1	Preparations	140
4.1.2	Possible Problems	141
4.1.3	Updating with YaST	141
4.1.4	Updating Individual Packages	142
4.2	Software Changes from Version to Version	142
4.2.1	From 8.0 to 8.1	142
4.2.2	From 8.1 to 8.2	144
4.2.3	From 8.2 to 9.0	145
4.2.4	From 9.0 to 9.1	145
4.2.5	From 9.1 to 9.2	152
4.3	RPM — the Package Manager	156
4.3.1	Verifying Package Authenticity	156
4.3.2	Managing Packages: Install, Update, and Uninstall	157
4.3.3	RPM and Patches	158
4.3.4	RPM Queries	160
4.3.5	Installing and Compiling Source Packages	163
4.3.6	Compiling RPM Packages with build	164
4.3.7	Tools for RPM Archives and the RPM Database	165
5	System Repair	167
5.1	Starting YaST System Repair	168
5.2	Automatic Repair	169
5.3	User-Defined Repair	170
5.4	Expert Tools	171
5.5	The SUSE Rescue System	172
5.5.1	Starting the Rescue System	172
5.5.2	Working with the Rescue System	173

II	System	177
6	32-Bit and 64-Bit Applications in a 64-Bit System Environment	179
6.1	Runtime Support	180
6.2	Software Development	181
6.3	Software Compilation on Biarch Platforms	181
6.4	Kernel Specifications	182
7	Booting and Boot Managers	185
7.1	The Boot Process	186
7.1.1	Master Boot Record	186
7.1.2	Boot Sectors	187
7.1.3	Booting DOS or Windows	187
7.2	Boot Management	187
7.3	Selecting a Boot Loader	188
7.4	Booting with GRUB	189
7.4.1	The GRUB Boot Menu	190
7.4.2	The File device.map	194
7.4.3	The File /etc/grub.conf	195
7.4.4	The GRUB Shell	196
7.4.5	Setting a Boot Password	196
7.5	Configuring the Boot Loader with YaST	198
7.5.1	The Main Window	198
7.5.2	Boot Loader Configuration Options	199
7.6	Uninstalling the Linux Boot Loader	201
7.7	Creating Boot CDs	201
7.8	Troubleshooting	202
7.9	For More Information	204

8	The Linux Kernel	205
8.1	Kernel Update	206
8.2	Kernel Sources	207
8.3	Kernel Configuration	207
8.3.1	Configuration on the Command Line	208
8.3.2	Configuration in Text Mode	208
8.3.3	Configuration in the X Window System	208
8.4	Kernel Modules	208
8.4.1	Hardware Detection with the Help of <code>hwinfo</code>	209
8.4.2	Handling Modules	209
8.4.3	<code>/etc/modprobe.conf</code>	210
8.4.4	<code>Kmod</code> — the Kernel Module Loader	210
8.5	Settings in the Kernel Configuration	211
8.6	Compiling the Kernel	211
8.7	Installing the Kernel	212
8.8	Cleaning Your Hard Disk after Compilation	213
9	Special Features of SUSE LINUX	215
9.1	Hints for Special Software Packages	216
9.1.1	The Package <code>bash</code> and <code>/etc/profile</code>	216
9.1.2	The <code>cron</code> Package	216
9.1.3	Log Files: Package <code>logrotate</code>	217
9.1.4	Man Pages	218
9.1.5	The Command <code>locate</code>	218
9.1.6	The Command <code>ulimit</code>	219
9.1.7	The <code>free</code> Command	220
9.1.8	The File <code>/etc/resolv.conf</code>	220
9.1.9	Settings for GNU Emacs	221
9.1.10	Brief Introduction to <code>vi</code>	222
9.2	Virtual Consoles	224
9.3	Keyboard Mapping	224
9.4	Language and Country-Specific Settings	225
9.4.1	Some Examples	226
9.4.2	Settings for Language Support	227

10	The SUSE LINUX Boot Concept	229
10.1	Booting with the Initial RAM Disk	230
10.1.1	Concept of the Initial RAM Disk	230
10.1.2	The Order of the Booting Process with initrd	231
10.1.3	Boot Loaders	232
10.1.4	Using initrd in SUSE	232
10.1.5	Possible Difficulties — Custom Kernels	234
10.1.6	Prospects	234
10.2	The init Program	235
10.3	Runlevels	235
10.4	Changing Runlevels	237
10.5	Init Scripts	238
10.5.1	Adding init Scripts	240
10.6	The YaST Runlevel Editor	241
10.7	SuSEconfig and /etc/sysconfig	243
10.8	The YaST sysconfig Editor	245
11	The X Window System	247
11.1	Optimizing the X Configuration	248
11.1.1	Screen Section	250
11.1.2	Device Section	251
11.1.3	Monitor and Modes Section	252
11.2	Installing and Configuring Fonts	253
11.2.1	Xft	254
11.2.2	X11 Core Fonts	257
11.2.3	CID-Keyed Fonts	258
11.3	OpenGL — 3D Configuration	258
11.3.1	Hardware Support	258
11.3.2	OpenGL Drivers	259
11.3.3	The Diagnosis Tool 3Ddiag	260
11.3.4	OpenGL Test Utilities	260
11.3.5	Troubleshooting	260
11.3.6	Installation Support	260
11.3.7	Additional Online Documentation	261

12 Printer Operation	263
12.1 Preparation and Other Considerations	264
12.2 Methods and Protocols for Connecting Printers	265
12.3 Installing the Software	266
12.4 Configuring the Printer	267
12.4.1 Local Printers	267
12.4.2 Network Printers	267
12.4.3 Configuration Tasks	268
12.5 Special Features in SUSE LINUX	271
12.5.1 CUPS Server and Firewall	271
12.5.2 Administrator for CUPS Web Front-End	272
12.5.3 Changes in the CUPS Print Service (cupsd)	272
12.5.4 PPD Files in Various Packages	274
12.6 Troubleshooting	276
12.6.1 Printers without Standard Printer Language Support	276
12.6.2 No Suitable PPD File Available for a PostScript Printer	277
12.6.3 Parallel Ports	277
12.6.4 Network Printer Connections	278
12.6.5 Defective Printouts without Error Message	281
12.6.6 Disabled Queues	281
12.6.7 CUPS Browsing: Deleting Print Jobs	281
12.6.8 Defective Print Jobs and Data Transfer Errors	282
12.6.9 Debugging the CUPS Print System	282
13 Mobile Computing with Linux	285
13.1 Laptops	286
13.1.1 Special Properties of Laptop Hardware	286
13.1.2 Power Conservation	286
13.1.3 Integration in Changing Operating Environments	287
13.1.4 Software Options	288
13.1.5 Data Security	291
13.2 Mobile Hardware	292
13.3 Cellular Phones and PDAs	293
13.4 For More Information	293

14 PCMCIA	295
14.1 Hardware	296
14.2 Software	296
14.2.1 Base Modules	296
14.2.2 Card Manager	297
14.3 Configuration	298
14.3.1 Network Cards	298
14.3.2 ISDN	299
14.3.3 Modem	299
14.3.4 SCSI and IDE	299
14.4 Utilities	300
14.5 Troubleshooting	300
14.5.1 PCMCIA Base System Does Not Work	300
14.5.2 PCMCIA Card Does Not Work Properly	301
14.6 For More Information	303
15 System Configuration Profile Management	305
15.1 Terminology	306
15.2 Configuration	307
15.2.1 Starting SCPM and Defining Resource Groups	307
15.2.2 Creating and Managing Profiles	308
15.2.3 Switching Configuration Profiles	308
15.2.4 Advanced Profile Settings	309
15.2.5 Selecting a Profile When Booting the System	310
15.3 Troubleshooting	310
15.3.1 Termination during the Switch Process	310
15.3.2 Changing the Resource Group Configuration	311
15.4 For More Information	311

16 Power Management	313
16.1 Power Saving Functions	314
16.2 APM	315
16.3 ACPI	316
16.3.1 ACPI in Action	317
16.3.2 Controlling the CPU Performance	320
16.3.3 ACPI Tools	321
16.3.4 Troubleshooting	321
16.4 Rest for the Hard Disk	323
16.5 The powersave Package	324
16.5.1 Configuring the powersave Package	325
16.5.2 Configuring APM and ACPI	327
16.5.3 Additional ACPI Features	329
16.5.4 Troubleshooting	329
16.6 The YaST Power Management Module	332
17 Wireless Communication	337
17.1 Wireless LAN	338
17.1.1 Hardware	338
17.1.2 Function	339
17.1.3 Configuration with YaST	341
17.1.4 Utilities	344
17.1.5 Tips and Tricks for Setting up a WLAN	344
17.1.6 Troubleshooting	345
17.1.7 For More Information	346
17.2 Bluetooth	346
17.2.1 Basics	346
17.2.2 Configuration	347
17.2.3 System Components and Utilities	350
17.2.4 Graphical Applications	351
17.2.5 Examples	352

17.2.6	Troubleshooting	353
17.2.7	For More Information	355
17.3	Infrared Data Transmission	355
17.3.1	Software	356
17.3.2	Configuration	356
17.3.3	Usage	356
17.3.4	Troubleshooting	357
18	The Hotplug System	359
18.1	Devices and Interfaces	360
18.2	Hotplug Events	361
18.3	Hotplug Agents	362
18.3.1	Activating Network Interfaces	363
18.3.2	Activating Storage Devices	363
18.4	Automatic Module Loading	364
18.5	Hotplug with PCI	365
18.6	The Boot Scripts Coldplug and Hotplug	365
18.7	Error Analysis	366
18.7.1	Log Files	366
18.7.2	Boot Problems	366
18.7.3	The Event Recorder	366
18.7.4	Excessive System Load or Too Slow While Booting	367
19	Dynamic Device Nodes with udev	369
19.1	Creating Rules	370
19.2	Automization with NAME and SYMLINK	371
19.3	Regular Expressions in Keys	371
19.4	Key Selection	372
19.5	Consistent Names for Mass Storage Devices	373

20 File Systems in Linux	375
20.1 Terminology	376
20.2 Major File Systems in Linux	376
20.2.1 Ext2	377
20.2.2 Ext3	378
20.2.3 Converting an Ext2 File System into Ext3	379
20.2.4 ReiserFS	379
20.2.5 JFS	380
20.2.6 XFS	381
20.3 Some Other Supported File Systems	382
20.4 Large File Support in Linux	383
20.5 For More Information	384
21 PAM — Pluggable Authentication Modules	385
21.1 Structure of a PAM Configuration File	386
21.2 The PAM Configuration of sshd	388
21.3 Configuration of PAM Modules	389
21.3.1 pam_unix2.conf	389
21.3.2 pam_env.conf	390
21.3.3 pam_pwcheck.conf	391
21.3.4 limits.conf	391
21.4 For More Information	391
III Services	393
22 Linux in the Network	395
22.1 Introduction to TCP/IP	396
22.1.1 Layer Model	397
22.1.2 IP Addresses and Routing	400
22.1.3 Domain Name System	403
22.2 IPv6 — The Next Generation Internet	404

22.2.1	Advantages of IPv6	405
22.2.2	The IPv6 Address System	406
22.2.3	Coexistence of IPv4 and IPv6	411
22.2.4	For More Information	412
22.3	Manual Network Configuration	412
22.3.1	Configuration Files	416
22.3.2	Start-up Scripts	422
22.4	Network Integration	422
22.4.1	Requirements	423
22.4.2	Configuring the Network Card with YaST	423
22.4.3	Modem	426
22.4.4	DSL	428
22.4.5	ISDN	430
22.4.6	Hotplug and PCMCIA	433
22.4.7	Configuring IPv6	434
22.5	Routing in SUSE LINUX	434
22.6	SLP Services in the Network	435
22.6.1	SLP Support in SUSE LINUX	435
22.6.2	For More information	438
22.7	DNS — Domain Name System	438
22.7.1	Starting the Name Server BIND	438
22.7.2	The Configuration File /etc/named.conf	440
22.7.3	Important Configuration Options	441
22.7.4	The Configuration Section Logging	442
22.7.5	Zone Entry Structure	443
22.7.6	Structure of Zone Files	444
22.7.7	Secure Transactions	447
22.7.8	Dynamic Update of Zone Data	449
22.7.9	DNSSEC	449
22.7.10	Configuration with YaST	450

22.7.11	For More Information	458
22.8	NIS — Network Information Service	459
22.8.1	NIS Master and Slave Servers	460
22.8.2	The NIS Client Module of YaST	463
22.9	LDAP — A Directory Service	464
22.9.1	LDAP versus NIS	465
22.9.2	Structure of an LDAP Directory Tree	466
22.9.3	Server Configuration with slapd.conf	469
22.9.4	Data Handling in the LDAP Directory	473
22.9.5	The YaST LDAP Client	477
22.9.6	For More Information	485
22.10	NFS — Shared File Systems	486
22.10.1	Importing File Systems with YaST	486
22.10.2	Importing File Systems Manually	486
22.10.3	Exporting File Systems with YaST	487
22.10.4	Exporting File Systems Manually	488
22.11	DHCP	491
22.11.1	The DHCP Protocol	491
22.11.2	DHCP Software Packages	492
22.11.3	The DHCP Server dhcpd	492
22.11.4	Hosts with Fixed IP Addresses	494
22.11.5	The SUSE LINUX Version	495
22.11.6	DHCP Configuration with YaST	496
22.11.7	For More Information	500
22.12	Time Synchronization with xntp	500
22.12.1	Configuration in the Network	501
22.12.2	Setting up a Local Reference Clock	502
22.12.3	NTP Client Configuration with YaST	502

23	The Apache Web Server	507
23.1	Basics	508
23.1.1	Web Server	508
23.1.2	HTTP	508
23.1.3	URLs	508
23.1.4	Automatic Display of a Default Page	509
23.2	Setting up the HTTP Server with YaST	509
23.3	Apache Modules	510
23.4	Threads	511
23.5	Installation	511
23.5.1	Selecting Packages in YaST	511
23.5.2	Activating Apache	512
23.5.3	Modules for Active Contents	512
23.5.4	Other Recommended Packages	512
23.5.5	Installing Modules with apxs	512
23.6	Configuration	513
23.6.1	Configuration with SuSEconfig	513
23.6.2	Manual Configuration	514
23.7	Using Apache	518
23.8	Active Contents	518
23.8.1	Server Side Includes	520
23.8.2	Common Gateway Interface	520
23.8.3	GET and POST	520
23.8.4	Languages for CGI	520
23.8.5	Generating Active Contents with Modules	521
23.8.6	mod_perl	521
23.8.7	mod_php4	523
23.8.8	mod_python	524
23.8.9	mod_ruby	524
23.9	Virtual Hosts	524

23.9.1	Name-Based Virtual Hosts	525
23.9.2	IP-Based Virtual Hosts	526
23.9.3	Multiple Instances of Apache	527
23.10	Security	527
23.10.1	Minimizing the Risk	527
23.10.2	Access Permissions	528
23.10.3	Staying Updated	528
23.11	Troubleshooting	528
23.12	For More Information	529
23.12.1	Apache	529
23.12.2	CGI	529
23.12.3	Security	530
23.12.4	Additional Sources	530
24	File Synchronization	531
24.1	Available Data Synchronization Software	532
24.1.1	Unison	532
24.1.2	CVS	533
24.1.3	subversion	533
24.1.4	mailsync	533
24.1.5	rsync	534
24.2	Determining Factors for Selecting a Program	534
24.2.1	Client-Server versus Peer-to-Peer	534
24.2.2	Portability	534
24.2.3	Interactive versus Automatic	534
24.2.4	Conflicts: Incidence and Solution	535
24.2.5	Selecting and Adding Files	535
24.2.6	History	535
24.2.7	Data Volume and Hard Disk Requirements	536
24.2.8	GUI	536
24.2.9	User Friendliness	536

24.2.10	Security against Attacks	536
24.2.11	Protection against Data Loss	537
24.3	Introduction to Unison	538
24.3.1	Requirements	538
24.3.2	Using Unison	538
24.3.3	For More Information	539
24.4	Introduction to CVS	539
24.4.1	Configuring a CVS Server	540
24.4.2	Using CVS	540
24.4.3	For More Information	542
24.5	Introduction to Subversion	542
24.5.1	Installing a Subversion Server	542
24.5.2	Usage and Operation	543
24.5.3	For More Information	545
24.6	Introduction to rsync	545
24.6.1	Configuration and Operation	545
24.6.2	For More Information	547
24.7	Introduction to mailsync	547
24.7.1	Configuration and Use	547
24.7.2	Possible Problems	549
24.7.3	For More Information	550
25	Samba	551
25.1	Configuring the Server	553
25.1.1	The global Section	554
25.1.2	Shares	555
25.1.3	Security Levels	556
25.2	Samba as Login Server	557
25.3	Configuring a Samba Server with YaST	559
25.4	Configuring Clients	560
25.4.1	Configuring a Samba Client with YaST	560
25.4.2	Windows 9x and ME	561
25.5	Optimization	562

26 Internet	563
26.1 smpppd as Dial-up Assistant	564
26.1.1 Program Components for the Internet Dial-Up	564
26.1.2 Configuring smpppd	564
26.1.3 Remote Use of kinternet, cinternet, and qinternet	565
26.2 Configuring an ADSL or T-DSL Connection	566
26.2.1 Default Configuration	566
26.2.2 DSL Connection by Dial-on-Demand	566
26.3 Proxy Server: Squid	567
26.3.1 Squid as Proxy Cache	568
26.3.2 Some Facts about Proxy Caches	568
26.3.3 System Requirements	570
26.3.4 Starting Squid	571
26.3.5 The Configuration File /etc/squid/squid.conf	573
26.3.6 Configuring a Transparent Proxy	578
26.3.7 cachemgr.cgi	581
26.3.8 squidGuard	582
26.3.9 Cache Report Generation with Calamaris	584
26.3.10 For More Information	585
27 Security in Linux	587
27.1 Masquerading and Firewalls	588
27.1.1 Packet Filtering with iptables	588
27.1.2 Masquerading Basics	590
27.1.3 Firewalling Basics	591
27.1.4 SuSEfirewall2	592
27.1.5 For More Information	597
27.2 SSH: Secure Network Operations	598
27.2.1 The OpenSSH Package	598
27.2.2 The ssh Program	598
27.2.3 scp — Secure Copy	599

27.2.4	sftp — Secure File Transfer	599
27.2.5	The SSH Daemon (sshd) — Server-Side	600
27.2.6	SSH Authentication Mechanisms	601
27.2.7	X, Authentication, and Forwarding Mechanisms	602
27.3	Encrypting Partitions and Files	603
27.3.1	Application Scenarios	603
27.3.2	Setting up a Crypto File System with YaST	603
27.3.3	Encrypting the Content of Removable Media	605
27.4	Security and Confidentiality	605
27.4.1	Local Security and Network Security	606
27.4.2	Some General Security Tips and Tricks	615
27.4.3	Using the Central Security Reporting Address	617

IV Administration 619

28	Access Control Lists in Linux	621
28.1	Advantages of ACLs	622
28.2	Definitions	623
28.3	Handling ACLs	623
28.3.1	Structure of ACL Entries	624
28.3.2	ACL Entries and File Mode Permission Bits	625
28.3.3	A Directory with Access ACL	626
28.3.4	A Directory with a Default ACL	629
28.3.5	The ACL Check Algorithm	632
28.4	ACL Support in Applications	632

29 System Monitoring Utilities	635
29.1 List of Open Files: lsof	636
29.2 User Accessing Files: fuser	637
29.3 File Properties: stat	638
29.4 Processes: top	638
29.5 Process List: ps	639
29.6 Process Tree: pstree	641
29.7 Who Is Doing What: w	642
29.8 Memory Usage: free	642
29.9 Kernel Ring Buffer: dmesg	643
29.10 File Systems and Their Usage: mount, df, and du	643
29.11 The /proc File System	644
29.12 procinfo	646
29.13 PCI Resources: lspci	647
29.14 System Calls of a Program Run: strace	648
29.15 Library Calls of a Program Run: ltrace	650
29.16 Specifying the Required Library: ldd	650
29.17 Additional Information about ELF Binaries	651
29.18 Interprocess Communication: ipc	651
29.19 Time Measurement with time	652
V Appendix	653
A Information Sources and Documentation	655
B Manual Page of reiserfsck	659
C Manual Page of e2fsck	663
D The GNU General Public License	669
Glossary	677
Bibliography	689

Welcome

Congratulations for your new Linux operating system and thank you for selecting SUSE LINUX 9.2. By purchasing this version, you have gained the right to benefit from installation support by phone and e-mail. To make use of this service, activate your support authorization in the SUSE LINUX Portal (<http://portal.suse.com>) with the help of the code printed on the CD case.

To make sure your system is always in a secure and up-to-date state, we recommend regular updates with the comfortable *YaST Online Update*. Moreover, we offer a free e-newsletter featuring security-relevant information and tips and tricks for SUSE LINUX. Simply subscribe by entering your e-mail address at <http://www.suse.com/us/private/newsletter.html>.

The SUSE LINUX *Administration Guide* provides background information about the way your SUSE LINUX system operates. This manual introduces you to Linux system administration basics, such as file systems, kernels, boot processes, and the configuration of the Apache web server. The SUSE LINUX *Administration Guide* has five major categories:

Installation System installation and configuration with YaST, details about special installation types, LVM, RAID, updates, and system recovery.

System Special features of SUSE LINUX, details about the kernel, boot concept, and init process, configuration of the boot loader and the X Window System, printing, and mobile computing in Linux.

Services Integration in heterogeneous networks, configuration of the Apache web server, file synchronization, and security aspects.

Administration File system ACLs and important system monitoring tools.

Appendix Important sources of information about Linux.

The digital versions of the SUSE LINUX manuals are located in the directory `file:///usr/share/doc/manual/`.

Changes in the Administration Guide

The documentation of the previous version (SUSE LINUX 9.1) has been modified as follows:

- The complete installation and configuration with YaST has been transferred from the *User Guide* to the first two chapters of this manual: *Installation with YaST* on page 7 and *System Configuration with YaST* on page 43.
- The chapter about system recovery has also been adapted from the *User Guide*. See *System Repair* on page 167.
- *Booting and Boot Managers* on page 185 has been revised and a description of the YaST module has been added.
- The printing chapter has been updated and restructured. See *Printer Operation* on page 263.
- The about Linux on mobile devices has been rewritten. See *Mobile Computing with Linux* on page 285. The information for SCPM, PCMCIA, and wireless communication have been revised and moved to independent chapters. See *System Configuration Profile Management* on page 305, *PCMCIA* on page 295, and *Wireless Communication* on page 337.
- *The Hotplug System* on page 359 is entirely new.
- *Dynamic Device Nodes with udev* on page 369 is also new.
- Information about PAM has been added in *PAM — Pluggable Authentication Modules* on page 385.
- The networking chapter contains a new section about SLP. See *SLP Services in the Network* on page 435.

Typographical Conventions

The following typographical conventions are used in this manual:

- YaST: programs
- `/etc/passwd`: file or directory names
- `<placeholder>`: replace `<placeholder>` with the actual value
- `PATH`: the environment variable `PATH`
- `ls`: commands
- `--help`: options and parameters
- `user`: users
- `(Alt)`: a key to press
- 'File': menu items, buttons
- "Process killed": system messages
- ► **x86, AMD64**
This section is only relevant for the specified architectures. The arrows mark the beginning and the end of the text block. ◀

Acknowledgment

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

Have a lot of fun!

Your SUSE Team

Part I

Installation

Installation with YaST

This chapter systematically guides you through the installation of the SUSE LINUX system with the SUSE LINUX system assistant YaST. The description of the preparation of the installation process is accompanied by background information to assist you in making the right decisions in the individual configuration stages.

1.1	System Start-up for Installation	8
1.2	The Boot Screen	10
1.3	Language Selection	12
1.4	Installation Mode	12
1.5	Installation Suggestion	13
1.6	Finishing the Installation	31
1.7	Hardware Configuration	39
1.8	Graphical Login	40

1.1 System Start-up for Installation

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.

1.1.1 Possible Problems when Booting the System

Problems can arise booting from the CD or DVD if you have older or unsupported hardware. Your CD-ROM drive might not be able to read the boot image on CD 1. In this case, use CD 2 to boot the system. CD 2 contains a conventional 2.88 MB boot image that can be read even by unsupported drives and allows you to perform the installation over the network.

The boot sequence might not be set correctly in the BIOS. Instructions for how to change the BIOS settings are available in the documentation of your motherboard. Basic instructions are provided in the following paragraphs.

The BIOS is the software that enables the very basic functions of a computer. Motherboard vendors provide a BIOS specifically made for their hardware. Normally, the BIOS setup can only be accessed at a specific time — when the machine is booting. During this initialization phase, the machine performs a number of diagnostic hardware tests. One of them is a memory check, indicated by a memory counter. When the counter appears, look for a line, usually below the counter or somewhere at the bottom, mentioning the key to press to access the BIOS setup. Usually the key to press is **(Del)**, **(F1)**, or **(Esc)**. Press this key until the BIOS setup screen appears.

Note

Keyboard Layout in the BIOS

The BIOS configuration often uses a 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 **(PgUp)** or **(PgDown)** 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). Then 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.

1.1.2 Boot Options

Boot options other than CD or DVD exist and can be used if problems arise booting from CD or DVD. These options are described in Table 1.1.

Table 1.1: Boot Options

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

1.2 The Boot Screen

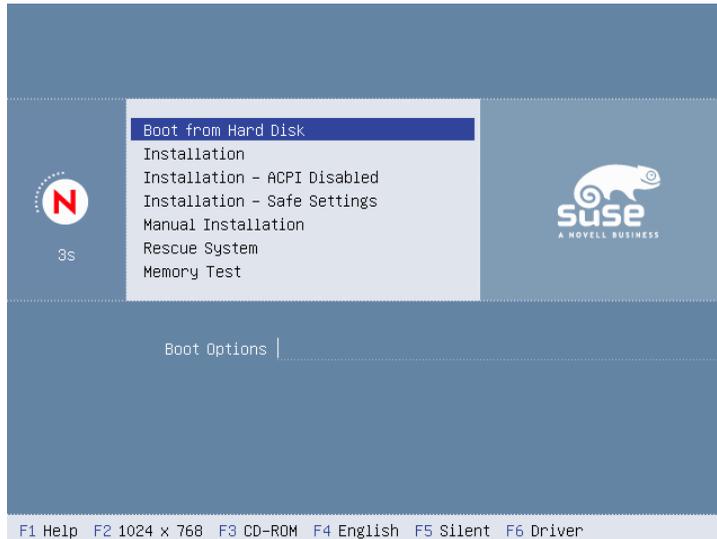


Figure 1.1: The Boot Screen

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

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

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 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.

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

- ⓕ1 Context-sensitive help for the active element of the boot screen.
- ⓕ2 Selection of various graphical display modes for the installation. The text mode can be selected if the graphical installation causes problems.
- ⓕ3 Normally, the installation is performed from the inserted installation medium. Other sources, like FTP or NFS servers, can be selected here. If the installation is carried out in a network with an SLP server, one of the installation sources available on the server can be selected with this option. Information about *SLP* is available in *SLP Services in the Network* on page 435.
- ⓕ4 Select the display language for the installation.
- ⓕ5 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 detailed information, select 'Verbose'.
- ⓕ6 Use this 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. If you enabled 'Native' or 'Verbose', messages and copyright notices scroll by and, at the end of the loading process, the YaST installation program starts. After a few more seconds, the screen should display the graphical interface.

The actual installation of SUSE LINUX begins at this point. All YaST screens have a common layout. All buttons, entry fields, and lists can be accessed with the mouse or the keyboard. If your mouse pointer does not move, the mouse has not been autodetected. In this case, use the keyboard for the time being.

1.3 Language Selection

YaST and SUSE LINUX in general can be configured to use different languages according to your needs. The language selected here is also used for the keyboard layout. In addition, YaST uses the language setting to guess a time zone for the system clock. These settings can be modified later. If your mouse does not work, select the language with the arrow keys and press (Tab) until 'Accept' is highlighted. Then press (Enter) to confirm your language selection.

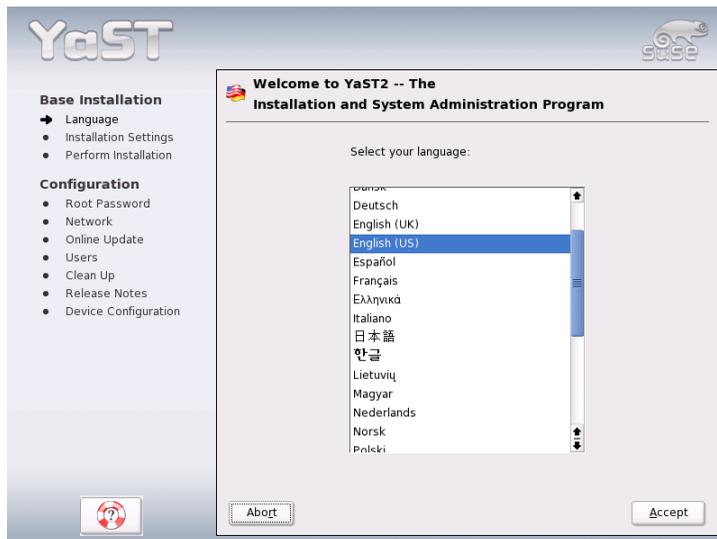


Figure 1.2: Selecting the Language

1.4 Installation Mode

Select 'New installation' or 'Update an existing system'. Updating is only possible if a SUSE LINUX system is already installed. If this is the case, the installed system can be booted with 'Boot installed system'. If the installed system fails to boot, perhaps because some important system configuration has been corrupted,

you can try to make the system bootable again with 'Repair installed system'. If no SUSE LINUX system is installed, you can only perform the new installation. See Figure 1.3.

The following sections describe the procedure of installing a new system. Detailed instructions for a system update can be found in *System Update* on page 56. A description of the system repair options can be found in *System Repair* on page 167.

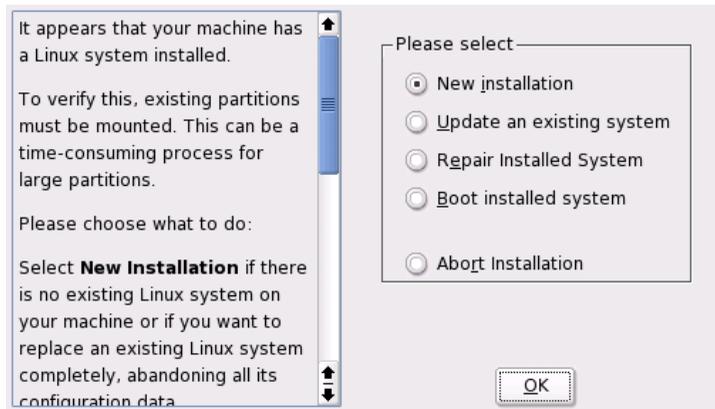


Figure 1.3: Selecting the Installation Mode

1.5 Installation Suggestion

After hardware detection, the suggestion window, shown in Figure 1.4 on the following page, 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.

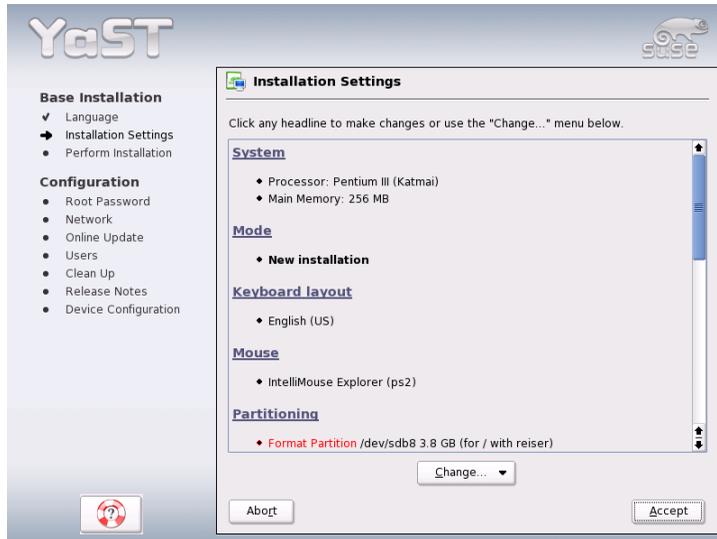


Figure 1.4: Suggestion Window

1.5.1 Installation Mode

Use this to change the previously selected installation mode. The options are the same as those described in *Installation Mode* on page 12.

1.5.2 Keyboard Layout

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.

1.5.3 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 1.5 on the facing page.

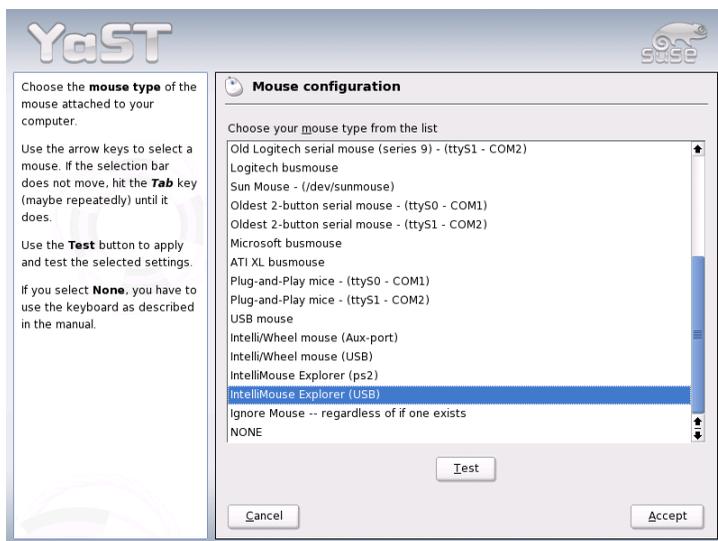


Figure 1.5: Selecting the Mouse Type

To select the mouse type, use \uparrow and \downarrow . Consult your mouse documentation for information about the mouse type. After selecting a mouse type, use $\text{Alt}+\text{T}$ to test whether the device works correctly without making the selection permanent. If the mouse does not behave as expected, use the keyboard to select another type and test again. Use Tab and Enter to make the current selection permanent.

1.5.4 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

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

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

If you need more than four partitions, create an extended partition as the fourth partition or earlier. This extended partition should span the entire remaining free cylinder range. Then create multiple logical partitions within the extended partition. The maximum number of logical partitions is fifteen on SCSI, SATA, and Firewire disks and sixty-three 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.

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: 2.5 GB This includes a modern desktop environment, like KDE or GNOME, and also provides enough space for large application suites, such as OpenOffice.org and Netscape or Mozilla.

The partitions to create depend on the available space. The following are some basic partitioning guidelines:

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). If you do not want to have separate partitions for these directories, add the suggested disk space to the root partition. The rest of the available space can be used for `/home`.

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 one 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, either create a separate partition for `/opt` or make the root partition large enough. KDE and GNOME are also installed in `/opt`.

Partitioning with YaST

When you select the partitioning item in the suggestion window for the first time, the YaST partitioning dialog displays the partition settings as currently proposed. Accept these current settings as they are 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 proposal as is'. If you select 'Base partition setup on this proposal', the 'Expert Partitioner' opens. It allows tweaking the partition setup in every detail. This dialog is explained in *Expert Partitioning with YaST* on page 19. The original setup as proposed by YaST is offered there as a starting point.

Selecting 'Create custom partition setup' opens the dialog as shown in Figure 1.7 on page 19. 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 was found on the disk, you are asked whether to delete or resize the partition. Before doing so, read *Resizing a Windows Partition* on page 22. If desired, go to the 'Expert Partitioner' dialog to create a custom partition setup at this point (see *Expert Partitioning with YaST* on page 19).

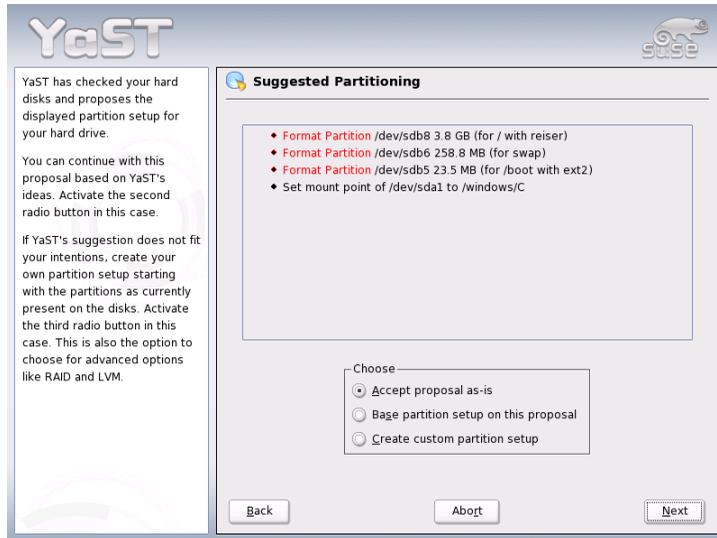


Figure 1.6: Editing the Partitioning Setup

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 changes the software selection. The proposal dialog displays 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.

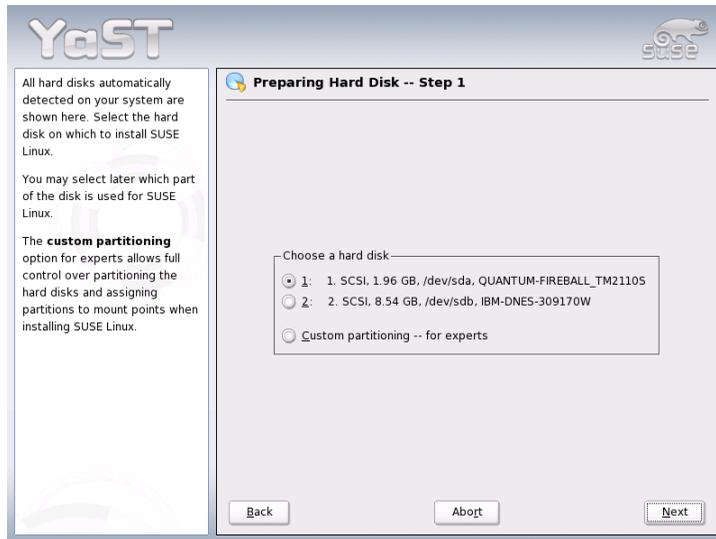


Figure 1.7: Selecting the Hard Disk

1.5.5 Expert Partitioning with YaST

With the expert dialog, shown in Figure 1.8 on the next page, manually modify the partitioning of one or several hard disks. Partitions can be added, deleted, and 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.

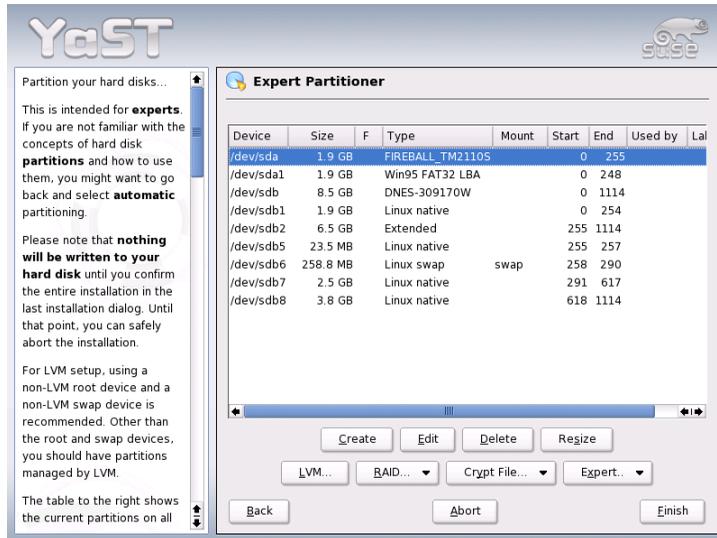


Figure 1.8: The YaST Partitioner in Expert Mode

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 *Partition Types* on page 15).

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. 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', and 'Linux RAID'. For details on LVM and RAID, refer to *LVM Configuration* on page 126 and *Soft RAID* on page 135.

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'. Refer to *File Systems in Linux* on page 375 for details on the various file systems.

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

File System Options Set various parameters for the selected file system here. Depending on the file system used, various options are offered for experts.

Encrypt File System If you activate the encryption, all data is written to the hard disk in encrypted form. This increases the security of sensitive data, but slightly reduces the system speed, because the encryption takes some time. More information about the encryption of file systems is provided in *Encrypting Partitions and Files* on page 603.

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

Mount Point Specifies the directory at which the partition should be mounted in the file system tree. Select from various YaST proposals or specify any other name.

3. Select 'Next' to activate the partition.

If you partition manually, create a swap partition of at least 256 MB. The swap partition is used to free the main memory of data that is not used at the present moment. This keeps the main memory free for the most frequently-used important data.

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 partition, it presents a dialog in which to choose one of these two options.

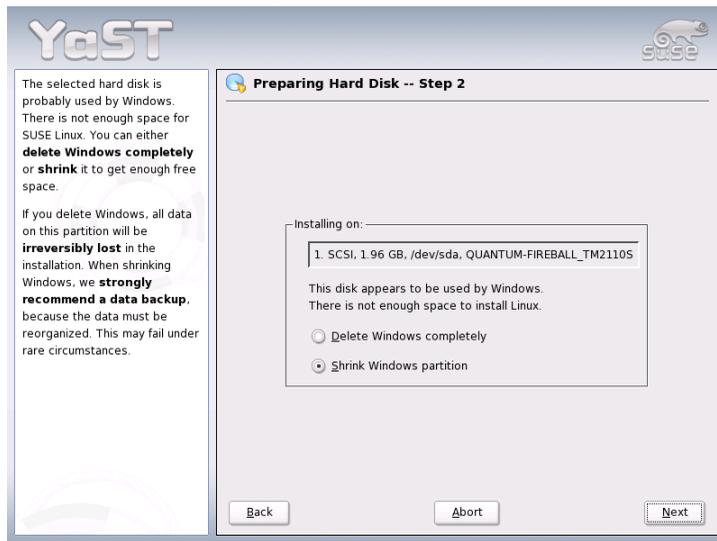


Figure 1.9: Possible Options for Windows Partitions

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 so 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 disable these Windows optimizations for the time being and reenale 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.

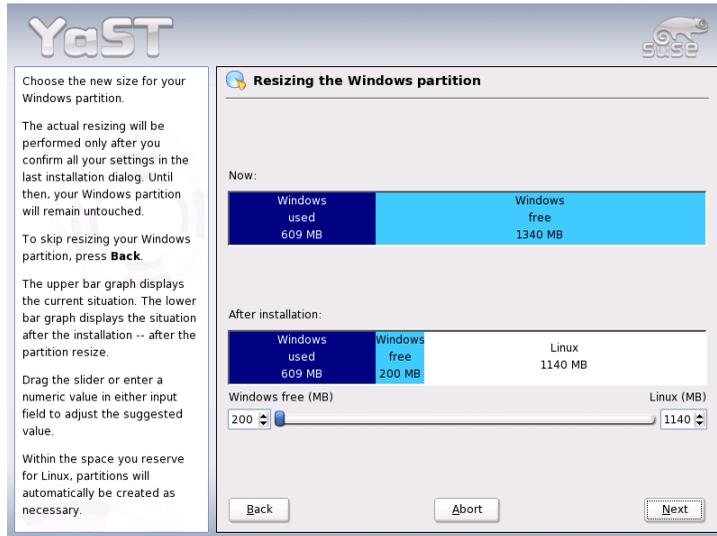


Figure 1.10: Resizing the Windows Partition

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. See Figure 1.10. Accept the proposed settings or use the slider to change the partition sizing (within certain limits).

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 only be read from Linux. This means 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, 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
/dev/dasda1  /data3  auto    noauto,user 0 0
```

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

Detailed background information and tips for partitioning are provided in *Partitioning for Experts* on page 122.

1.5.6 Software

SUSE LINUX contains a number of software packages for various application purposes. Because 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. No office programs are installed.

Default System (with GNOME and office suite)

This is one of the largest of the predefined systems. It includes the GNOME desktop together with most of the GNOME programs and the office programs.

Default System (with KDE and office suite)

This system includes the KDE desktop together with most of the KDE programs and the office programs. This is the most suitable system type for normal stand-alone machines.

Click 'Software' 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 1.11 on the next page.

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.

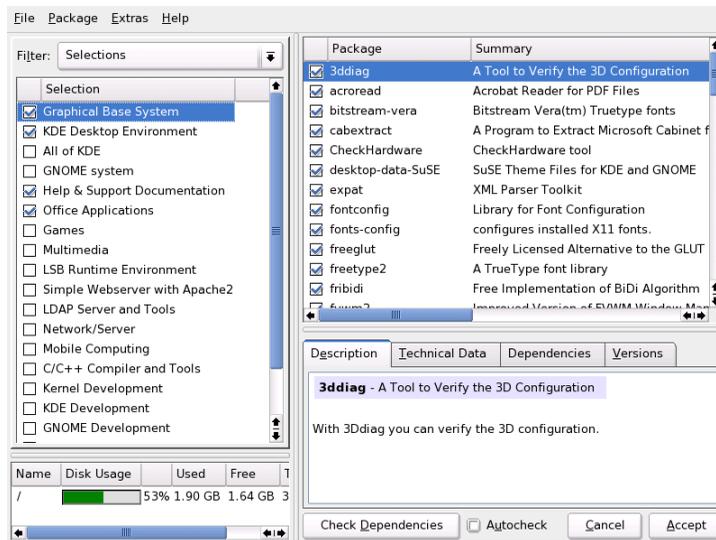


Figure 1.11: Installing and Removing Software with the YaST Package Manager

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 table column furthest to the left 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. To learn more about them, read the detailed description of this module in *Installing and Removing Software* on page 48.

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 *Installing and Removing Software* on page 48.

Package Dependencies and Conflicts

You cannot simply install any combination of software packages. 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 *Installing and Removing Software* on page 48, which provides detailed information about the operation of the package manager and a brief summary of the software organization in Linux.

Caution

The software preselected for installation is based on long-standing experience and is usually suitable for the needs of most newcomers and advanced home users. In general, there is no need to change anything here. However, if you decide to select or deselect any packages, you should be aware of the consequences. In particular, observe any warnings and avoid deselecting any packages of the base system.

Caution

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.

1.5.7 Boot Configuration

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

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 other 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 *Configuring the Boot Loader with YaST* on page 198.

Note

The boot method should only be changed by experienced computer users.

Note

1.5.8 Time Zone

In this dialog, shown in Figure 1.12 on the following 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 saving time and back automatically.

1.5.9 Language

The language was selected at the beginning of the installation as described in *Language Selection* 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.

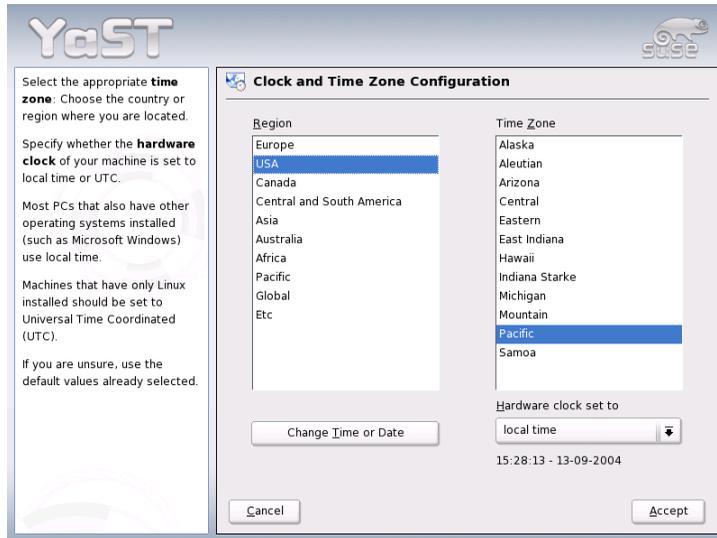


Figure 1.12: Selecting the Time Zone

- 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.

1.5.10 Launching the Installation

After specifying all 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.

1.6 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. You can also configure a name server for centralized user administration in a local network. Finally, you can configure the hardware devices connected to the machine.

1.6.1 `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, as shown in Figure 1.13 on the following page. Do not 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

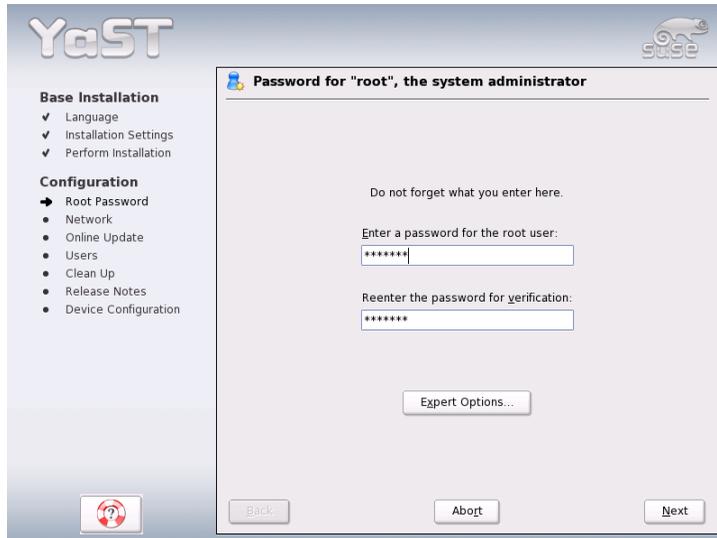


Figure 1.13: Setting the root Password

1.6.2 Network Configuration

You can now configure any network devices for a connection to the outside world, such as network cards, modems, and ISDN or DSL hardware. 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 include them in the installation.

To configure your network hardware at this stage, refer to *Network Integration* on page 422. Otherwise, select 'Skip Configuration' and click 'Next'. The network hardware can also be configured after the system installation has been completed.

1.6.3 Firewall Configuration

When you connect to a network, a firewall is started automatically on the configured interface. The configuration of the firewall is suitable for the respective interface. The firewall settings are also displayed in the network configuration dialog. The configuration proposal for the firewall is updated automatically every

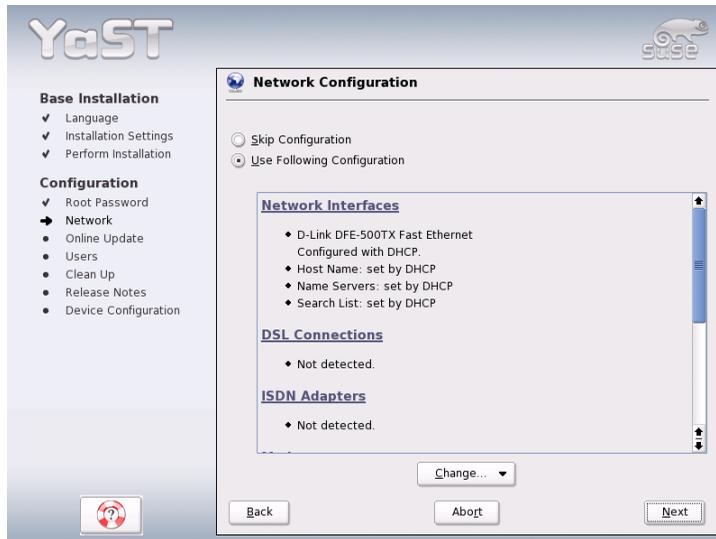


Figure 1.14: Configuring the Network Devices

time the configuration of the interfaces or services is modified. To adapt the automatic settings to your own preferences, click 'Change' → 'Firewall'. In the new dialog, determine whether the firewall should be started. If you do not want the firewall to be started, select the appropriate option and exit the dialog. To start and configure the firewall, click 'Next' for a series of dialogs similar to those described in *Configuration with YaST* on page 593.

1.6.4 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.

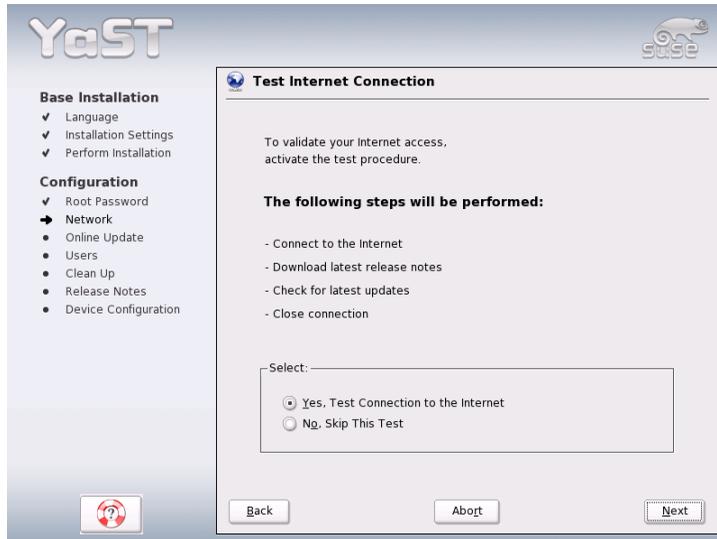


Figure 1.15: Testing the Internet Connection

1.6.5 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.

Note

Downloading Software Updates

The download of updates might take quite some time, depending on the bandwidth of the Internet connection and on the size of the update files.

Note

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 *YaST Online Update* on page 46. 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'.

1.6.6 User Authentication

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

Local User Administration Local user administration on the installed host. This is a suitable option for stand-alone workstations. The user data is managed by the local file `/etc/passwd`.

LDAP Central user administration on an LDAP server for all systems in the network.

NIS Central user administration on a NIS server for all systems in the network.

Samba SMB authentication in mixed Linux and Windows networks.

If all requirements are met, YaST opens a dialog in which to select the user administration method. It is shown in Figure 1.16 on the following page. If you do not have the necessary network connection, create local user accounts.

1.6.7 Configuring the Host as a NIS Client

To implement the user administration via NIS, configure a NIS client in the next step. This section only describes the configuration of the client side. Configuration of a NIS server with YaST is described in *NIS — Network Information Service* on page 459.

In the following dialog, shown in Figure 1.17 on page 37, first select whether the host has a static 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. Information about DHCP is available in *DHCP* on page 491. If a static IP address is used, specify the NIS domain and the NIS server manually.

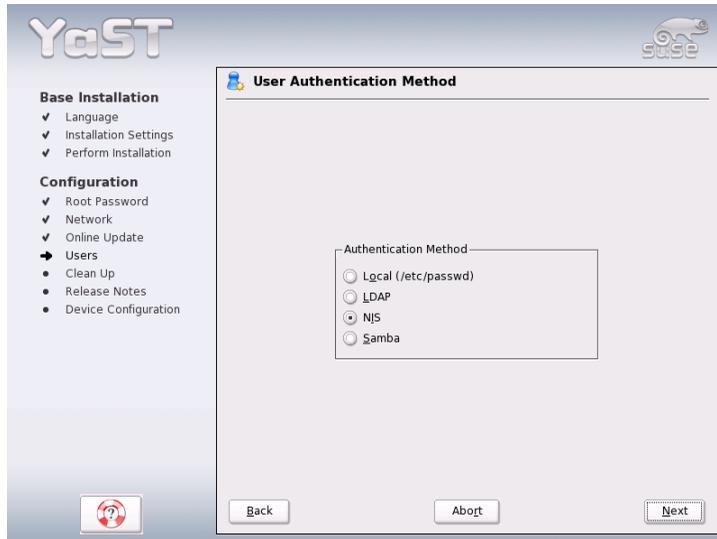


Figure 1.16: User Authentication

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 `yplibind`.

1.6.8 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.) is stored and managed on the installed system.

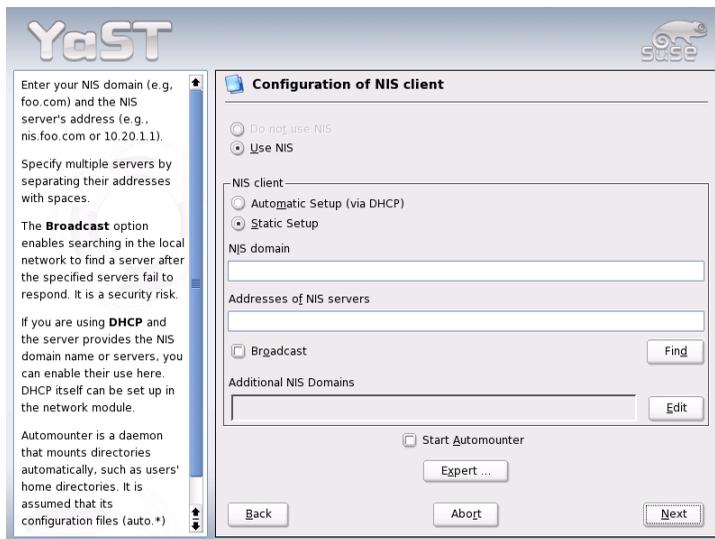


Figure 1.17: NIS Client Configuration

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.

A user account can be created using the dialog shown in Figure 1.18 on the following page. After entering the first name and last name, specify a 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). The user name tells the system who a user is and the password is used to verify this identity.

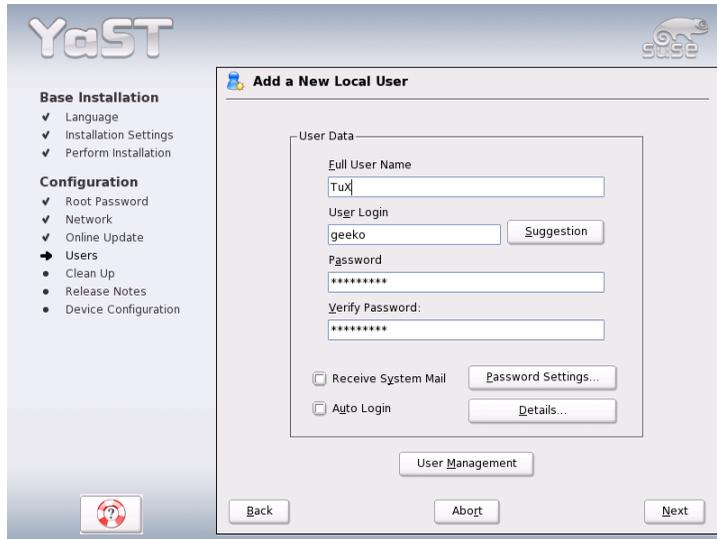


Figure 1.18: Entering the User Name and Password

Caution

User Name and Password

Remember both your user name and the password because they are needed each time you log in to the system.

Caution

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. This option is useful for the most frequently used account, because it is highly recommended to log in as `root` only in special cases.

‘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.

Caution

Automatic Login

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

Caution

1.6.9 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.

1.7 Hardware Configuration

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

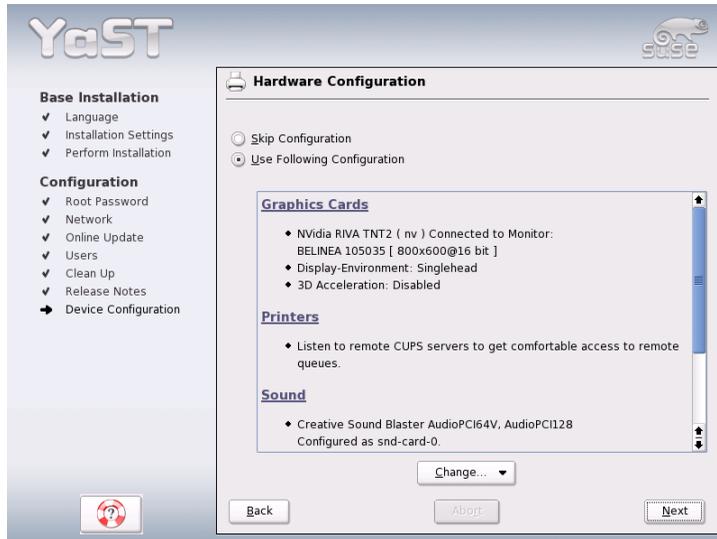


Figure 1.19: Configuring the System Components

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 *Graphics Card and Monitor (SaX2)* on page 64. After YaST has written the configuration data, finish the installation of SUSE LINUX with 'Finish' in the final dialog.

1.8 Graphical Login

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 1.20 on the facing page. Enter your login and password to log in to the system.

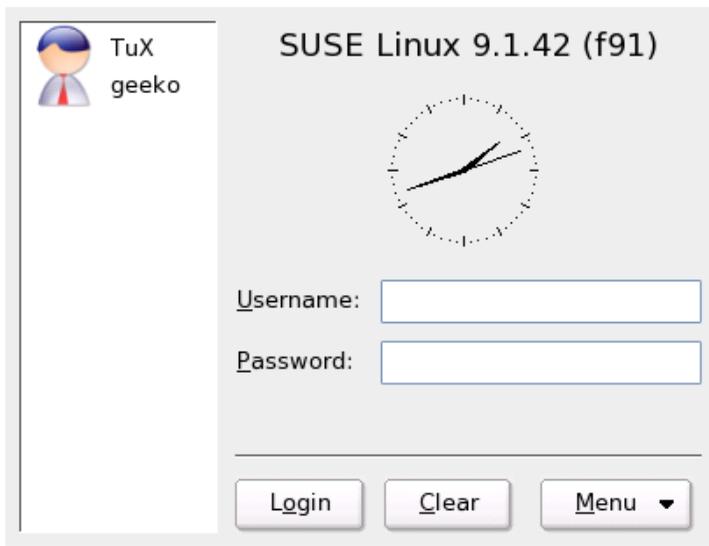


Figure 1.20: The Login Screen (KDE)

System Configuration with YaST

YaST, the setup tool used for the installation, is also the configuration tool for SUSE LINUX. This chapter covers the configuration of your system with YaST. This includes most of the hardware, the graphical user interface, Internet access, security settings, user administration, installation of software, system updates, and system information. This chapter also provides instructions for using YaST in text mode.

2.1	Starting YaST	44
2.2	The YaST Control Center	45
2.3	Software	46
2.4	Hardware	58
2.5	Network Devices	79
2.6	Network Services	79
2.7	Security and Users	83
2.8	System	88
2.9	Miscellaneous	93
2.10	YaST in Text Mode (ncurses)	95
2.11	Online Update from the Command Line	99

2.1 Starting YaST

The system configuration with YaST takes place by means of various YaST modules. Depending on the hardware platform and the installed software, there are different ways to access YaST in the installed system.

In KDE or GNOME, start the YaST Control Center from the SUSE menu ('System' → 'YaST'). Additionally, the individual YaST configuration modules are integrated in the KDE Control Center. Before YaST starts, you are prompted to enter the root password, because YaST needs system administrator permissions to change the system files.

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

Note

To change the language of YaST, select 'System' → 'Select Language' in the YaST Control Center. Choose a language, exit the YaST Control Center, log out from the system, and log in again. The next time you start YaST, the new language setting will be active.

Note

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

To start YaST in text mode from another system, use `ssh root@<system-to-configure>` to open the connection. Then start YaST with `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, opens. The left frame contains the categories 'Software', 'Hardware', 'Network Devices', 'Network Services', 'Security and Users', 'System', and 'Miscellaneous'. If you click the icons, the 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 consists of several steps. Press 'Next' to proceed to the following step.

The left frame displays a help text, 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.

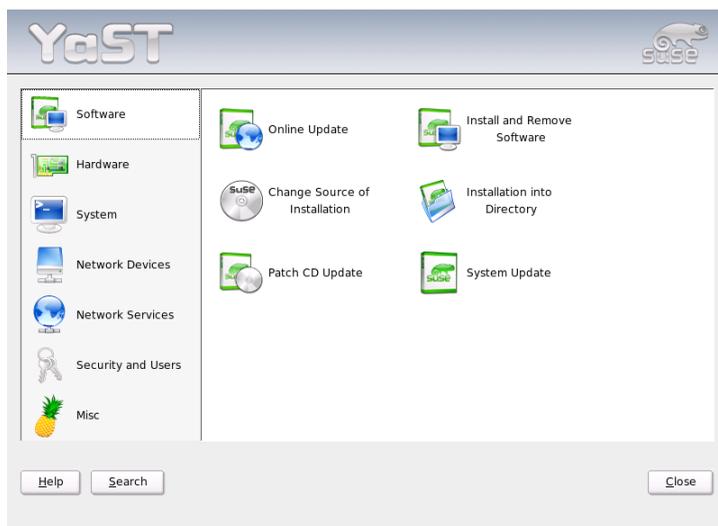


Figure 2.1: The YaST Control Center

2.3 Software

2.3.1 Change Installation Source

YaST can administer a number of different installation sources. It enables their selection for installation or update purposes.

When this module starts, it displays a list of all previously registered sources. Following a normal installation from CD, only the installation CD is listed. Click 'Add' to include additional sources in this list. As well as removable media like CDs and DVDs, you can add network sources, like NFS and FTP servers. Even directories on the local hard disk can be selected as the installation medium. See the detailed YaST help text.

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 a suitable entry from the list of activated installation sources. When you exit the module with 'Close', the current settings are saved and applied to the configuration modules 'Install and Remove Software' and 'System Update'.

2.3.2 YaST Online Update

The YaST Online Update (YOU) enables the installation of important updates and improvements. These patches are available for download on the SUSE FTP server and various mirror servers.

Under '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. You can also specify local URLs in the format `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.

When the module starts, 'Manual Selection of Patches' is active, enabling selection of the patches to fetch. 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.

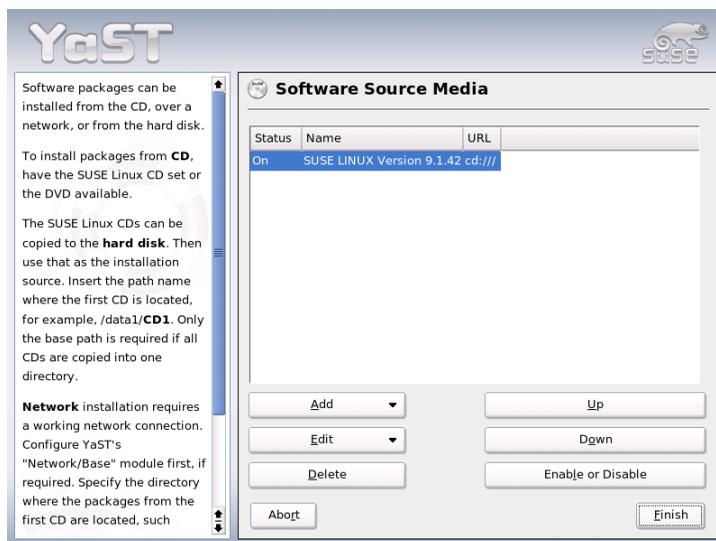


Figure 2.2: Changing the Installation Source

Additionally, the system can be updated automatically. Click 'Configure Fully Automatic Update' to configure a process that automatically looks for updates and applies them on a regular basis. This procedure is fully automated. Of course, the system must be able to connect to the update server at the scheduled time.

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 *Installing and Removing Software* on the following page. In the package manager, the filter for YOU patches is activated, enabling selection of updates to install. At start-up, the available security patches and recommended patches are preselected, provided the respective packages are installed on the system. This proposal should be accepted.

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 a problematic package. Prior to the installation, some patches open a window displaying details.

While the updates are downloaded and installed, you can track all 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, select 'Remove Source Packages after Update' for them to be deleted after the update. Finally, SuSEconfig is executed to adjust the system configuration as needed.

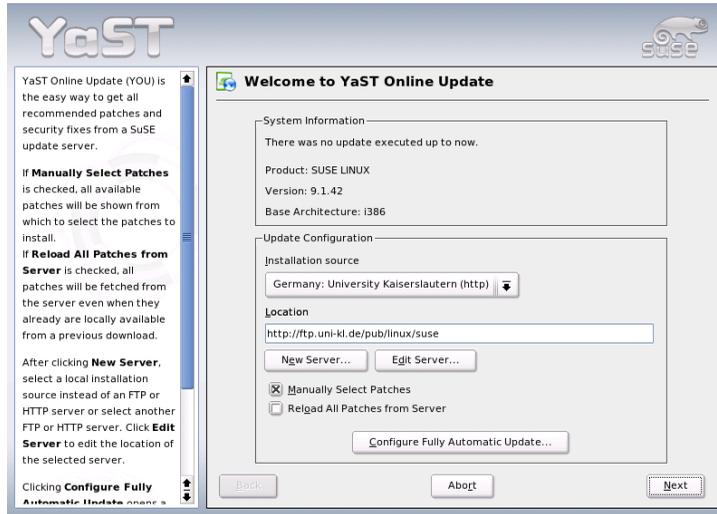


Figure 2.3: YaST Online Update

2.3.3 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: the program itself, the configuration files, and documentation. A package containing the source files for the program is normally available as well. The sources are not needed for running the program, but you may want to install the sources to compile a custom version of the program.

Some packages depend on other packages. This means that the software of the package only works properly if another package is also installed. Furthermore, the installation 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 provides an extremely efficient tool for this purpose: the software installation module, usually referred to as the package manager. When the package manager starts, 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 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 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 next 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.

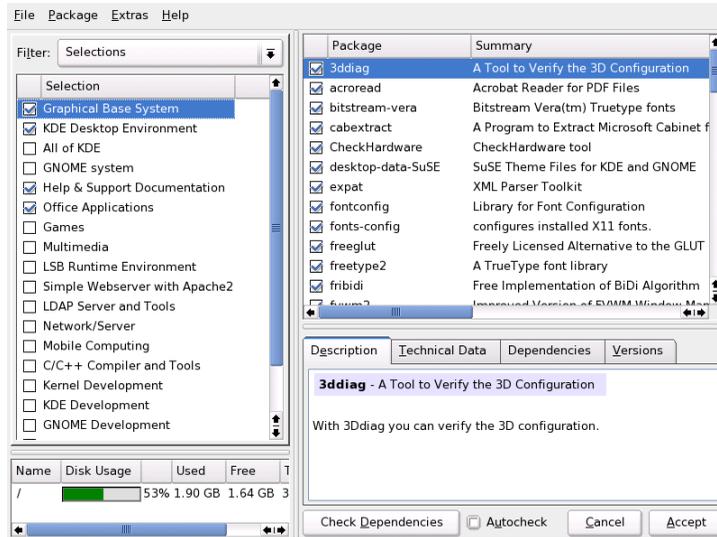


Figure 2.4: YaST Package Manager

The Filter Window

The package manager offers various filter methods for arranging the packages in categories and limiting the number of packages displayed. The filter window is located to the left under the menu bar. It controls and displays various filter methods. The filter selection box at the top determines what is 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 who are 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.

Additionally, this filter provides the possibility to display all packages in alphabetic order without any categorization. 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, you can 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 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.

Note

Quick Search

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

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 selecting the desired status 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 because it is required by another package (resolution of package dependencies). To deselect such a package, you may need to use the status "Taboo".

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 that this package be deleted. For example, this may be the case if the current package has been replaced by a different package.

Automatic Installation (after selection)

This package 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, you can decide whether 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 version 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 other packages on which it depends, or the version information.

The Resource Window

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

The Menu Bar

The menu bar at the top left of the window provides access to most of the functions described above and 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 selected 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 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 need 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 under the information window. If you click ‘Check Dependencies’, the package manager checks if the current package selection results in any unresolved package dependencies or conflicts. In the event of unresolved dependencies, the required additional packages are selected automatically. For package conflicts, the package manager opens a dialog that shows the conflict and offers various options for solving the problem. If you activate ‘Autocheck’, any change of a package status triggers an automatic check. This is a useful feature, because the consistency of the package selection is monitored permanently. However, this process consumes resources and can slow down the package manager. For this reason, the 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 following 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

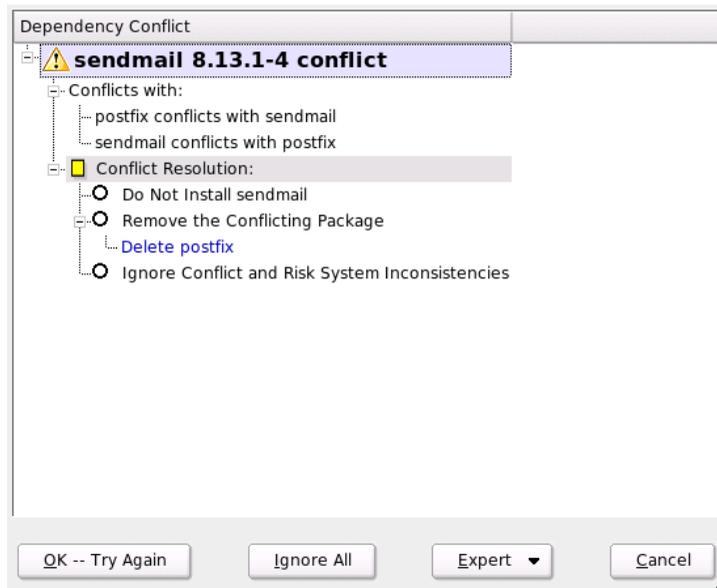


Figure 2.5: Conflict Management of the Package Manager

2.3.4 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 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 a new installation. Initially, YaST examines the system, determines a suitable update strategy, and presents the results in a suggestion dialog. 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 (see *Language Selection* on page 12). This section only covers update-specific settings.

Selected for Update

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

Update Options

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

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 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.

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 *Installing and Removing Software* on page 48.

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 first check which version is installed on the computer then determine what needs to be done to replace the old version with the new version correctly. YaST also tries to adopt any personal settings of the installed packages. Some configurations may cause problems because the old configuration is unable to handle the new program version as expected or because unexpected inconsistencies arise between various configurations.

The older the existing version is and the more the configuration of the packages to update diverges from the standard, the more problematic the update will be. Sometimes, the old configuration cannot be adopted correctly. In this case, an entirely new configuration must be made. Before starting the update, the existing configuration should be saved.

2.4 Hardware

New hardware must first be installed or connected as specified by the vendor. Turn on external devices, such as the printer or the modem, 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 and DVD 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.

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.

Usually, this data 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 to a printer language is achieved with a print filter.

Some Standard Printer Languages

Standard printer languages can be categorized in the following three groups:

ASCII Text Every normal printer can print ASCII text directly. However, there are printers that cannot print ASCII text directly, but can be addressed by means of one of the following standard printer languages.

PostScript PostScript is the established printer language on Unix and Linux systems. However, PostScript data can be printed directly only by PostScript printers.

PCL3, PCL4, PCL5e, PCL6, ESC/P, ESC/P2, and ESC/P Raster

If a PostScript printer is not available, the print filter can use Ghostscript to convert PostScript data into one of these other standard printer languages. Ghostscript relies on different drivers for different printers to make the best possible use of specific features offered by the various models (such as color settings).

How a Print Job is Processed

1. The user or an application generates a new print job.
2. The print data is temporarily stored in the print queue. The print spooler sends it from there to the printer filter.
3. Now the printer filter performs the following steps:
 - (a) It determines the type of print data.
 - (b) The print data is converted into PostScript, if not in PostScript already.
 - (c) If necessary, the PostScript data is converted to another printer language.
 - If the printer is a PostScript model, the PostScript data is sent directly to the printer.
 - If the printer is not a PostScript model, Ghostscript uses a driver suitable for the respective printer to generate the printer-specific data that is subsequently sent to the printer.
4. After the entire job has been sent to the printer, the print spooler removes the job from the queue.

Supported Printers

Because most Linux printer drivers are not written by the manufacturer of the hardware, it is crucial that the printer can be driven through one of the generally known languages. Normal printers understand at least one of these common languages. In the case of a GDI printer, however, the manufacturer has built a device that relies on its own special control sequences. Many inexpensive ink jet models belong to this group. Such a printer only runs out of the box under the versions of the operating systems for which the manufacturer has included a driver. Because the device cannot be operated through one of the standard languages, it cannot be used with Linux or can only be used with difficulties.

Nevertheless, some of these printers are supported by SUSE LINUX. However, their use is often rather problematic and some features might not be available at all. For example, the printer could be limited to low resolution monochrome printing. Refer to *Proprietary Printers (usually GDI printers)* on page 265 and *Printers without Standard Printer Language Support* on page 276 for more information about the use of these devices.

Configuration with YaST

To configure the printer, select 'Hardware' → 'Printer' in the YaST control center. This opens the main printer configuration window, where the detected devices are listed in the upper part. The lower part lists any queues configured so far. If your printer was not detected, configure it manually.

Automatic Configuration

YaST is able to configure the printer automatically if the parallel or USB port can be set up automatically and the connected printer can be detected. The printer database must also contain the ID string of the printer that YaST retrieves during the automatic hardware detection. If the hardware ID differs from the model designation, 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 that is being tested.

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 with the connection on the hardware level. In this case, 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.

Printer Model and PPD File All printer-specific parameters, such as the Ghostscript driver to use and the printer filter parameters for the driver, are stored in a PPD (PostScript Printer Description) file. See *Installing the Software* on page 266 for more information about PPD files.

For many printer models, several PPD files are available, for example, if several Ghostscript drivers work with the given model. When you select a manufacturer and a model, YaST selects the PPD file that corresponds to the printer. If several PPD files are available for the model, YaST defaults to one of them (normally the one marked recommended). You can change the default PPD file after selecting 'Edit'.

For non-PostScript models, all printer-specific data is produced by the Ghostscript driver. For this reason, the driver configuration is the single most important factor determining the output quality. The printout is affected both by the kind of Ghostscript driver (PPD file) selected and the options specified for it. If necessary, change additional options (as made available by the PPD file) after selecting 'Edit'.

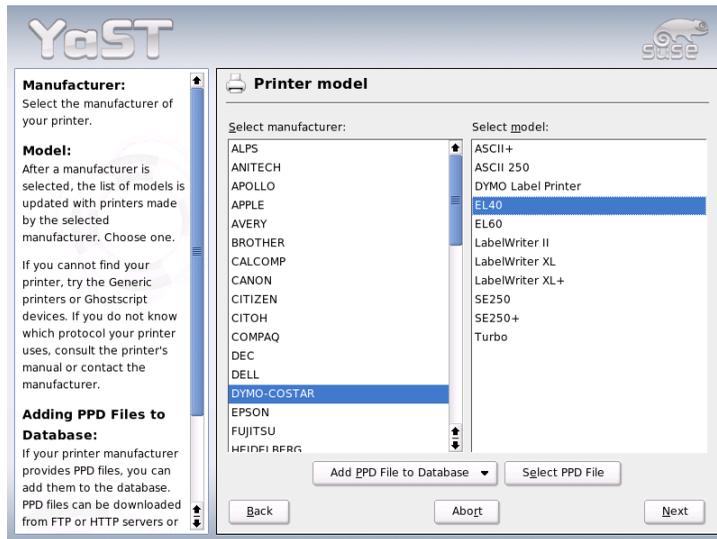


Figure 2.6: Selecting the Printer Model

Always check whether your settings work as expected by printing the test page. If the output is garbled, for example, with several pages almost empty, you should be able to stop the printer by first removing all paper then stopping the test from YaST.

If the printer database does not include an entry for your model, you can use a collection of generic PPD files to make the printer work with one of the standard printer languages. To do so, select 'UNKNOWN MANUFACTURER' as your printer manufacturer.

Advanced Settings Normally, you do not need to change any of these settings.

Configuration for Applications

Applications rely on the existing printer queues in the same way as any command-line tools do. There is usually no need to reconfigure the printer for a particular application, because you should be able to print from applications using the available queues.

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

Printing from Applications Using the Command-Line Tool

Some applications rely on the above-mentioned `lp` command for printing. In this case, enter the correct command in the application's print dialog (but usually without specifying *<filename>*), for example, `lp -d <queuename>`. To make this work with KDE programs, enable 'Print through an external program'. Otherwise you cannot enter the print command.

Using the CUPS Printing System Tools such as `xpp` and the KDE program `kprinter` provide a graphical interface to choose among queues and to set both CUPS standard options and printer-specific options as made available through the PPD file. You can use `kprinter` as the standard printing interface of other (non-KDE) applications by specifying `kprinter` or `kprinter --stdin` as the print command in the print dialogs of these applications. The behavior of the application itself determines which of these two commands to choose. If set up correctly, the application should call the `kprinter` dialog whenever a print job is issued from it, so you can use the dialog to select a queue and to set other printing options. This requires that the application's own print setup does not conflict with that of `kprinter` and that printing options are only changed through `kprinter` after it has been enabled.

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 in the correct way. This could cause the output to be garbled and use up large amounts of paper. To correct this, follow the instructions in *Defective Print Jobs and Data Transfer Errors* on page 282.

For More Information

Detailed information about printing in Linux is available in *Printer Operation* on page 263, which covers general printing issues. Solutions to many specific problems are presented in the Support Database. If you experience problems with printers, refer to the Support Database articles *Installing a Printer* and *Printer Configuration from SUSE LINUX 9.2*, which you can find by searching for the keyword “printer”.

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

This is an expert tool. Your system may no longer boot if you make incorrect settings. If you make changes, use the test option.

Caution

2.4.4 Graphics Card and Monitor (SaX2)

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 2.7.

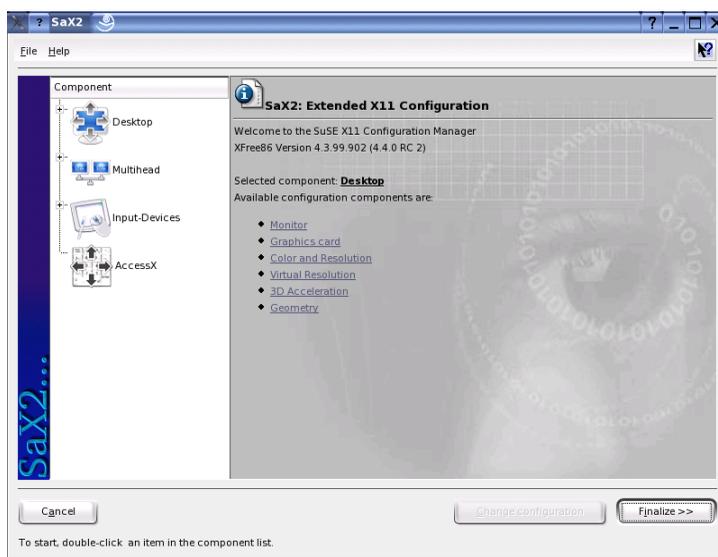


Figure 2.7: The Main Window of SaX2

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 be configured under ‘Input devices’. Use ‘Multihead’ to configure multiple screens (see *Multihead* on page 70). ‘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. 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 'Manufacturer Disk'.

'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. Additionally, 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. Incorrect values may destroy your monitor. If in doubt, refer to the manual of the monitor.

Caution

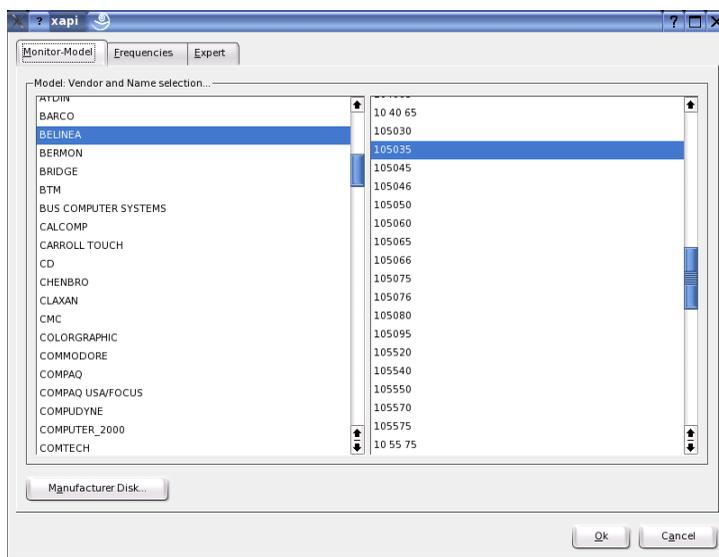


Figure 2.8: Monitor Selection

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.

‘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.

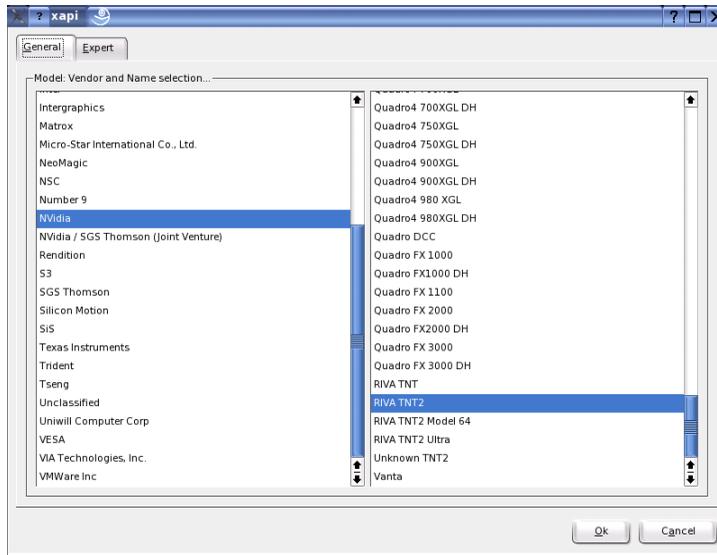


Figure 2.9: Selecting the Graphics Card

‘Resolution’ The module offers all 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.

‘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.

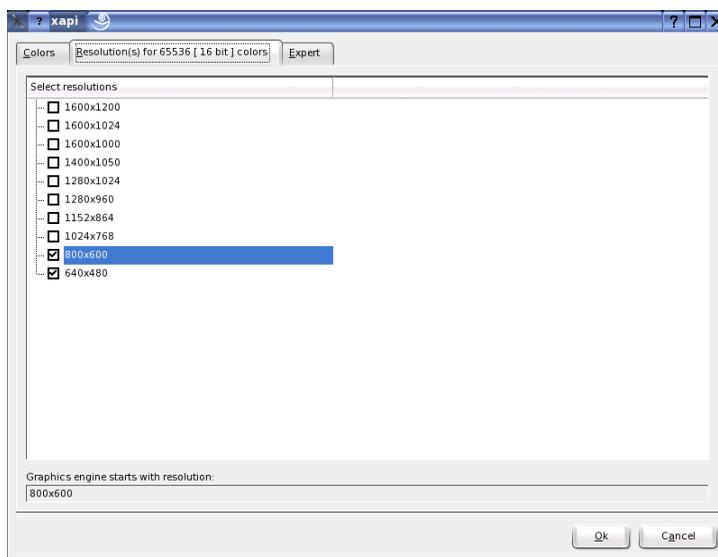


Figure 2.10: Configuring the Resolution

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.

3D Acceleration

If you did not activate 3D acceleration during the initial installation or when you installed a new graphics card, you can do this here.

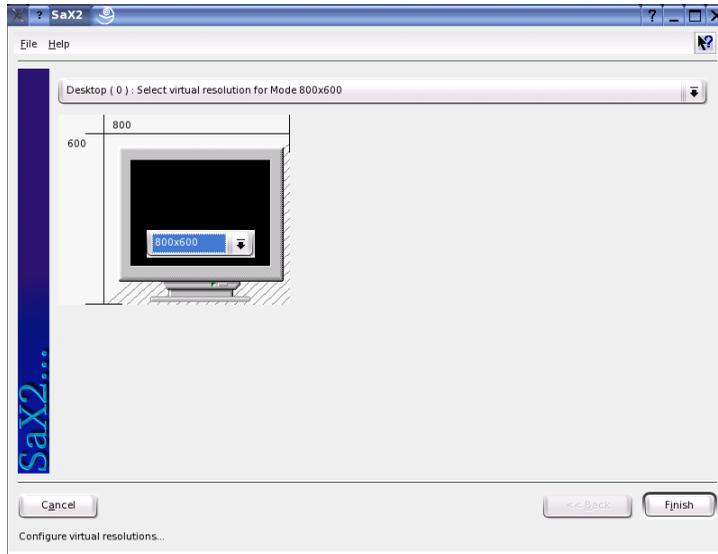


Figure 2.11: Configuring the Virtual Resolution

Image Position and Size

Under these two tabs, precisely adjust the size and the position of the image with the arrows. See Figure 2.12 on the facing page. If you have a multihead environment (more than one screen), use 'Next screen' to switch to the other monitors to adjust their sizes and positions. Press 'Save' to save your settings.

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.

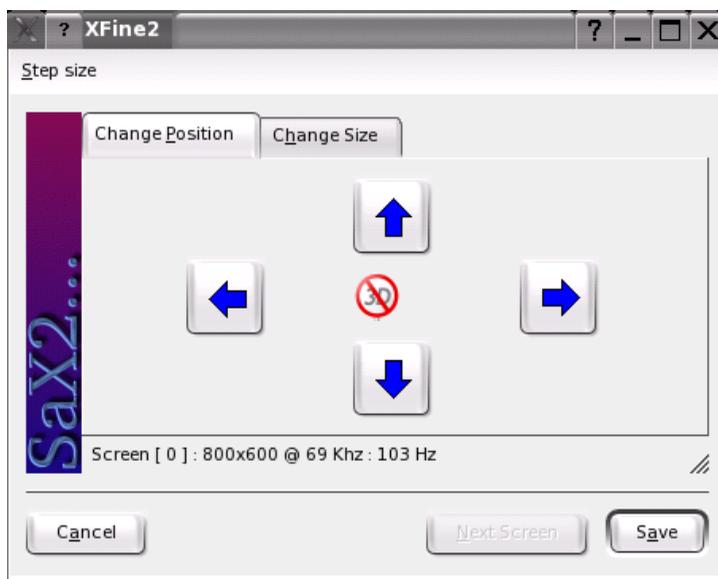


Figure 2.12: Adjusting the Image Geometry

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 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, X.Org 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, X.Org 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 for a description of the functions of the different keys. Use the slider to set the speed of the mouse pointer movement when a key is pressed.

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
⓪	releases the click-and-hold event previously invoked with ⑪
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

For More Information

For more information about the X Window System and its properties, refer to *The X Window System* on page 247.

2.4.5 Hardware Information

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

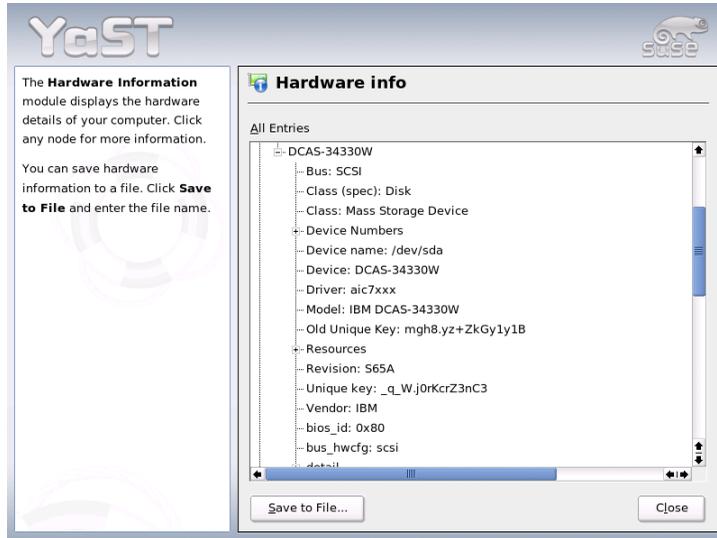


Figure 2.13: Displaying Hardware Information

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, because 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 Joystick

Use this module to configure your joystick by selecting the manufacturer and the model from the displayed list. With 'Test', check if your joystick responds correctly. The test dialog shows three charts for the analog axes of the joystick and marks for the four standard buttons. When you move the joystick or press the buttons, you should be able to see a reaction in the test dialog. Because joysticks are usually connected to the sound card, you can also access this module from the sound card configuration.

2.4.8 Keyboard Layout Selection

The desired keyboard layout usually matches the selected language, but can also be selected regardless of the language. Use the test field to see if special characters such as the pipe symbol | are displayed correctly.

2.4.9 Mouse

Configure your mouse with this YaST module. The procedure for the selection of the mouse was already explained for installation, so refer to *Mouse* on page 14.

2.4.10 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/sdg0`. 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> and <http://www.mostang.com/sane>.

Caution

Assigning a Scanner Manually

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://cdb.suse.de/index.php?LANG=en> for a list of Linux-compatible devices.
- The SCSI controller was not installed correctly.

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

Caution

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

Caution

For more information about scanning, refer to the chapter about kooka in the *User Guide*.

2.4.11 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. 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'.

Setup

Choose the configuration level in the first setup screen. 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. 'Advanced Setup' allows you to customize the sound card options manually.

In this dialog, also find a shortcut to the joystick configuration. Click the respective check box. Select the joystick type in the following dialog and click 'Next'.

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 is then 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. 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, 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/`.

For playback of MIDI files, activate 'Start Sequencer'. This way, the modules for sequencer support are loaded along with the sound 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/modprobe.conf`.

2.4.12 TV and Radio Cards

After starting and initializing this YaST module, the 'TV and Radio Cards' dialog appears. If your card was automatically detected, it is displayed at the top of the list. In this case, highlight the line with the mouse and select 'Configure'. If your card was not detected, select 'Other (not recognized)'. Press 'Configure' to proceed with the manual selection in which to select your card from the list of vendors and models.

If you have already configured TV or radio cards, modify existing configurations with ‘Change’. In this case, a dialog presents a list of all configured cards. Select a card and start the manual configuration with ‘Edit’.

During the automatic hardware detection, YaST attempts to assign the correct tuner to your card. If you are not sure, simply keep the setting ‘Default (recognized)’ and check whether it works. If you are not able to set all channels, this might be due to a failure of the automatic detection of the tuner type. In this case, click ‘Select Tuner’ and highlight the correct tuner type in the list.

If you are familiar with the technical details, you can use the expert dialog to specify settings for a TV or radio card. Select a kernel module and its parameters in this dialog. Also check all parameters of your TV card driver. To do this, select the respective parameters and enter the new value in the parameter line. Confirm the new values with ‘Apply’ or restore the default values with ‘Reset’.

The dialog ‘TV and Radio Cards, Audio’ enables you to connect your TV or radio card with the installed sound card. You must use a cable to connect the output of the TV or radio card with the external audio input of the sound card. This only works if the sound card is already configured and the external input is active. If you have not yet configured your sound card, select ‘Configure Sound Card’ to go to the respective dialog, described in *Sound* on page 77.

If your TV or radio card has speaker jacks, you can also connect the speakers directly without configuring the sound card. There are also TV cards without any sound function, which do not require an audio configuration, such as those for CCD cameras.

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 *Network Integration* on page 422. The configuration of network devices for wireless communication is described in *Wireless Communication* on page 337.

2.6 Network Services

This group mostly contains tools for the name resolution, user authentication, and file and print services in large networks.

2.6.1 Administration from a Remote Host

To maintain your system over a VNC connection from a remote host, permit the establishment of connections with this YaST module.

2.6.2 DHCP Server

YaST can set up a custom DHCP server in only a few steps. *DHCP* on page 491 provides basic knowledge about the subject as well as a step-by-step description of the configuration process in YaST.

2.6.3 Host Name and DNS

Use this module to configure the host name and DNS, if these settings were not already been made while configuring the network devices. Also use it to change the host name and domain name. If the provider has been configured correctly for DSL, modem, or ISDN access, the list of name servers contains the entries that were extracted automatically from the provider data. If you are located in a local network, you might receive your host name via DHCP, in which case you should not modify the name.

2.6.4 DNS Server

The configuration of a DNS server that is responsible for the name resolution is recommended for larger networks. Configuration with YaST is described in *Configuration with YaST* on page 450. *DNS — Domain Name System* on page 438 provides background information about DNS.

2.6.5 HTTP Server

To run your own web server, configure Apache with YaST. More information is available in *The Apache Web Server* on page 507.

2.6.6 LDAP Client

LDAP can be used instead of NIS for the user authentication in the network. Background information for LDAP and a detailed description of the client configuration with YaST are available in *LDAP — A Directory Service* on page 464.

2.6.7 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.

Alternatively, 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.

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 AMoViS 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.

2.6.8 NFS Client and NFS Server

NFS enables you to run a file server that all members of your network can access. This file server can be used to make certain applications, files, and storage space available to users. In the ‘NFS Server’ module, you can configure your host as an NFS server and determine the directories to export for general use by the network users. All users with the appropriate permissions can mount these directories in their own file trees. A description of the YaST module and background information about NFS are provided in *NFS — Shared File Systems* on page 486.

2.6.9 NIS Client and NIS Server

If you run more than one system, local user administration (using the files `/etc/passwd` and `/etc/shadow`) is impractical and requires a lot of maintenance. In this case, the user data should be administered on a central server and distributed to the clients from there. Apart from NIS is a possible solution, as are LDAP and Samba. Detailed information about NIS and the configuration with YaST is available in *NIS — Network Information Service* on page 459.

2.6.10 NTP Client

NTP (network time protocol) is a protocol for synchronizing hardware clocks over a network. Background information about NTP and a description of the configuration with YaST is available in *Time Synchronization with xntp* on page 500.

2.6.11 Network Services (inetd)

Use this tool to determine the network services (such as `finger`, `talk`, and `ftp`) to start when SUSE LINUX boots. These services enable external hosts to connect to your computer. Various parameters can be configured for every service. By default, the master service that manages the individual services (`inetd` or `xinetd`) is not started.

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

Caution

Configuring Network Services (inetd)

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

Caution

2.6.12 Routing

You only need this tool if you are located in a local network or connected to the Internet with a network card, as is the case with DSL. For DSL, the gateway data is only needed for configuring the network cards. However, the entries are merely dummies without any function. This value is only important if you are located in a local network and use a special host as gateway. More information on routing is available in *Routing in SUSE LINUX* on page 434.

2.6.13 Configuring Samba Servers and Clients

In a heterogeneous network consisting of Linux and Windows hosts, Samba controls the communication between the two worlds. Information about Samba and the configuration of clients and servers is provided in *Samba* on page 551.

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 (for example, `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 networks.

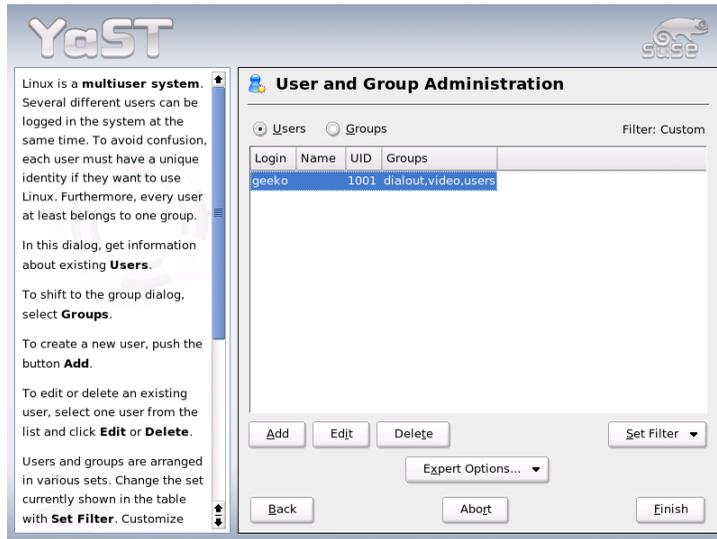


Figure 2.14: User Administration

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 selected line is 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.

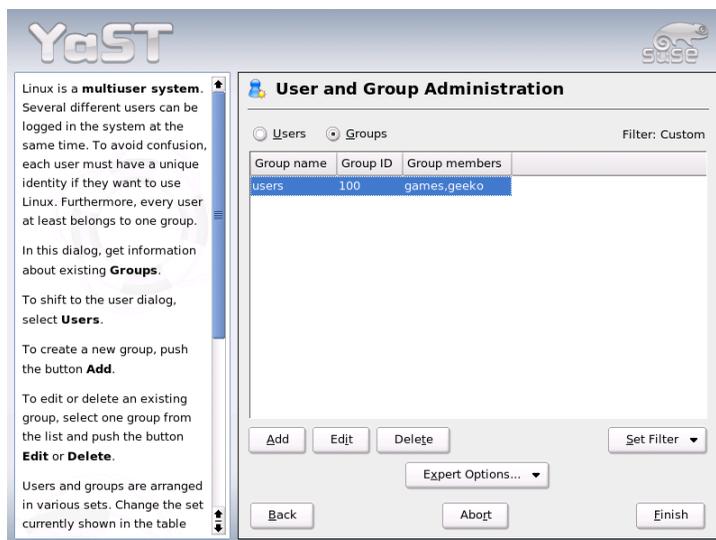


Figure 2.15: Group Administration

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 stand-alone 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 desired action. 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. Because 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. 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.

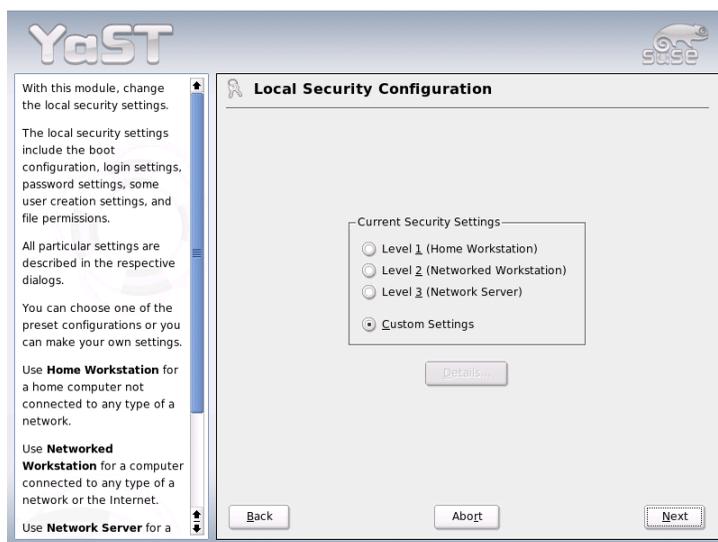


Figure 2.16: Security Settings

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 *Masquerading and Firewalls* on page 588.

Note**Automatic Activation of the Firewall**

YaST automatically starts a firewall with suitable settings on every configured network interface. You only need to start this module if you want to reconfigure the firewall with custom settings or deactivate it.

Note

2.8 System

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 include the entire system. It 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 2.17 on the facing 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 is a dialogs for uninstalling packages that were added since the last backup and one for reinstalling 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**

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

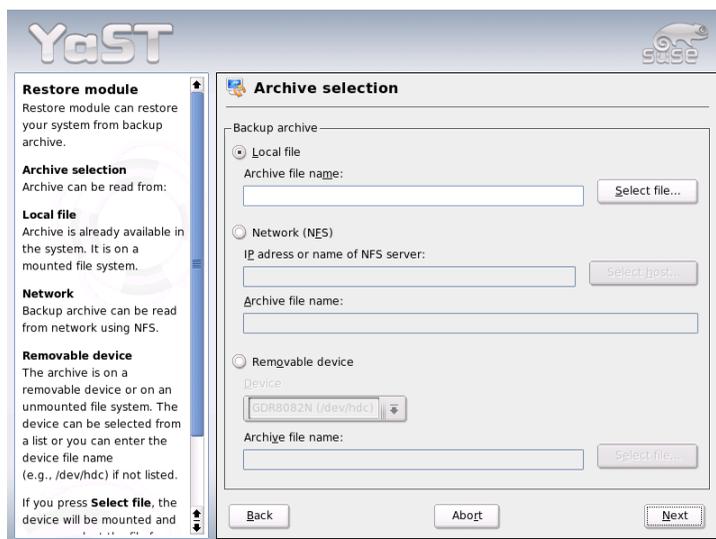
Caution

Figure 2.17: 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 (for example, to access a RAID system).

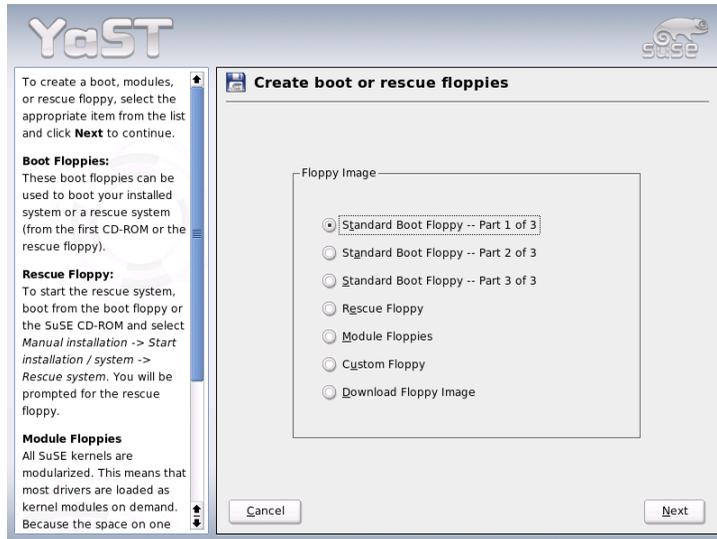


Figure 2.18: Creating a Boot, Rescue, or Module Disk

‘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 drivers 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’, a dialog opens 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 Because the standard kernel only supports normal IDE drives, you 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.

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 required module disk and the modules contained 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 LVM

The logical volume manager (LVM) is a tool for custom partitioning of hard disks with logical drives. More information about LVM is available in *LVM Configuration* on page 126.

2.8.5 Partitioning

Although it is possible to modify the partitions in the installed system, this should be handled by experts. Otherwise the risk of losing data is very high. If you decide to use this tool, refer to the description in *Partitioning* on page 15 (the partitioning tool during the installation is the same as in the installed system).

2.8.6 Profile Manager (SCPM)

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, because it enables the use of various hardware components or test configurations. For more information about SCPM basics and handling, refer to *System Configuration Profile Management* on page 305.

2.8.7 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 you 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`.

For more information about the runlevels in SUSE LINUX and a description of the YaST runlevel editor, refer to *The SUSE LINUX Boot Concept* on page 229.

2.8.8 Sysconfig Editor

The directory `/etc/sysconfig` contains the files with the most important settings for SUSE LINUX. 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, because the files are automatically adapted when a package is installed or a service is configured. More information about `/etc/sysconfig` and the YaST sysconfig editor is available in *The SUSE LINUX Boot Concept* on page 229.

2.8.9 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.

2.8.10 Language Selection

Here, select the language for your Linux system. The language selected in YaST applies to the entire system, including YaST and the desktop environment.

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 <http://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. See Figure 2.19 on the following page. Also read the YaST help text, which explains how best to describe the problem so the support team can help you.

Note

If you need advanced support (such as for special problems), refer to <http://support.novell.com/linux/> for details.

Note

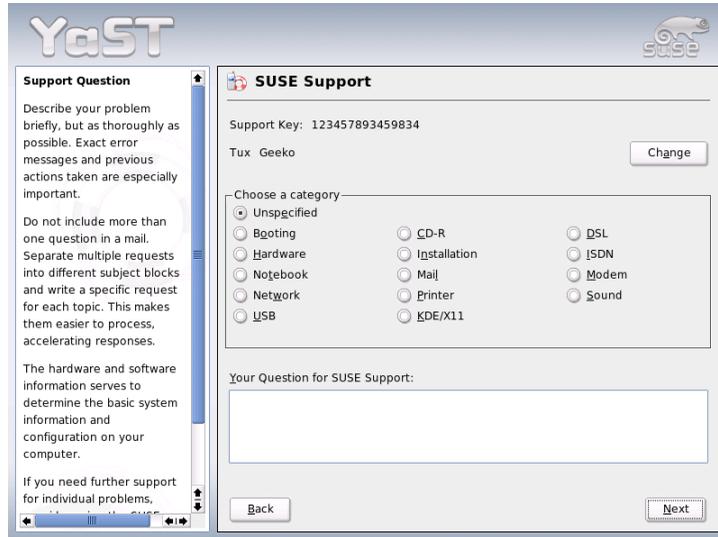


Figure 2.19: Submitting a Support Request

2.9.2 Boot Log

The boot log `/var/log/boot.msg` contains the screen messages displayed when the computer starts. 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/messages`. Kernel messages are recorded here, sorted according to date and time.

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)

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

When YaST is started in text mode, the YaST Control Center appears first. See Figure 2.20 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 \downarrow and \uparrow to change the category. To start a module from the selected category, press \rightarrow . The module selection now appears with a thick border. Use \downarrow and \uparrow 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 $\text{Alt} + \text{yellow_letter}$ 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 .

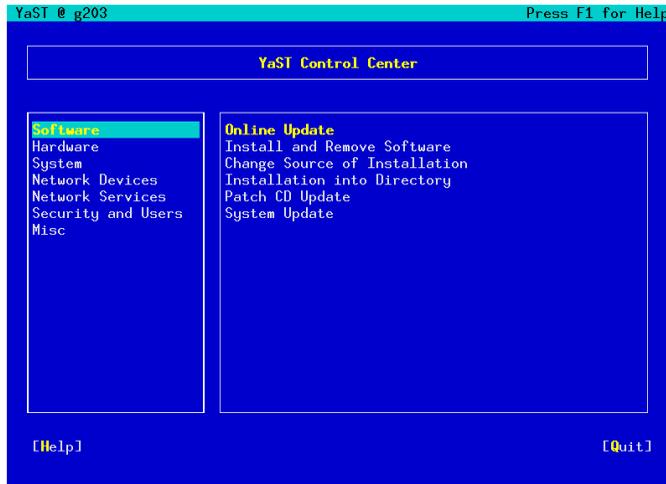


Figure 2.20: Main Window of YaST in Text Mode

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 *Restriction of Key Combinations* on page 98 for information about possible exceptions.

Navigation among Buttons and Selection Lists

Use **(Tab)** and **(Alt)-(Tab)** or **(Shift)-(Tab)** to navigate among the buttons and the frames containing selection lists.

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

Buttons, Radio Buttons, and Check Boxes

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

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

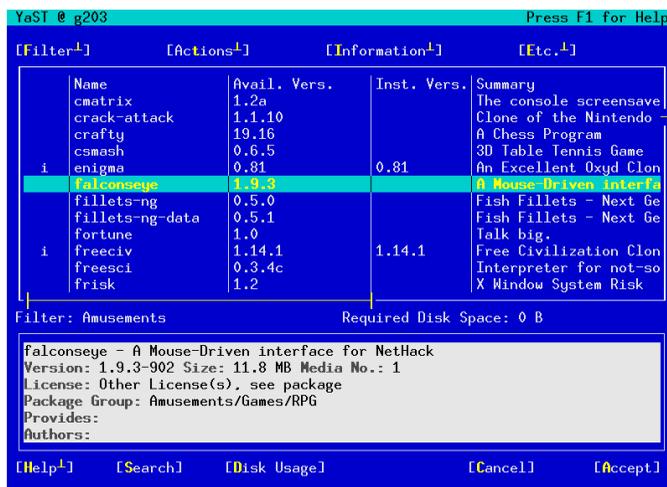


Figure 2.21: The Software Installation Module

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

2.10.3 Starting the Individual Modules

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

2.10.4 The YOU Module

The YaST Online Update (YOU) module can be started from the command line as root like any other YaST module:

```
yast online_update .url <url>
```

`yast online_update` starts the respective module. The option `url` can be used to specify the server (local or on the Internet) from which YOU should get all information and patches. If you do not specify a server when starting the module, select the server or the directory in the YaST dialog. Configure cron jobs for automating the update with 'Configure Fully Automatic Update'.

2.11 Online Update from the Command Line

Using the command-line tool `online_update`, the system can be updated automatically (e.g., by means of scripts). For instance, you may want your system to search a specific server for updates and download the patches and patch information at a specified time in regular intervals. However, you may not want the patches to be installed automatically. Instead, you may want to review the patches and select the patches for installation at a later time.

To use the tool, first configure a cron job that executes the following command:

```
online_update -u <URL> -g <type_specification>
```

`-u` introduces the base URL of the directory tree from which the patches should be downloaded. The supported protocols are `http`, `ftp`, `smb`, `nfs`, `cd`, `dvd`, and `dir`. `-g` downloads the patches to a local directory without installing them. Optionally, filter the patches by specifying the type: `security`, `recommended`, or `optional`. If no filter is specified, `online_update` downloads all new `security` and `recommended` patches.

The downloaded packages can be installed immediately without reviewing the individual patches. `online_update` saves the patches in the directory `/var/lib/YaST2/you/mnt`. To install the patches, execute the following command:

```
online_update -u /var/lib/YaST2/you/mnt/ -i
```

The parameter `-u` specifies the (local) URL of the patches to install. `-i` starts the installation procedure.

To review the downloaded patches prior to the installation, start the YOU dialog:

```
yast online_update .url /var/lib/YaST2/you/mnt/
```

YOU starts and uses the local directory containing the downloaded patches instead of a remote directory on the Internet. Select the patches to install in the same way as packages for installation in the package manager.

The behavior of the YaST Online Update can be controlled with command-line parameters. The syntax is `online_update [command-line parameter]`. The possible parameters and their functions are listed below.

- u **URL** Base URL of the directory tree from which the patches should be downloaded.
- g Only download patches. Do not install.
- i Install downloaded patches. Do not download.
- k Check if new patches are available.
- c Show the current configuration. Do not perform any action.
- p **product** Product for which patches should be downloaded.
- v **version** Product version for which patches should be downloaded.
- a **architecture** Base architecture of the product for which patches should be downloaded.
- d Dry run. Download patches and simulate installation (system remains unchanged; for test purposes only).
- n No signature check of the downloaded files.
- s Display a list of available patches.
- V Verbose mode.
- D Debug mode for experts and for troubleshooting.

For more information about `online_update`, enter `online_update -h`.

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 adaptations. The following sections cover various installation procedures and the use of diverse installation sources, including CD-ROM and NFS. This chapter also features information about resolving problems encountered during the installation and a detailed section about partitioning.

3.1	linuxrc	102
3.2	Installation with VNC	111
3.3	Text-Based Installation with YaST	112
3.4	Starting SUSE LINUX	114
3.5	Special Installation Procedures	116
3.6	Tips and Tricks	116
3.7	ATAPI CD-ROM Hangs while Reading	120
3.8	Permanent Device Names for SCSI Devices	121
3.9	Partitioning for Experts	122
3.10	LVM Configuration	126
3.11	Soft RAID	135

3.1 linuxrc

Every machine has special routines that are executed on start-up to initialize the hardware to enable booting. During the actual boot process, these routines, referred to as the BIOS, load an image that is executed by the machine. This image can be a boot manager, but it is possible to load the kernel directly. During the installation of SUSE LINUX, a boot image containing a kernel and a program called `linuxrc` is loaded.

`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. In SUSE LINUX, `linuxrc` starts after the system analysis by YaST. The automatic hardware detection performed by YaST is usually quite reliable. However, if you want to load kernel modules manually or enter special parameters, you can use `linuxrc` interactively. In this case, start a manual installation.

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 *The SUSE Rescue System* on page 172 for 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 \uparrow and \downarrow to select a menu item and \leftarrow and \rightarrow to select an action, such as 'OK' or 'Cancel'. Confirm the selection by pressing `Enter`.

After starting, `linuxrc` automatically prompts you to select your language and keyboard layout. 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`, shown in Figure 3.2 on page 104. Normally, `linuxrc` is used to start Linux, in which case you should select 'Start Installation or System'. Whether you can directly proceed with this option depends on the kind of hardware you use and on the scope of the installation project. Refer to *Text-Based Installation with YaST* on page 112 for more information.



Figure 3.1: Selecting the Language

3.1.2 System Information

With the ‘System Information’ menu, shown in Figure 3.3 on page 105, 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.

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: IC35L060AVER07-0, ATA DISK drive
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
hdc: DV-516E, ATAPI CD/DVD-ROM drive
ide1 at 0x170-0x177,0x376 on irq 15
hda: max request size: 128KiB
hda: 120103200 sectors (61492 MB) w/1916KiB Cache, CHS=65535/16/63, UDMA(100)
hda: hda1 hda2 hda3
```

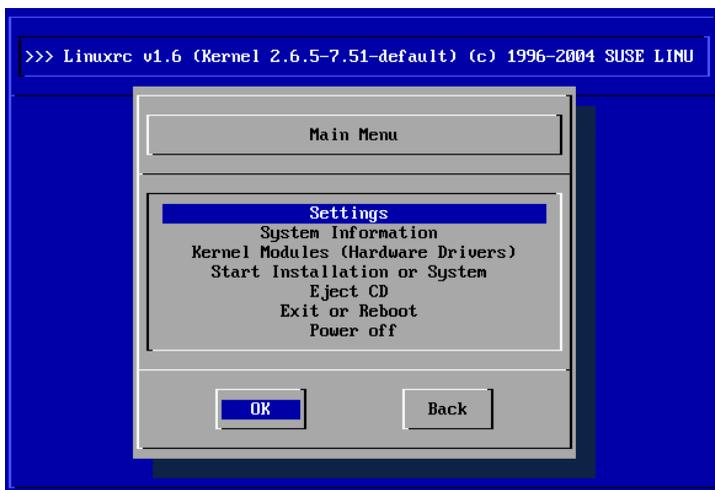


Figure 3.2: The linuxrc Main Menu

If you want to integrate a SCSI adapter in your system, load the respective SCSI module (see *Loading Modules* on the next page). As much as possible, these modules are precompiled in the kernels shipped by SUSE. When a SCSI adapter and connected devices are detected, messages such as the following are displayed:

```
SCSI subsystem initialized
scsi0 : Adaptec AIC7XXX EISA/VLB/PCI SCSI HBA DRIVER, Rev 6.2.36
        <Adaptec aic7890/91 Ultra2 SCSI adapter>
        aic7890/91: Ultra2 Wide Channel A, SCSI Id=7, 32/253 SCBs

(scsi0:A:0): 40.000MB/s transfers (20.000MHz, offset 15, 16bit)
  Vendor: IBM          Model: DCAS-34330W      Rev: S65A
  Type:   Direct-Access          ANSI SCSI revision: 02
scsi0:A:0:0: Tagged Queuing enabled.  Depth 32
SCSI device sda: 8467200 512-byte hdwr sectors (4335 MB)
SCSI device sda: drive cache: write back
  sda: sdal sda2
Attached scsi disk sda at scsi0, channel 0, id 0, lun 0
(scsi0:A:6): 20.000MB/s transfers (20.000MHz, offset 16)
  Vendor: TEAC         Model: CD-ROM CD-532S   Rev: 1.0A
  Type:   CD-ROM              ANSI SCSI revision: 02
```

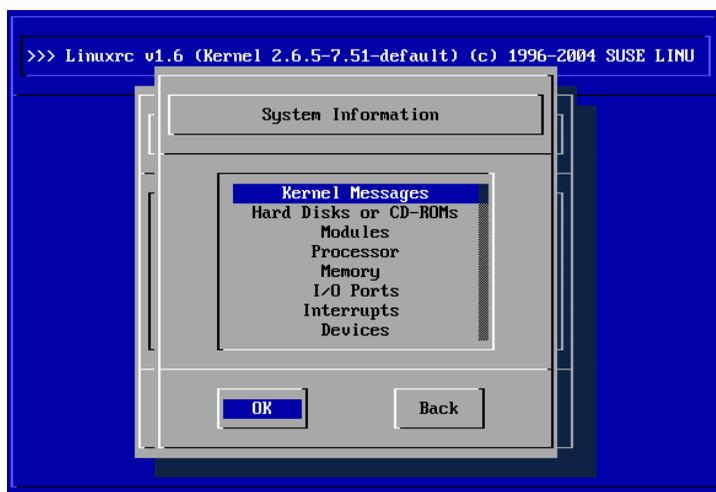


Figure 3.3: System Information

3.1.3 Loading Modules

Select the modules (drivers) needed. linuxrc offers the available drivers in a list. The name of the 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.

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.

In many cases, it is not necessary to specify the hardware in detail, because 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, because they wait until all attached devices respond.

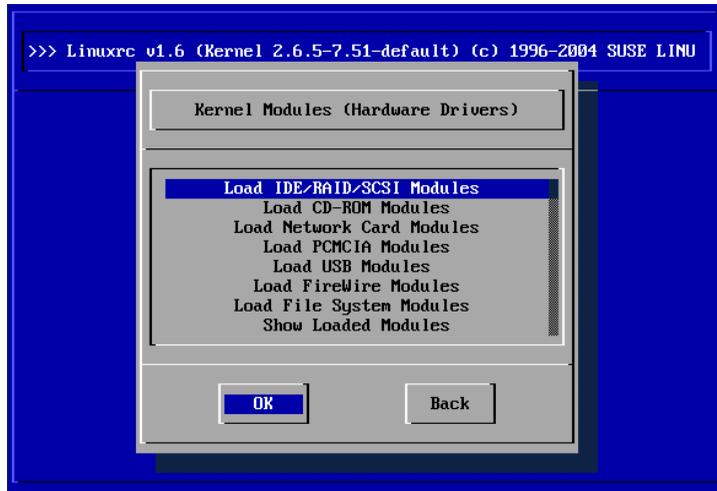


Figure 3.4: Loading Modules

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.

Note

If no driver is included for your installation device (proprietary or parallel port CD-ROM drive, network card, or 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 *Tips and Tricks* on page 116). 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

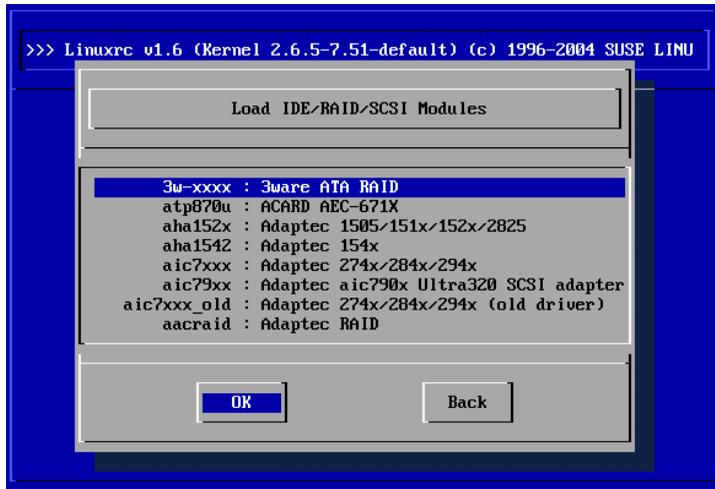


Figure 3.5: Selecting SCSI Drivers

3.1.5 Start Installation or System

After setting up the kernel support for your hardware to the extent that is needed for the installation, 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 *The SUSE Rescue System* on page 172), and 'Eject CD'.

To begin the installation, select 'Start Installation or Update' from the menu and press **(Enter)**. You are then prompted to select the installation source as shown in Figure 3.8 on page 110. In most cases, you can leave this at the preselected 'CD-ROM' item. However, other sources can be selected for installation and similarly for the rescue system. See *The SUSE Rescue System* on page 172) for information on the SUSE LINUX rescue system.

After pressing **(Enter)**, the installation environment loads from the selected medium. As soon as this process is completed, YaST starts and the installation begins.

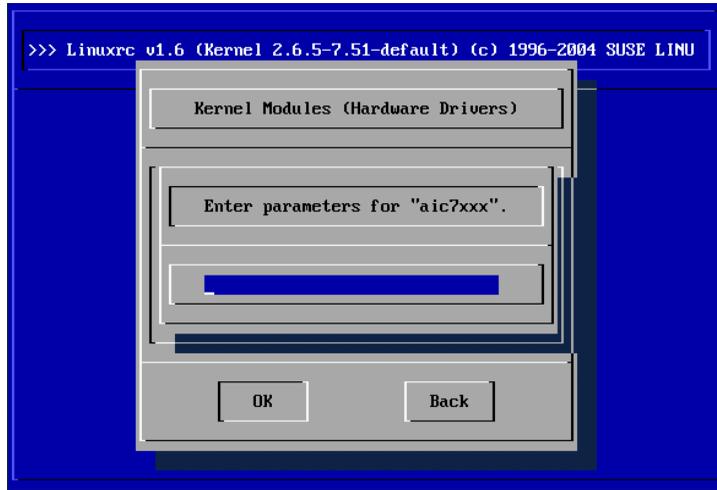


Figure 3.6: Entering Parameters for a Module

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 *ATAPI CD-ROM Hangs while Reading* on page 120.

The system hangs when loading data into a RAM disk.

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.

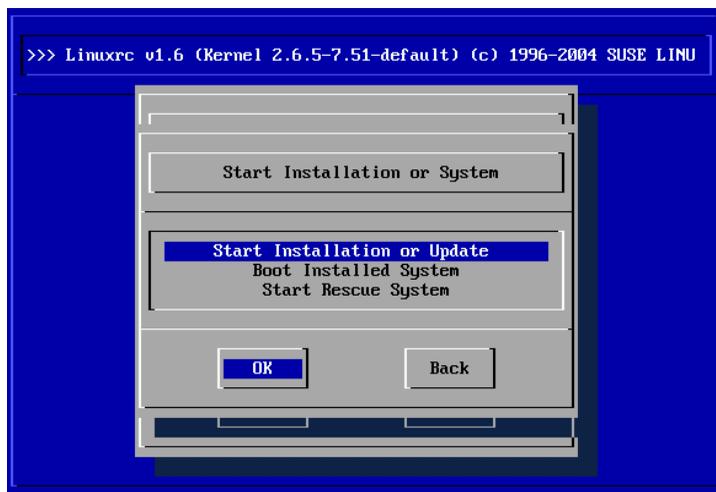


Figure 3.7: The *linuxrc* Installation Menu

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.) Specifies 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>`

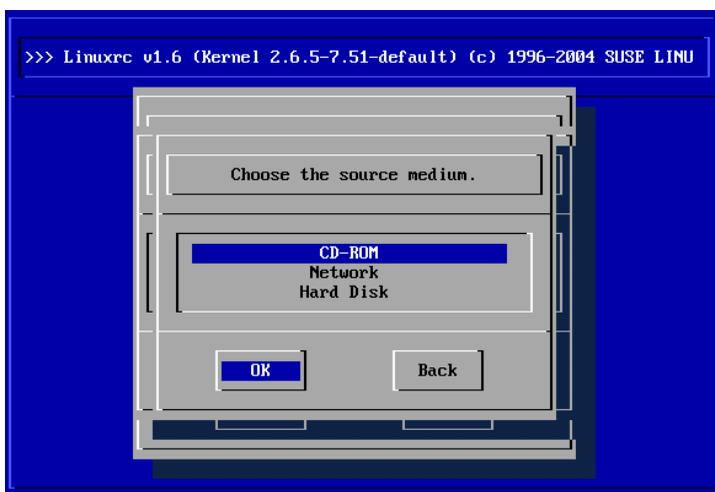


Figure 3.8: Selecting the Source Medium in linuxrc

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.

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 user `root` to access `linuxrc`.

Insmod: <module parameters> This specifies a module the kernel should load and 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 accessed 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 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

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=<source>
```

`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 *SLP Services in the Network* on page 435.

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 `xorg-x11-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 address of the installation system>: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, such as for server systems) or if the graphics card is not supported by the X Window System. Using this installation mode, visually impaired users can install SUSE LINUX with the aid of suitable output devices.

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 the installed system from starting automatically. If your hardware requires special parameters, which is not usually the case, enter these in **Boot Options**. If you select the language of your keyboard as the installation language, the keyboard layout will be correct. This facilitates the input of parameters.

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 *ATAPI CD-ROM Hangs while Reading* on page 120. 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 Disables PCI IRQ routing of the new ACPI system.

In this connection, search for Support Database articles with the keyword “acpi” at <https://portal.suse.com>.

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 in the Support Database under the keyword “memtest86”. 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.

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 allow selection from various operating systems prior to booting. The boot loader can either be configured during the installation or later with the help of YaST.

Boot Disk You can boot Linux from a *boot disk*. This approach only works if your system has a floppy drive. The boot disk can be created with YaST. See *Creating a Boot, Rescue, or Module Disk* on page 89.

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.

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. Solve this problem by entering the BIOS and looking for corresponding adjustable settings. For example, 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 is available in *Booting and Boot Managers* on page 185.

3.4.1 The Graphical SUSE Screen

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

3.4.2 Disabling the SUSE Screen

There are three ways to disable the SUSE screen:

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. *Booting and Boot Managers* on page 185 provides more information about this. However, if you prefer the text mode, which was the default in earlier versions, set `vga=normal`.

Completely disabling the SUSE screen

Compile a new kernel and disable the option 'Use splash screen instead of boot logo' in 'framebuffer support'.

Note

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

Note

3.5 Special Installation Procedures

3.5.1 Installation without CD-ROM Support

Sometimes a standard installation using a CD-ROM drive is not possible. For example, your CD-ROM may not be supported because it is an older proprietary drive. A secondary machine, like a laptop, might not have a CD-ROM drive at all, but only an ethernet adapter. SUSE LINUX offers the possibility of performing the installation on machines without a CD-ROM drive over a network connection. Usually this is done by means of NFS or FTP over ethernet. *Installation from a Network Source* on this page describes the procedure.

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, CD, or the network and the network must be configured.

The installation source can be made available over various protocols, such as NFS and FTP. See *Passing Parameters to linuxrc* on page 109 for information about the actual installation.

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. 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. You need at least 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 to create 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 :`, `rwritE` 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. 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 *Compiling the Kernel* on page 211):

```
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 *Creating a Boot Disk in DOS* on page 116 or *Creating a Boot Disk in a UNIX-Type System* on the facing 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 support and boots the kernel accordingly.
2. The monitor data (DDC info) is read.
3. The first block of the first hard disk (MBR) is read to map BIOS IDs to Linux device names during the boot loader configuration. The program attempts to read the block by means of the the lba32 functions of the BIOS to determine if the BIOS supports these functions.

Note

If you keep `(Shift)` pressed when SYSLINUX starts, all these steps are skipped. For troubleshooting purposes, insert the line

```
verbose 1
```

in `syslinux.cfg` for the boot loader to display which action is currently being performed.

Note

If the machine does not boot from the floppy disk, you may need to change the boot sequence in the BIOS to `A, C, CDROM`.

► x86

On x86 systems, 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). ◀

3.6.4 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, the installation is not possible. In this case, check if the manufacturer of your SCSI controller offers drivers for Linux.
- Many vendor-specific CD-ROM drives are supported in Linux. Nevertheless, problems may be encountered with this kind of drive. 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 *Booting from a Floppy Disk (SYSLINUX)* on the page before. Before booting 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. Additional devices should continue in this pattern.

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 *misjumped*. This means it is jumped 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:

hd x =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 a 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 the 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 is assigned a new name and the entry in the mount table `/etc/fstab` does not match.

To avoid this problem, the system start-up script `boot.scsidev` could be used. Enable this script using `/sbin/insserv` and set parameters for it in `/etc/sysconfig/scsidev`. The script `/etc/rc.d/boot.scsidev` handles the setup of the SCSI devices during the boot procedure and enters permanent device names under `/dev/scsi/`. These names can then be used in `/etc/fstab`. In addition, `/etc/scsi.alias` can be used to define persistent names for the SCSI configuration. See also `man scsidev`.

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`.

Note**Device Names and `\}{protect\}{dbapplication{udev}`**

For SUSE LINUX, although `boot.scsidev` is still supported, the preferred way to create persistent device names is to use `udev` to create device nodes with persistent names in `/dev/by-id/`.

Note

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 are their sizes and types (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 a reasonable value. Never configure your system without any swap memory.

If you use hibernation (suspend to disk), the content of the main memory is saved to the swap partition. In this case, the swap partition must be larger than the main 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 twenty-five users. If the average user requires 1000–1500 MB for personal data, a 40 GB partition mounted under `/home` is probably sufficient. For fifty users, you would need 80 GB. If possible, split `/home` to two 40 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 for a basic server without X).

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 excerpt of the output of the command `df` in Example 3.1 on the facing page.

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 *The free Command* on page 220.

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 this YaST module.

Note

Background information and partitioning tips can be found in *Partitioning for Experts* on page 122.

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 must 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 section always refers to LVM version 2.

Instead of LVM2, you can use EVMS (enterprise volume management system), which offers a uniform interface for logical volumes and 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. Because 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 (LVs) 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, an 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 by combining LVM with 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 the physical hard disk. Another advantage of 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 selecting '0x8e Linux LVM'. To continue partitioning with LVM, click 'LVM...'

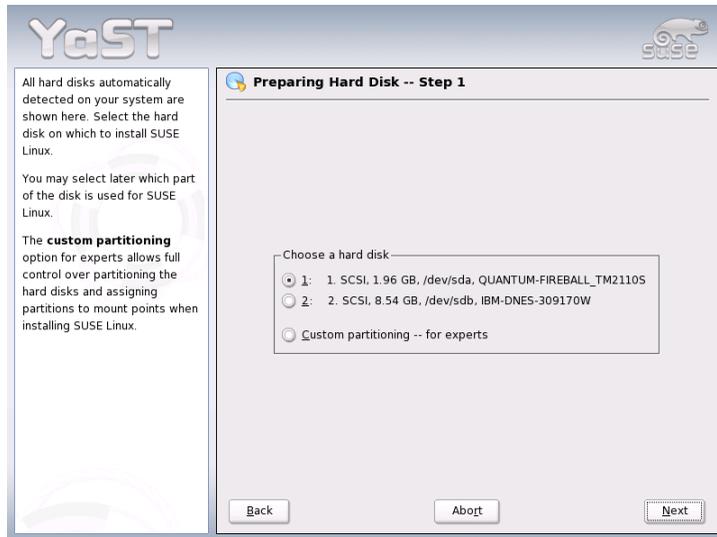


Figure 3.9: Activating LVM during Installation

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 identifier 8E. These partitions are indicated with "Linux LVM" in the partition list.

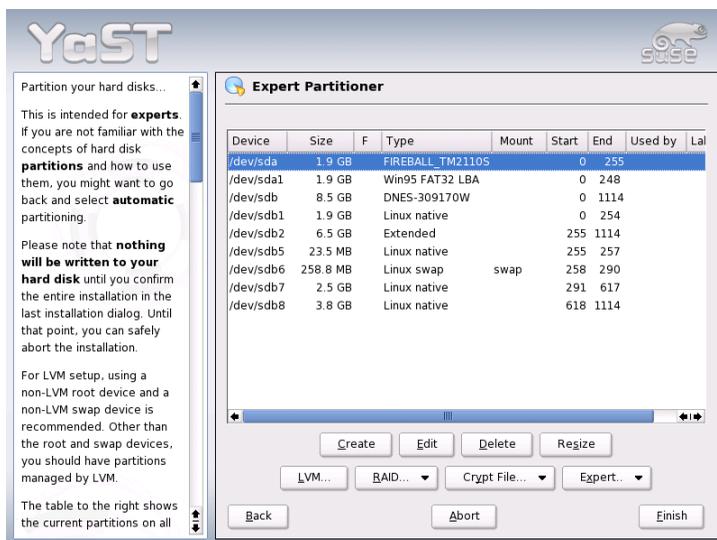


Figure 3.10: YaST: LVM Partitioner

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 the 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

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 LVM. If this is successfully activated, any disks containing a partition belonging to an activated volume group cannot be repartitioned. The Linux kernel refuses to read the modified partitioning of a hard disk as long as any 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 'Edit'. In view of the flexibility of LVM, it is recommended to 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', manage the LVM volume groups. If no volume group exists on your system yet, 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 4 MB. 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).

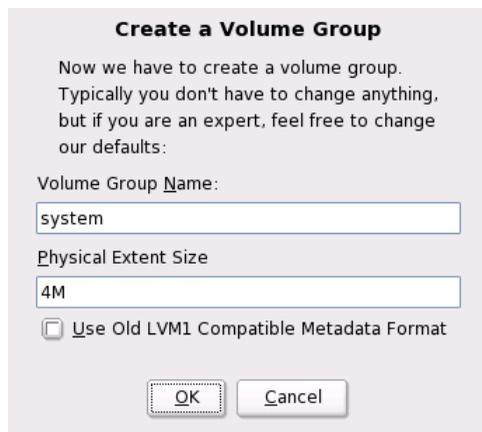


Figure 3.11: Adding a Volume Group

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 with “--”.

Modify the current volume group in the selection box to the upper left. The buttons in the upper right enable creation of additional volume groups and deletion of existing volume groups. Only volume groups 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).

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

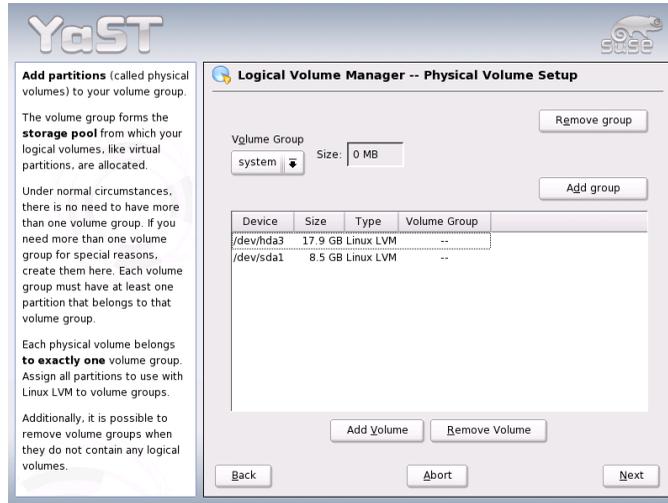


Figure 3.12: Partition List

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, such as reiserfs or ext2, is created on a logical volume and is then designated a mount point. The files stored on this logical volume can be found at this mount point on the installed system. All normal Linux partitions to which a mount point is assigned, all swap partitions, and all already existing logical volumes are listed here.

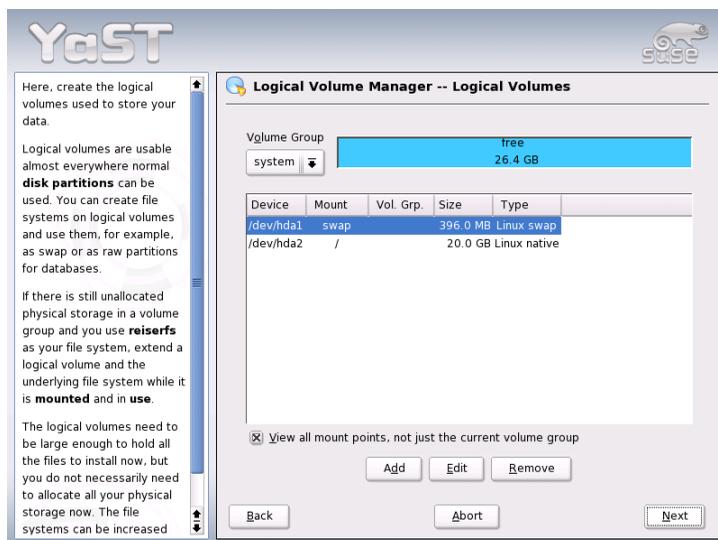


Figure 3.13: Logical Volume Management

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 (like reiserfs or ext2), and the mount point (for example, /var, /usr, /home).

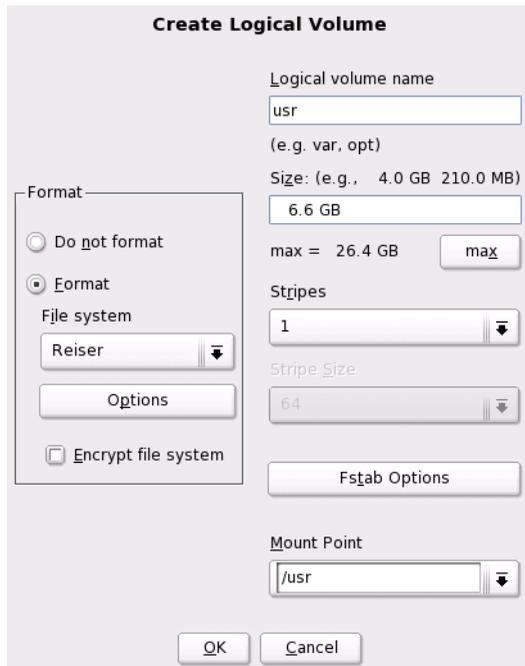


Figure 3.14: Creating Logical Volumes

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.

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 in order to optimize the performance and data security. Using this method, however, one advantage is sacrificed for another. The *RAID level* defines how the hard disks are connected and how the RAID controller addresses the hard disks. Most RAID controllers use the SCSI protocol because it can address a larger number of hard disks in a more effective way than the IDE protocol and is more suitable for parallel processing of commands. Meanwhile, there are some RAID controllers that support IDE or SATA hard disks. Refer to the Hardware Database at <http://cdb.suse.de>.

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, but the RAID system is destroyed and your data lost if even one hard disk fails.

RAID 1 This level provides adequate security for your data, because the data is copied to another hard disk 1:1. This is known as *hard disk mirroring*. If a disk is destroyed, a copy of its contents is available on another one. All of them except one could be damaged without endangering your data. The writing performance suffers a little in the copying process compared to 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 XOR, 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, see the `/dev/md0` device and others indicated with *RAID* on the expert page in the partitioning module.

3.11.3 Troubleshooting

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

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>.

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.

4.1	Updating SUSE LINUX	140
4.2	Software Changes from Version to Version	142
4.3	RPM — the Package Manager	156

4.1 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 or, in DOS or Windows, the `README.DOS` file on the CD. This file contains any changes made after this manual went to print.

Note

4.1.1 Preparations

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

Before starting your update, make note of the root partition. The command `df /` lists the device name of the root partition. In Example 4.1, the root partition to write down is `/dev/hda2` (mounted as `/`).

Example 4.1: List with `df -h`

Filesystem	Size	Used	Avail	Use%	Mounted on
<code>/dev/hda1</code>	1.9G	189M	1.7G	10%	<code>/dos</code>
<code>/dev/hda2</code>	8.9G	7.1G	1.4G	84%	<code>/</code>
<code>/dev/hda5</code>	9.5G	8.3G	829M	92%	<code>/home</code>

4.1.2 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 only necessary if you actually used PostgreSQL prior to your update.

4.1.3 Updating with YaST

Following the preparation procedure outlined in *Preparations* on the facing page, you can now update your system:

1. Boot the system as for the installation, described in *System Start-up for Installation* on page 8. 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 *Preparations* on the facing page). 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, for example, 'Default System'. Adjustments can be made later with YaST.

If you encounter difficulties with the automatic hardware detection of YaST, initialize the update from `linuxrc`. See *linuxrc* on page 102 for information.

4.1.4 Updating Individual Packages

Regardless of your overall updated environment, you can always update individual packages. From this point on, however, it is your responsibility to ensure that your system remains consistent. Update advice can be found at <http://www.suse.de/en/support/download/updates/>.

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

4.2 Software Changes from Version to Version

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

Problems and special issues of the respective versions are published online as they are identified. See the links listed below. Important updates of individual packages can be accessed at <http://www.suse.de/en/support/download/>.

4.2.1 From 8.0 to 8.1

Problems and Special Issues:

<http://portal.suse.com/sdb/en/2002/10/bugs81.html>

- Changes in the user and group names: To comply with UnitedLinux, some entries in `/etc/passwd` and `/etc/group` have been adjusted accordingly.
 - ▷ Changed users: `ftp` is now in group `ftp` (not in `daemon`).
 - ▷ Renamed groups: `www` (was `wwwadmin`), `games` (was `game`).

- ▷ New groups: `ftp` (with GID 50), `floppy` (with GID 19), `cdrom` (with GID 20), `console` (with GID 21), `utmp` (with GID 22).
- Changes related to FHS (see *Standards and Specifications* on page 656):
 - ▷ An example environment for `httpd` (Apache) is created at `/srv/www` (formerly `/usr/local/httpd`).
 - ▷ An example environment for FTP is created at `/srv/ftp` (formerly `/usr/local/ftp`). `ftplib` is required for this purpose.
- To ease access to the desired software, the single packages are not stored in a few bulky series any more, but rather in accessible *RPM groups*. This also means the CDs no longer contain cryptic directories under `suse`, but only few directories named after architectures, for example, `ppc`, `i586`, or `noarch`.
- The installation status of the following programs has changed for a new installation:
 - ▷ The boot loader GRUB, which offers more possibilities than LILO, is installed by default. However, LILO is retained if the system is updated.
 - ▷ The mailer `postfix` instead of `sendmail`.
 - ▷ The modern mailing list application `mailman` is installed instead of `majordomo`.
 - ▷ Select `hardened_suse` manually if needed and read the enclosed documentation.
- Split packages: `rpm` to `rpm` and `rpm-devel`, `popt` to `popt` and `popt-devel`, `libz` to `zlib` and `zlib-devel`.
`yast2-trans-*` is now split by languages: `yast2-trans-cs` (Czech), `yast2-trans-de` (German), `yast2-trans-es` (Spanish), etc. Not all languages are installed to save hard disk space. Install the required packages for the YaST language support as needed.
- Renamed packages: `bzip` to `bzip2`.
- Packages no longer supplied: `openldap`, use `openldap2`; `su1`, use `sudo`.

4.2.2 From 8.1 to 8.2

Problems and Special Issues: <http://portal.suse.com/sdb/en/2003/04/bugs82.html>

- 3D support for nVidia-based graphics cards (changes): The RPM `NVIDIA_GLX/NVIDIA_kernel` (including the script `switch2nvidia_glx`) is no longer included. Download the nVidia installer for Linux IA32 from the nVidia web site (<http://www.nvidia.com>), install the driver with this installer, and use `SOX2` or `YaST` to activate 3D support.
- During a new installation, `xinetd` is installed instead of `inetd` and configured with secure values. See the directory `/etc/xinetd.d`. However, during a system update, `inetd` is retained.
- The current PostgreSQL version is 7.3. When switching from version 7.2.x, perform a *dump/restore* with `pg_dump`. If your application queries the system catalogs, additional adaptations are necessary, because schemas were introduced in version 7.3. For more information, see http://www.ca.postgresql.org/docs/momjian/upgrade_tips_7.3.
- Version 4 of `stunnel` no longer supports any command-line options. However, the enclosed script `/usr/sbin/stunnel3_wrapper` can convert the command-line options into a configuration file that is suitable for `stunnel` and use it when the program is started (replace `OPTIONS` with your options):

```
/usr/sbin/stunnel3_wrapper stunnel OPTIONS
```

The generated configuration file will be printed to the default output, enabling the use of these specifications for generating a permanent configuration file.

- `openjade` (`openjade`) is the DSSSL engine currently used instead of `jade` (`jade_dsl`) when `db2x.sh` (`docbook-toys`) is run. For compatibility reasons, the individual programs are also available without the prefix `o`.

If your own applications depend on the directory `jade_dsl` and the files previously installed there, adapt them to the new directory `/usr/share/sgml/openjade` or create a link as root with `cd /usr/share/sgml rm jade_dsl ln -s openjade jade_dsl`.

To avoid a conflict with `rsz`, the command-line tool `sx` continues to be called `s2x`, `sgml2xml`, or `osx`.

4.2.3 From 8.2 to 9.0

Problems and Special Issues: <http://portal.suse.com/sdb/en/2003/07/bugs90.html>

- Version 4 of the RPM package manager is now available. The functionality for building packages has been shifted to the separate program `rpmbuild`. `rpm` continues to be used for the installation, updates, and database queries. See *RPM — the Package Manager* on page 156.
- The package `foomatic-filters` is now available for printing. The content was split from `cups-drivers`, because it can be used for printing even if CUPS is not installed. In this way, YaST supports configurations that are independent of the print system (CUPS, LPRng). The configuration file for this package is `/etc/foomatic/filter.conf`.
- The packages `foomatic-filters` and `cups-drivers` are now also required for LPRng and `lpdfilter`.
- The XML resources of the enclosed software packages can be accessed by means of the entries in `/etc/xml/suse-catalog.xml`. This file should not be edited with `xmlcatalog`, because this would result in the deletion of structural comments required for correct updates. `/etc/xml/suse-catalog.xml` is accessed by means of a `nextCatalog` statement in `/etc/xml/catalog`, enabling XML tools like `xmllint` or `xsltproc` to find the local resources automatically.

4.2.4 From 9.0 to 9.1

Refer to the article “Known Problems and Special Features in SUSE LINUX 9.1” in the SUSE Support Database at <http://portal.suse.com> under the keyword *special features*. These articles are published for every SUSE LINUX version.

Upgrading to Kernel 2.6

SUSE LINUX is now based entirely on kernel 2.6. The predecessor version 2.4 can no longer be used, because the enclosed applications do not work with kernel 2.4. 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 tries to convert the file (also see 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.

Standby and Suspend with ACPI

The new kernel 2.6 now supports standby and suspend with ACPI. This function is still in an experimental stage and may not be supported by some hardware components. To use this function, you need the `powersave` package. Information about this package is available in `/usr/share/doc/packages/powersave`. A graphical front-end is available in the `kpowersave` package.

Input Devices

Regarding the changes in connection with the input devices, refer to the above-mentioned Portal article “Known Problems and Special Features in SUSE LINUX 9.1” in the Support Database at <http://portal.suse.com> under the keyword *special features*.

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, because 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, 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 is included in SUSE LINUX 9.1 as the thread package. NPTL is binary-compatible with the older `linuxthreads` library. However, areas in which `linuxthreads` violates the POSIX standard require NPTL adaptations. This includes the following: signal handling, `getpid` returning the same value in all threads, and thread handlers registered with `pthread_atfork` not working 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 nonexistent 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. 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`.

Sound Configuration

Following an update, the sound cards must be reconfigured. This can be done with the YaST sound module. As `root`, enter the command `yast2 sound`.

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, such as `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 file 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 (free of charge, but registration is required) at <http://www.unix.org>.

Table 4.1: Comparison POSIX 1992 vs. POSIX 2001

POSIX 1992	POSIX 2001
<code>chown tux.users</code>	<code>chown tux:users</code>
<code>tail +3</code>	<code>tail -n +3</code>
<code>head -1</code>	<code>head -n 1</code>
<code>sort +3</code>	<code>sort -k +3</code>
<code>nice -10</code>	<code>nice -n 10</code>
<code>split -10</code>	<code>split -l 10</code>

Note

Third-party software may not yet comply with the new standard. In this case, set the environment variable as described above.

Note

/etc/gshadow Obsolete

`/etc/gshadow` has been abandoned and removed, because 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

Because 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 standard-noncompliant 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, 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. Detailed documentation for version 2.0 is available at the web page `http://httpd.apache.org/docs-2.0/en/`. On a system with an HTTP server installation, an update removes the Apache package and installs Apache 2. Subsequently, the system must be adapted with YaST or manually. The configuration files in `/etc/httpd` are now located in `/etc/apache2`.

Either threads or processes can be selected for handling multiple concurrent queries. The process management has been moved to an independent module, the multiprocessing module (MPM). Accordingly, Apache 2 needs either the `apache2-prefork` package (recommended for stability) or the `apache2-worker` package. Depending on the MPM, Apache 2 reacts differently to queries. This affects the performance as well as the use of modules. These characteristics are discussed in detail in *Threads* on page 511.

Apache 2 now supports the next-generation Internet protocol IPv6.

A mechanism has been implemented that enables module programmers to specify the desired loading sequence of the modules, relieving users of this task. The sequence in which modules are executed is often important and used to be determined by means of the loading sequence. For instance, a module that only gives authenticated users access to certain resources must be loaded first to prevent users without access permissions from seeing the pages.

Queries to and responses from Apache can be processed with filters.

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 still 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, because 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 (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`, because `libiodbc` has been discarded.

XML Resources in `/usr/share/xml`

FHS (see *Standards and Specifications* on page 656) 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 and 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`. To mount the medium, simply change to the respective device directory in `/media`. Media cannot be ejected as long as they are accessed by a program.

4.2.5 From 9.1 to 9.2

Refer to the article “Known Problems and Special Features in SUSE LINUX 9.2” in the SUSE Support Database at <http://portal.suse.com> under the keyword *special features*.

Activation of the Firewall in the Proposal Dialog during the Installation

To increase the security, the enclosed firewall solution SuSEFirewall2 is activated at the end of the installation in the proposal dialog. This means that all ports are closed initially and can be opened in the proposal dialog if necessary.

If network access is required during the installation or configuration of a service, the respective YaST module opens the needed TCP and UDP ports of all internal and external interfaces. If this is not desired, the user can close the ports in the YaST module or specify other detailed firewall settings.

Table 4.2: Ports Used by Important Services

Service	Ports
HTTP server	Firewall is set up according to the “list” statements (TCP only)
Mail (postfix)	smtp 25/TCP
Samba server	netbios-ns 137/TCP; netbios-dgm 138/TCP; netbios-ssn 139/TCP; microsoft-ds 445/TCP
DHCP server	bootpc 68/TCP
DNS server	domain 53/TCP; domain 53/UDP
DNS server	Plus special support for port mapper in SuSEFirewall2
Port mapper	sunrpc 111/TCP; sunrpc 111/UDP
NFS server	nfs 2049/TCP
NFS server	Plus port mapper
NIS server	Activates portmap
TFTP	tftp 69/TCP
CUPS (IPP)	ipp 631/TCP; ipp 631/UDP

Print System Configuration

At the end of the installation (proposal dialog), the ports needed for the print system must be open in the firewall configuration. Port 631/TCP and port 631/UDP are needed for CUPS and should not be closed for normal operation. Port 515/TCP (for the old LPD protocol) and the ports used by Samba must also be open for printing via LPD or SMB.

Change to X.Org

The change from XFree86 to X.Org is facilitated by compatibility links that enable access to important files and commands with the old names.

Table 4.3: Commands

XFree86	X.Org
XFree86	Xorg
xf86config	xorgconfig
xf86cfg	xorgcfg

Table 4.4: Log Files in /var/log

XFree86	X.Org
XFree86.0.log	Xorg.0.log
XFree86.0.log.old	Xorg.0.log.old

In the course of the change to X.Org, the packages were renamed from XFree86* to xorg-x11*.

Changes in the powersave Package

The configuration files in `/etc/sysconfig/powersave` have changed:

Table 4.5: Split Configuration Files in /etc/sysconfig/powersave

Old	Now split into
/etc/sysconfig/powersave/common	common
	cpufreq
	events
	battery
	sleep
	thermal

`/etc/powersave.conf` has become obsolete. Existing variables have been moved to the files listed in Table 4.5. If you changed the “event” variables in `/etc/powersave.conf`, these must now be adapted in `/etc/sysconfig/powersave/events`.

The names of sleep states has changed from:

- suspend (ACPI S4, APM suspend)
- standby (ACPI S3, APM standby)

To:

- suspend to disk (ACPI S4, APM suspend)
- suspend to ram (ACPI S3, APM suspend)
- standby (ACPI S1, APM standby)

OpenOffice.org (OOo)

Paths: OOo is now installed in `/usr/lib/ooo-1.1` instead of `/opt/OpenOffice.org`. The default directory for user settings is now `~/.ooo-1.1` instead of `~/OpenOffice.org1.1`.

Wrapper: There are some new wrappers for starting the OOo components. The new names are shown Table 4.6 on the next page.

Table 4.6: Wrapper

Old	New
/usr/X11R6/bin/OOo-calc	/usr/bin/oocalc
/usr/X11R6/bin/OOo-draw	/usr/bin/oodraw
/usr/X11R6/bin/OOo-impress	/usr/bin/ooimpress
/usr/X11R6/bin/OOo-math	/usr/bin/oomath
/usr/X11R6/bin/OOo-padmin	/usr/sbin/oopadmin
/usr/X11R6/bin/OOo-setup	-
/usr/X11R6/bin/OOo-template	/usr/bin/oofromtemplate
/usr/X11R6/bin/OOo-web	/usr/bin/ooweb
/usr/X11R6/bin/OOo-writer	/usr/bin/oowriter
/usr/X11R6/bin/OOo	/usr/bin/ooffice
/usr/X11R6/bin/OOo-wrapper	/usr/bin/ooo-wrapper

The wrapper now supports the option `--icons-set` for switching between KDE and GNOME icons. The following options are no longer supported: `--default-configuration`, `--gui`, `--java-path`, `--skip-check`, `--lang` (the language is now determined by means of locales), `--messages-in-window`, and `--quiet`.

KDE and GNOME Support: KDE and GNOME extensions are available in the `OpenOffice_org-kde` and `OpenOffice_org-gnome` packages.

Sound Mixer `kmix`

The sound mixer `kmix` is preset as the default. For high-end hardware, there are other mixers, like `QAMix/KAMix`, `envy24control` (only ICE1712), or `hdspmixer` (only RME Hammerfall).

4.3 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 checking of packages; and signing packages. `rpmbuild` can be used to build installable packages from pristine sources.

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

`rpm` can be used to administer LSB-compliant packages. Refer to *Standards and Specifications* on page 656 for 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

4.3.1 Verifying Package Authenticity

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

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

The command `rpm --checksig 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.

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

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

- If a configuration file was not changed by the system administrator, `rpm` installs the new version of the appropriate file. No action by the system administrator is required.
- If a configuration file was changed by the system administrator before the update, `rpm` saves the changed file with the extension `.rpmorig` or `.rpmsave` (backup file) and installs the version from the new package, but only if the originally installed file and the newer version are different. If this is the case, compare the backup file (`.rpmorig` or `.rpmsave`) with the newly installed file and make your changes again in the new file. Afterwards, be sure to delete all `.rpmorig` and `.rpmsave` files to avoid problems with future updates.

- `.rpmnew` files appear if the configuration file already exists *and* if the `noreplace` label was specified in the `.spec` file.

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

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

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

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

4.3.3 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 -qpP1 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 -qP1 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`.

4.3.4 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 4.7.

Table 4.7: The Most Important RPM Query Options

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

For example, the command `rpm -q -i wget` displays the information shown in Example 4.2 on the facing page.

Example 4.2: 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 Example 4.3. Pass the partial file name to the script shown as a parameter when running it.

Example 4.3: 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. Initiate these with `-V`, `-y`, or `--verify`. With this option, `rpm` shows all files in a package that have been changed since installation. `rpm` uses eight character symbols to give some hints about the following changes:

Table 4.8: RPM Verify Options

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

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

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

The files of the RPM database are placed in `/var/lib/rpm`. If the partition `/usr` has a size of 1 GB, this database can occupy nearly 30 MB, especially after a complete update. If the database is much larger than expected, it is useful to rebuild the database with the option `--rebuilddb`. Before doing this, make a backup of the old database. The `cron` script `cron.daily` makes daily copies of the database (packed with `gzip`) and stores them in `/var/adm/backup/rpmdb`. The number of copies is controlled by the variable `MAX_RPMDL_BACKUPS` (default: 5) in `/etc/sysconfig/backup`. The size of a single backup is approximately 3 MB for 1 GB in `/usr`.

4.3.5 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. When you install a source package, only the source code is added to the system. The software itself must be compiled. 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.), because 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** Skip some steps.

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

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

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

4.3.7 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. Display the `HEADER` with `(F3)`. View the archive structure with the cursor keys and `(Enter)`. Copy archive components 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 also offers a feature for repairing the installed system. This chapter describes the various types and steps of system repair.

5.1	Starting YaST System Repair	168
5.2	Automatic Repair	169
5.3	User-Defined Repair	170
5.4	Expert Tools	171
5.5	The SUSE Rescue System	172

5.1 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 *Installation with YaST* on page 7 to get to the dialog offering the various installation options then select 'Repair Installed System'. This dialog is shown in Figure 5.1.

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

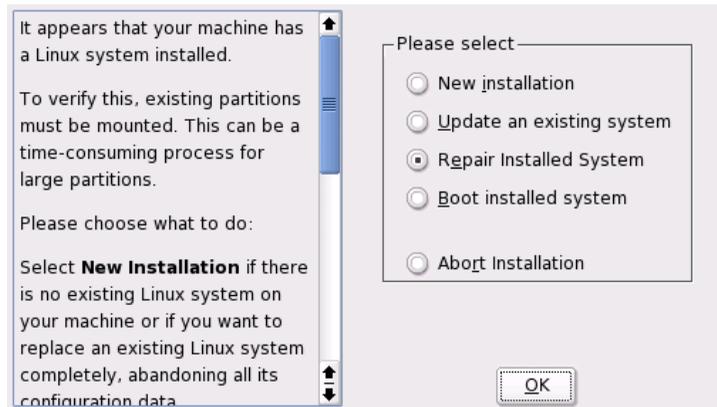


Figure 5.1: Selecting the YaST System Repair Utility

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.

5.2 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 takes quite some time due to the large number 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 5.2 on the following 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 also to analyze the base packages, this takes a long time because of their vast number.

Whenever an error is encountered, the procedure stops and a dialog opens, offering details and possible solutions. It is not possible to describe all these cases. Read the messages 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 case and no repair is ever performed automatically without prompting the user.

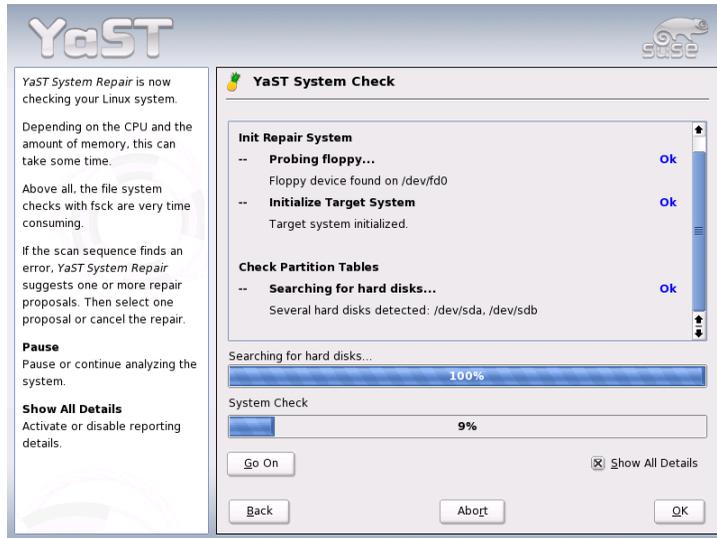


Figure 5.2: Automatic Repair Mode

5.3 User-Defined Repair

The automatic repair explained in the preceding section performs all tests. This is useful if the extent of the system damage is unknown. 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 number of necessary test runs.

5.4 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 *Configuring the Boot Loader with YaST* on page 198.

Run Partitioning Tool This starts the expert partitioning tool in YaST. Details can be found in *Expert Partitioning with YaST* on page 19.

Fix File System This checks the file systems of your installed system. You are first offered a selection of all detected partitions and can then choose the ones to check.

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 saves 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.

5.5 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. 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/>.

5.5.1 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 needed 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 109) and specify the desired source medium (Figure 5.3 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 *Installation from a Network Source* on page 116). A submenu offers protocols, such as NFS, FTP, and SMB (see Figure 5.4 on page 174).



Figure 5.3: Source Medium for the Rescue System

‘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 is loaded.

Regardless of the medium chosen, the rescue system is decompressed, loaded onto a RAM disk as a new root file system, mounted, and started. Now it is ready for use.

5.5.2 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.

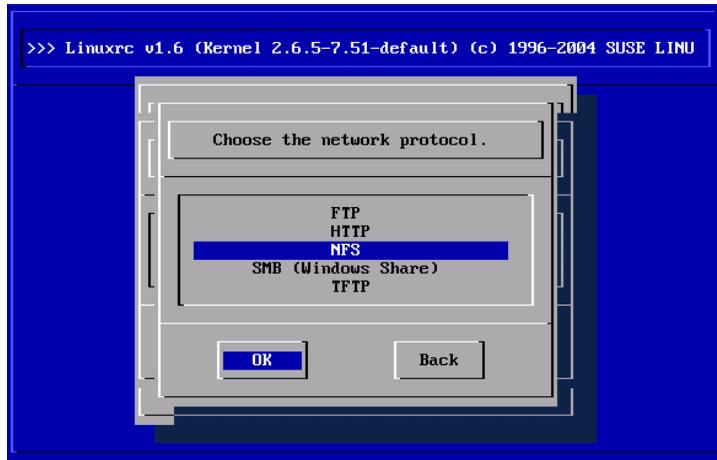


Figure 5.4: Network Protocols

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, including `reiserfsck` and `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 Example 5.1.

Example 5.1: 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, 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 recovering lost partitions with the `fdisk` program by simply setting them up again, 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 `reiserfsck`, `e2fsck`, and `dump_e2fs` (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 *Manual Page of reiserfsck* on page 659 and *Manual Page of e2fsck* on page 663.

If mounting an `ext2` 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, because the superblock copy is intact, everything is fine.

Part II

System

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

SUSE LINUX is available for several 64-bit platforms. This does not necessarily mean that all the applications included have already been ported to 64-bit platforms. SUSE LINUX supports the use of 32-bit applications in a 64-bit system environment. This chapter offers a brief overview of how this support is implemented on 64-bit SUSE LINUX platforms. It explains how 32-bit applications are executed (runtime support) and how 32-bit applications should be compiled to enable them to run both in 32-bit and 64-bit system environments. Additionally, find information about the Kernel API and an explanation of how 32-bit applications can run under a 64-bit kernel.

6.1	Runtime Support	180
6.2	Software Development	181
6.3	Software Compilation on Biarch Platforms	181
6.4	Kernel Specifications	182

SUSE LINUX for the 64-bit platforms AMD64 and EM64T is designed so that existing 32-bit applications run in the 64-bit environment “out-of-the-box.” This support means that you can continue to use your preferred 32-bit applications without waiting for a corresponding 64-bit port to become available.

6.1 Runtime Support

Note

Conflicts between Application Versions

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, 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.

To retain compatibility with the 32-bit version, the libraries are stored at the same place in the system as in the 32-bit environment. The 32-bit version of `libc.so.6` is located under `/lib/libc.so.6` in both the 32-bit and 64-bit environments.

All 64-bit libraries and object files are located in directories called `lib64`. The 64-bit object files you would normally expect to find under `/lib`, `/usr/lib`, and `/usr/X11R6/lib` are now found under `/lib64`, `/usr/lib64`, and `/usr/X11R6/lib64`. This means that there is space for the 32-bit libraries under `/lib`, `/usr/lib` and `/usr/X11R6/lib`, so the 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 are still 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).

6.2 Software Development

A biarch development toolchain allows generation of 32-bit and 64-bit objects. The default is to compile 64-bit objects. It is possible to generate 32-bit objects by using special flags. For GCC, this special flag is `-m32`.

All header files must be written in an architecture-independent form. The installed 32-bit and 64-bit libraries must have an API (application programming interface) that matches the installed header files. The normal SUSE environment is designed according to this principle. In the case of manually updated libraries, resolve these issues yourself.

6.3 Software Compilation on Biarch Platforms

To develop binaries for the other architecture on a biarch architecture, the respective libraries for the second architecture must additionally be installed. These packages are called `rpmname-32bit`. You also need the respective headers and libraries from the `rpmname-devel` packages and the development libraries for the second architecture from `rpmname-devel-32bit`.

Most Open Source programs use an `autoconf`-based program configuration. To use `autoconf` for configuring a program for the second architecture, overwrite the normal compiler and linker settings of `autoconf` by running the `configure` script with additional environment variables.

The following example refers to an AMD64 or EM64T system with x86 as the second architecture:

1. Set `autoconf` to use the 32-bit compiler:

```
CC="gcc -m32"
```

2. Instruct the linker to process 32-bit objects:

```
LD="ld -m elf64_i386"
```

3. Set the assembler to generate 32-bit objects:

```
AS="gcc -c -m32"
```

4. Determine that the libraries for `libtool` and so on come from `/usr/lib`:

```
LDLDFLAGS="-L/usr/lib"
```

5. Determine that the libraries are stored in the `lib` subdirectory:

```
--libdir=/usr/lib
```

6. Determine that the 32-bit X libraries are used:

```
--x-libraries=/usr/X11R6/lib/
```

Not all of these variables are needed for every program. Adapt them to the respective program.

```
CC="gcc -m64" \
LDLDFLAGS="-L/usr/lib64;" \
    .configure \
    --prefix=/usr \
    --libdir=/usr/lib64
make
make install
```

6.4 Kernel Specifications

The 64-bit kernels for 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.

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, must be compiled as 64-bit programs to function properly.

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

Booting and Boot Managers

This chapter describes the boot process of your Linux system. Learn how to configure GRUB, the boot loader currently used in SUSE LINUX. A special YaST module is available for performing all settings. If you are not familiar with the subject of booting in Linux, read the following sections to acquire some background information. This chapter also describes some of the problems frequently encountered when booting with GRUB and their solutions.

7.1	The Boot Process	186
7.2	Boot Management	187
7.3	Selecting a Boot Loader	188
7.4	Booting with GRUB	189
7.5	Configuring the Boot Loader with YaST	198
7.6	Uninstalling the Linux Boot Loader	201
7.7	Creating Boot CDs	201
7.8	Troubleshooting	202
7.9	For More Information	204

7.1 The Boot Process

During the boot process, the system control passes from the BIOS to the boot loader to the operating system kernel in a three-stage process. After switching on the computer, the BIOS initializes the screen and the keyboard and tests the main memory. Up to this stage, the machine does not have access to any mass storage media. Subsequently, the information about the current date, time, and the most important peripherals are loaded from the CMOS values (*CMOS Setup*). When the first hard disk and its geometry is recognized, the system control passes from the BIOS to the boot loader.

The first physical 512-byte data sector of the first hard disk is loaded into the main memory and the program (the *boot loader*) at the beginning of this sector takes over. The sequence of the commands executed by the boot loader determines the remaining part of the boot process. Therefore, the first 512 bytes on the first hard disk are referred to as the *Master Boot Record*. Up to the point when the MBR is loaded, the boot process is the same on all computers regardless of the installed system and the computer only has the routines (drivers) stored in the BIOS for accessing peripheral devices.

Subsequently, the boot loader configuration determines which operating system should be booted with which options. In the final stage, the boot loader passes the system control to the actual operating system. As soon as the control passes to the operating system, all drivers contained in the operating system are available to support your hardware.

7.1.1 Master Boot Record

The structure of the MBR is defined by an operating system-independent convention. The first 446 bytes are reserved for program code. The next 64 bytes provide space for a partition table of up to four entries (see *Partitioning for Experts* on page 122). The partition table contains information about the partitioning of the hard disk and the file system type. The operating system needs this table for handling the hard disk. The last two bytes of the MBR must contain a static “magic number” (AA55). An MBR containing a different value is regarded as invalid by the BIOS and all PC operating systems.

7.1.2 Boot Sectors

Boot sectors are the first sectors of hard disk partitions except for the extended partition, which merely serves as a “container” for other partitions. These boot sectors have 512 bytes of space for code used to boot an operating system installed in the respective partition. This applies to boot sectors of formatted DOS, Windows, and OS/2 partitions, which also contain some important basic data of the file system. In contrast, the boot sectors of Linux partitions are initially empty after setting up a file system. Therefore, a Linux partition is *not bootable by itself*, even if it contains a kernel and a valid root file system. A boot sector with valid code for booting the system has the same magic number as the MBR in its last 2 bytes (AA55).

7.1.3 Booting DOS or Windows

If the MBR contains generic boot code, the system to boot can be determined by means of just one primary partition that is marked as active or bootable. Normally, the validity of the boot sector of this partition is also checked. After booting the system, another system can easily be selected with `fdisk`.

If a DOS or Windows partition is active, the boot sector loads the `.sys` drivers needed for booting the system. In DOS, only one primary partition can be marked as active. Therefore, the DOS system cannot be installed on logical drives in an extended partition.

Windows 2000 and XP can also be installed in a logical partition. Multiple Windows installations at the same time are even possible. However, the boot files are written to a primary partition. If another 2000 or XP system is installed, it is automatically added to the boot menu. Thus, the condition that Windows needs a primary partition is still valid.

7.2 Boot Management

In the easiest case — if only one operating system is installed on a computer — the boot management takes place as described above. If several operating systems are installed on a computer, the following options are available:

Booting Additional Systems from External Media

One of the operating systems is booted from the hard disk. The other operating systems are booted by means of a boot manager installed on an external medium (floppy disk, CD, USB storage medium). Because GRUB is able to boot all other operating systems, no external boot loader needs to be used.

Installing a Boot Manager in the MBR

A boot manager enables the concurrent installation and alternate use of several systems on one computer. The users can select the system to boot during the boot process. To change to a another system, the computer must be rebooted. This is only possible if the selected boot manager is compatible with the installed operating systems. GRUB, the boot manager used in SUSE LINUX, is able to boot all common operating systems. By default, SUSE LINUX installs the selected boot manager in the MBR.

7.3 Selecting a Boot Loader

By default, the boot loader GRUB is used in SUSE LINUX. However, in some cases and for special hardware and software constellations, LILO may be more suitable. If you update an older SUSE LINUX version that used LILO, LILO is installed. For a new installation, GRUB is installed, unless the root partition is installed on the following systems:

- CPU-dependent RAID controllers (such as many Promise or Highpoint controllers)
- Software RAID
- LVM

Information about the installation and configuration of LILO is available in the Support Database under the keyword “LILO”.

7.4 Booting with GRUB

GRUB (Grand Unified Bootloader) comprises two stages. `stage1` consists of 512 bytes and is written to the MBR or the boot sector of a hard disk partition or floppy disk. Subsequently, `stage2` is loaded. This stage contains the actual program code. The only task of the first stage is to load the second stage of the boot loader.

`stage2` is able to access file systems. Currently, Ext2, Ext3, ReiserFS, Minix, and the DOS FAT file system used by Windows are supported. To a certain extent, JFS, XFS, and UFS and FFS used by BSD systems are also supported. Since version 0.95, GRUB is also able to boot from a CD or DVD containing an ISO 9660 standard file system pursuant to the “El Torito” specification. Even before the system is booted, GRUB can access file systems of supported BIOS disk devices (floppy disks or hard disks, CD drives, and DVD drives detected by the BIOS). Therefore, changes to the GRUB configuration file (`menu.lst`) no longer require a reinstallation of the boot manager. When the system is booted, GRUB reloads the menu file together with the valid paths and partition data of the kernel or the initial RAM disk (`initrd`) and locates these files.

The actual configuration of GRUB is based on three files that are described below:

`/boot/grub/menu.lst` This file contains all information about partitions or operating systems that can be booted with GRUB. Without this information, the system control cannot be passed to the operating system.

`/boot/grub/device.map` This file translates the device names from the GRUB and BIOS notation to Linux device names.

`/etc/grub.conf` This file contains the parameters and options the GRUB shell needs for installing the boot loader correctly.

GRUB can be controlled in various ways. Boot entries from an existing configuration can be selected from the graphical menu (splash screen). The configuration is loaded as it is from the file `menu.lst`.

In GRUB, all boot parameters can be changed prior to booting. For example, errors made when editing the menu file can be corrected in this way. Boot commands can also be entered interactively over a kind of input prompt (see *Editing Menu Entries during the Boot Procedure* on page 193). GRUB offers the possibility of determining the location of the kernel and the `initrd` prior to booting. In this

way, you can even boot an installed operating system for which no entry exists in the boot loader configuration.

The *GRUB shell* provides an emulation of GRUB in the installed system. It can be used to install GRUB or test new settings before applying them. See *The GRUB Shell* on page 196.

7.4.1 The GRUB Boot Menu

The graphical splash screen with the boot menu is based on the GRUB configuration file `/boot/grub/menu.lst`, which contains all information about all partitions or operating systems that can be booted by means of the menu.

Every time the system is booted, GRUB loads the menu file from the file system. For this reason, GRUB does not need to be reinstalled after every change to the file. The YaST boot loader module can be used to modify the GRUB configuration as described in *Configuring the Boot Loader with YaST* on page 198.

The menu file contains commands. The syntax is very simple. Every line contains a command followed by optional parameters separated by spaces like in the shell. For historical reasons, some commands permit an `=` in front of the first parameter. Comments are introduced by a hash (`#`).

To identify the menu items in the menu overview, specify a `title` for every entry. The text (including any spaces) following the keyword `title` is displayed as a selectable option in the menu. All commands up to the next `title` are executed when this menu item is selected.

The simplest case is the redirection to boot loaders of other operating systems. The command is `chainloader` and the argument is usually the boot block of another partition in GRUB block notation. For example:

```
chainloader (hd0,3)+1
```

The device names in GRUB are explained in *Naming Conventions for Hard Disks and Partitions* on the facing page. The above example specifies the first block of the fourth partition of the first hard disk.

Use the command `kernel` to specify a kernel image. The first argument is the path to the kernel image in a partition. The other arguments are passed to the kernel on the command line.

If the kernel does not have built-in drivers for access to the root partition, `initrd` must be specified with a separate GRUB command whose only argument is the path to the `initrd` file. Because the loading address of the `initrd` is

written to the loaded kernel image, the command `initrd` must follow immediately after the `kernel` command.

The command `root` simplifies the specification of kernel and `initrd` files. The only argument of `root` is a GRUB device or a partition on a GRUB device. This device is used for all kernel, `initrd`, or other file paths for which no device is explicitly specified until the next `root` command. This command is not used in the `menu.lst` file generated during the installation. It merely facilitates manual editing.

The `boot` command is implied at the end of every menu entry, so it does not need to be written into the menu file. However, if you use GRUB interactively for booting, you must enter the `boot` command at the end. The command itself has no arguments. It merely boots the loaded kernel image or the specified chain loader.

After writing all menu entries, define one of them as the `default` entry. Otherwise, the first one (entry 0) is used. You can also specify a time-out in seconds after which the default entry should boot. `timeout` and `default` usually precede the menu entries. An example file is described in *An Example Menu File* on the next page.

Naming Conventions for Hard Disks and Partitions

The naming conventions GRUB uses for hard disks and partitions differ from those used for normal Linux devices. In GRUB, the numbering of the partitions starts with zero. Thus, `(hd0, 0)` is the first partition of the first hard disk. On a common desktop machine with a hard disk connected as primary master, the corresponding Linux device name is `/dev/hda1`.

The four possible primary partitions are assigned the partition numbers 0 to 3. The logical partitions are numbered from 4:

```
(hd0,0)  first primary partition of the first hard disk
(hd0,1)  second primary partition
(hd0,2)  third primary partition
(hd0,3)  fourth primary partition (usually an extended partition)
(hd0,4)  first logical partition
(hd0,5)  second logical partition
```

GRUB does not distinguish between IDE, SCSI, and RAID devices. All hard disks recognized by the BIOS or other controllers are numbered according to the boot sequence preset in the BIOS.

Unfortunately, GRUB is not able to map the Linux device names to BIOS device names exactly. It generates this mapping with a help of an algorithm and saves it to the file `device.map`, which can be edited if necessary. Information about the file `device.map` is available in *The File device.map* on page 194.

A complete GRUB path consists of a device name written in parentheses and the path to the file in the file system in the specified partition. The path begins with a slash. For example, the bootable kernel could be specified as follows on a system with a single IDE hard disk containing Linux in its first partition:

```
(hd0,0)/boot/vmlinuz
```

An Example Menu File

The following example shows the structure of a GRUB menu file. The example installation comprises a Linux boot partition under `/dev/hda5`, a root partition under `/dev/hda7`, and a Windows installation under `/dev/hda1`.

```
gfxmenu (hd0,4)/message
color white/blue black/light-gray
default 0
timeout 8

title linux
    kernel (hd0,4)/vmlinuz root=/dev/hda7 vga=791
    initrd (hd0,4)/initrd
title windows
    chainloader(hd0,0)+1
title floppy
    chainloader(fd0)+1
title failsafe
    kernel (hd0,4)/vmlinuz.shipped root=/dev/hda7 ide=nodma \
    apm=off acpi=off vga=normal nosmp maxcpus=0 3
    initrd (hd0,4)/initrd.shipped
```

The first block defines the configuration of the splash screen:

```
gfxmenu (hd0,4)/message The background image message is located in
/dev/hda5.
```

```
color white/blue black/light-gray
    Color scheme: white (foreground), blue (background), black (selection), and
```

light gray (background of the selection). The color scheme has no effect on the splash screen, only on the customizable GRUB menu that you can access by exiting the splash screen with `(Esc)`.

`default 0` The first menu entry `title linux` is the one to boot by default.

`timeout 8` After eight seconds without any user input, GRUB automatically boots the default entry.

The second and largest block lists the various bootable operating systems. The sections for the individual operating systems are introduced by `title`.

- The first entry (`title linux`) is responsible for booting SUSE LINUX. The kernel (`vmlinuz`) is located in the first logical partition (the boot partition) of the first hard disk. Kernel parameters, such as the root partition and VGA mode, are appended here. The root partition is specified according to the Linux naming convention (`/dev/hda7/`), because this information is read by the kernel and has nothing to do with GRUB. The `initrd` is also located in the first logical partition of the first hard disk.
- The second entry is responsible for loading Windows. Windows is booted from the first partition of the first hard disk (`hd0 , 0`). The command `chainloader +1` causes GRUB to read and execute the first sector of the specified partition.
- The next entry enables booting from floppy disk without modifying the BIOS settings.
- The boot option `failsafe` starts Linux with a selection of kernel parameters that enables Linux to boot even on problematic systems.

The menu file can be changed whenever necessary. GRUB then uses the modified settings during the next boot. Edit the file permanently using YaST or an editor of your choice. Alternatively, make temporary changes interactively using the edit function of GRUB (see *Editing Menu Entries during the Boot Procedure* on the current page).

Editing Menu Entries during the Boot Procedure

In the graphical GRUB boot menu, the operating system to boot can be selected with the arrow keys. If you select a Linux system, you can enter additional boot parameters at the boot prompt. To edit individual menu entries directly, press `(Esc)` to exit the splash screen and press `(E)`. Changes made in this way only apply to the current boot procedure and are not adopted permanently.

Note**Keyboard Layout during the Boot Procedure**

The US keyboard layout is the only one available when booting.

Note

After activating the editing mode, use the arrow keys to select the menu entry of which to edit the configuration. To make the configuration editable, press **(E)** once more. In this way, edit incorrect partition or path specifications before they have a negative effect on the boot process. Press **(Enter)** to exit the editing mode and return to the menu. Then press **(B)** to boot this entry. Further possible actions are displayed in the help text at the bottom.

To enter changed boot options permanently and pass them to the kernel, open the file menu `.lst` as the user `root` and append the respective kernel parameters to the existing line, separated by spaces:

```
title linux
kernel (hd0,0)/vmlinuz root=/dev/hda3 <additional parameter>
initrd (hd0,0)/initrd
```

GRUB automatically adopts the new parameters the next time the system is booted. Alternatively, this change can also be made with the YaST boot loader module. Append the new parameters to the existing line, separated by spaces.

7.4.2 The File `device.map`

The file `device.map` maps GRUB device names to Linux device names. In a mixed system containing IDE and SCSI hard disks, GRUB must try to determine the boot sequence by means of a special procedure, because GRUB does not have access to the BIOS information on the boot sequence. GRUB saves the result of this analysis in the file `/boot/grub/device.map`. For a system on which the boot sequence in the BIOS is set to IDE before SCSI, the file `device.map` could appear as follows:

```
(fd0) /dev/fd0
(hd0) /dev/hda
(hd1) /dev/sda
```

Because the order of IDE, SCSI, and other hard disks depends on various factors and Linux is not able to identify the mapping, the sequence in the file `device.map` can be set manually. If you encounter problems when booting, check if the sequence in this file corresponds to the sequence in the BIOS and use the GRUB shell to modify it temporarily if necessary (see *The GRUB Shell* on the following page). Once the Linux system has booted, the file `device.map` can be edited permanently with the YaST boot loader module or an editor of your choice.

After manually changing the file `device.map`, execute the following command to reinstall GRUB. This command causes the file `device.map` to be reloaded and the commands listed in `grub.conf` to be executed:

```
grub --batch < /etc/grub.conf
```

7.4.3 The File `/etc/grub.conf`

The third important GRUB configuration file apart from `menu.lst` and `device.map` is `/etc/grub.conf`. This file contains the parameters and options the command `grub` needs for installing the boot loader correctly:

```
root (hd0,4)
install /grub/stage1 d (hd0) /grub/stage2 0x8000 (hd0,4)/grub/menu.lst
quit
```

Meaning of the individual entries:

root (hd0,4) This command tells GRUB to apply the following commands to the first logical partition of the first hard disk (the location of the boot files).

install parameter The command `grub` should be run with the parameter `install`. `stage1` of the boot loader should be installed in the 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.

7.4.4 The GRUB Shell

GRUB actually exists in two versions: as a boot loader and as a normal Linux program in `/usr/sbin/grub`. This program is referred to as the *GRUB shell*. The functionality to install GRUB as boot loader on a hard disk or floppy disk is integrated in GRUB in the form of the commands `install` and `setup`. This is available in the GRUB shell when Linux is loaded.

However, the commands `setup` and `install` are also available during the boot procedure before Linux is started. This facilitates the repair of a defective system that can no longer be booted, because the faulty configuration file of the boot loader can be circumvented by manually entering parameters. The manual specification of parameters during the boot procedure is also useful for testing new settings to avoid impairing the native system. Simply enter the experimental configuration file with a syntax similar to that in `menu.lst`. Then test the functionality of this entry without changing the existing configuration file. For example, to test a new kernel, enter the command `kernel` and the path to the new kernel. If the boot procedure fails, you can continue using the intact `menu.lst` the next time you boot. Similarly, the command line interface can also be used to boot a system despite a faulty `menu.lst` file by entering the corrected parameters. In the running system, the correct parameters can be entered in `menu.lst` to make the system permanently bootable.

The mapping of GRUB devices to Linux device names is only relevant when running the GRUB shell as a Linux program (by entering `grub` as described in *The File device.map* on page 194). For this purpose, the program reads the file `device.map`. For more information, see *The File device.map* on page 194.

7.4.5 Setting a Boot Password

Even before the operating system is booted, GRUB enables access to file systems. Users without root permissions can access files in your Linux system to which they have no access once the system is booted. To block this kind of access or to prevent users from booting certain operating systems, a boot password can be set.

Proceed as follows as the user `root` to set a boot password:

1. At the root prompt, enter `grub`.
2. Encrypt the password in the GRUB shell:

```
grub> md5crypt
Password: ****
Encrypted: $1$1S2dv/$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$1S2dv/$JOYcdxIn7CJk9xShzzJVw/
```

Now GRUB commands can only be executed at the boot prompt after pressing (P) and entering the password. However, users can still boot all operating systems from the boot menu.

4. To prevent one or several operating systems from being booted from the boot menu, add the entry lock to every section in menu.lst that should not be bootable without entering a password. Example:

```
title linux
kernel (hd0,4)/vmlinuz root=/dev/hda7 vga=791
initrd (hd0,4)/initrd
lock
```

After rebooting the system and selecting the Linux entry from the boot menu, the following error message is displayed:

```
Error 32: Must be authenticated
```

Press (Enter) to enter the menu. Then press (P) to get a password prompt. After entering the password and pressing (Enter), the selected operating system (Linux in this case) should boot.

Note

Boot Password and Splash Screen

If you use a boot password for GRUB, the usual splash screen will not be displayed.

Note

7.5 Configuring the Boot Loader with YaST

The easiest way to configure the boot loader in your SUSE LINUX system is to use the YaST module. In the YaST Control Center, select 'System' → 'Boot Loader Configuration'. The current boot loader configuration of your system is displayed, enabling you to make any needed changes. See Figure 7.1.

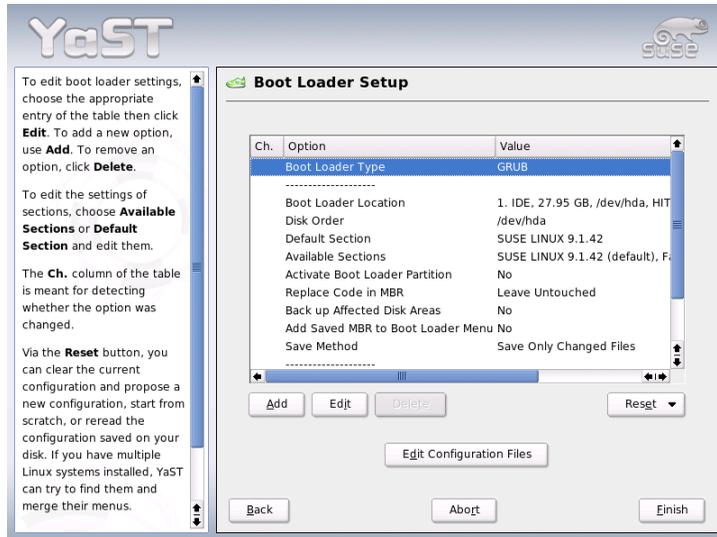


Figure 7.1: Configuring the Boot Loader with YaST

7.5.1 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' offers the following options:

Propose New Configuration Generates a new configuration suggestion. Older Linux versions or other operating systems found on other partitions are included in the boot menu, enabling you to boot Linux or its old boot loader. The latter takes you to a second boot menu.

Start from Scratch Enables you to create the entire configuration from scratch. No suggestions are generated.

Reread Configuration from Disk If you already performed some changes and are not satisfied with the result, 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 is 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 of Hard Disk The backup MBR saved on the hard disk is written back.

Use 'Edit Configuration Files' 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.

7.5.2 Boot Loader Configuration Options

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. Continue 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.

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 the kernel or operating system that should be booted by default. The selected system is booted after the time-out. In this menu, get a list of all boot menu entries with the button 'Edit'. Select an entry from the list and click 'Set as Default'. At this point, you may also modify any entry using the 'Edit'.

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 If GRUB was installed in the MBR or you are installing the system on a new hard disk and do not want to install GRUB in the MBR, restore the generic boot code to the MBR with this option.

Back 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.

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'. For details about the possible options, refer to the respective manual pages (`man grub` or `man lilo`) and the online documentation at <http://www.gnu.org/software/grub/manual/>.

7.6 Uninstalling the Linux Boot Loader

YaST can be used to uninstall the Linux boot loader and restore the MBR to the state prior to the installation of Linux. During the installation, YaST automatically creates a backup copy of the original MBR and restores it on request, overwriting GRUB.

To uninstall GRUB, start the YaST boot loader module ('System' → 'Boot Loader Configuration'). In the first dialog, select 'Reset' → 'Restore MBR of Hard Disk' and exit the dialog with 'Finish'. In the MBR, GRUB is overwritten with the data of the original MBR.

7.7 Creating Boot CDs

If problems occur booting your system using a boot manager or if the boot manager cannot be installed on the MBR of your hard disk or a floppy disk, it is also possible to create a bootable CD with all the necessary start-up files for Linux. This requires a CD writer installed in your system.

Creating a bootable CD-ROM with GRUB merely requires a special form of *stage2* called *stage2_eltorito* and, optionally, a customized *menu.lst*. The classic files *stage1* and *stage2* are not required.

Create a directory in which to create the ISO image, for example, with `cd /tmp` and `mkdir iso`. Also create a subdirectory for GRUB:

```
mkdir -p iso/boot/grub
```

Copy the file *stage2_eltorito* into the directory *grub* :

```
cp /usr/lib/grub/i386-pc/stage2_eltorito iso/boot/grub
```

Also copy the kernel (*/boot/vmlinuz*), the *initrd* (*/boot/initrd*), and the file */boot/message* to *iso/boot/* :

```
cp /boot/vmlinuz iso/boot/  
cp /boot/initrd iso/boot/  
cp /boot/message iso/boot/
```

To make them available to GRUB, copy the file `menu.lst` to `iso/boot` and adjust the path entries to make them point to a CD-ROM device. Do this by replacing the device name of the hard disks, listed in the format `(hd*)`, in the path names with the device name of the CD-ROM drive, which is `(cd)`:

```
gfxmenu (cd)/boot/message
timeout 8
default 0

title Linux
    kernel (cd)/boot/vmlinuz root=/dev/hda5 vga=794 resume=/dev/hda1
    splash=verbose showopts
    initrd (cd)/boot/initrd
```

Finally, create the ISO image with the following command:

```
mkisofs -R -b boot/grub/stage2_eltorito -no-emul-boot \
-boot-load-size 4 -boot-info-table -o grub.iso iso
```

Then write the resulting file `grub.iso` to a CD using the utility of your preference.

7.8 Troubleshooting

This section lists some of the problems frequently encountered when booting with GRUB and a short description of possible solutions. Some of the problems are covered in articles in the Support Database at <http://portal.suse.de/sdb/en/index.html>. If your specific problem is not included in this list, use the search dialog of the Support Database at <https://portal.suse.com/PM/page/search.pm> to search for keywords like “GRUB”, “boot”, and “boot loader”.

GRUB and XFS XFS leaves no room for `stage1` in the partition boot block.

Therefore, do not specify an XFS partition as the location of the boot loader. This problem can be solved by creating a separate boot partition that is not formatted with XFS.

GRUB and JFS Although technically possible, the combination of GRUB with JFS is problematic. In this case, create a separate boot partition (`/boot`) and format it with Ext2. Install GRUB in this partition.

GRUB Reports "GRUB Geom Error" GRUB checks the geometry of connected hard disks when the system is booted. Sometimes, the BIOS returns inconsistent information and GRUB reports a "GRUB Geom Error". If this is the case, use LILO or update the BIOS. Detailed information about the installation, configuration, and maintenance of LILO is available in the Support Database under the keyword LILO.

GRUB also returns this error message if Linux was installed on an additional hard disk that is not registered in the BIOS. *stage1* of the boot loader is found and loaded correctly, but *stage2* is not found. This problem can be remedied by registering the new hard disk in the BIOS.

System Containing IDE and SCSI Hard Disks Does Not Boot

During the installation, YaST may have determined the boot sequence of the hard disks incorrectly (and you may not have corrected it). For example, GRUB may regard `/dev/hda` as `hd0` and `/dev/sda` as `hd1`, though the boot sequence in the BIOS is the other way round (SCSI *before* IDE).

In this case, correct the hard disks during the boot process with the help of the GRUB command line. After the system has booted, edit the file `device.map` to apply the new mapping permanently. Then check the GRUB device names in the files `/boot/grub/menu.lst` and `/boot/grub/device.map` and reinstall the boot loader with the following command:

```
grub --batch < /etc/grub.conf
```

Bootimg Windows from the Second Hard Disk

Some operating systems, such as Windows, can only boot from the first hard disk. If such an operating system is installed on a hard disk other than the first hard disk, you can effect a logical change for the respective menu entry.

```
...
title windows
    map (hd0) (hd1)
    map (hd1) (hd0)
    chainloader(hd1,0)+1
...

```

In this example, Windows is started from the second hard disk. For this purpose, the logical order of the hard disks is changed with `map`. This change does not affect the logic within the GRUB menu file. Therefore, the second hard disk must be specified for `chainloader`.

7.9 For More Information

Extensive information about GRUB is available at <http://www.gnu.org/software/grub/>. If `texinfo` is 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.

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, 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 remains functional and can be booted if necessary.

8.1	Kernel Update	206
8.2	Kernel Sources	207
8.3	Kernel Configuration	207
8.4	Kernel Modules	208
8.5	Settings in the Kernel Configuration	211
8.6	Compiling the Kernel	211
8.7	Installing the Kernel	212
8.8	Cleaning Your Hard Disk after Compilation	213

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.

Often the behavior of the installed kernel can be modified by means of kernel parameters. For example, the parameter `desktop` sets shorter time slices for the scheduler, resulting in a subjective acceleration of the system. Information is available in the kernel documentation in the directory `/usr/src/linux/Documentation`, assuming the package `kernel-source` is installed.

Several `Makefiles` are provided with the kernel to automate the process. Select the hardware settings and other kernel features. Because 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.

8.1 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-$(uname -r) /boot/vmlinuz.old
cp /boot/initrd-$(uname -r) /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 RAM disk*. Therefore, use the command `mk_initrd` to write the new initial RAM disk. 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 *Booting and Boot Managers* on page 185). 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.

8.2 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. Because 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.

8.3 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 are queried. If the file `.config` does not exist, the default configuration included in the kernel sources is used.

8.3.1 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)`, and `(M)`. `(M)` means that this device should 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.

8.3.2 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.

8.3.3 Configuration in the X Window System

If you installed and configured the X Window System (package `xf86`) and the QT development package (`qt3-devel`), 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 are loaded from the file `.config`. Because 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.

8.4 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 runtime 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.

8.4.1 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.

8.4.2 Handling Modules

The following commands are available:

`insmod` `insmod` loads the requested module after searching for it in a sub-directory of `/lib/modules/<version>`. It is better, however, to use `modprobe` rather than `insmod`.

`rmmmod` 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 is 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, including probing all modules of a given type until one is successfully loaded. Unlike `insmod`, `modprobe` checks `/etc/modprobe.conf`, making it 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 and 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. Because this information is extracted from the module itself, only the information built in by the driver developers can be displayed. The information may include the author, a description, the license, module parameters, dependencies, and aliases.

8.4.3 `/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, for example, CD-ROM driver or network driver, may need system-specific options. 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`.

8.4.4 **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.

8.5 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`.

8.6 Compiling the Kernel

Compiling a “`bzImage`” is recommended. 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 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
```

These two commands can be entered as one command line:

```
make clean bzImage
```

After a successful compilation, the compressed kernel is located in `/usr/src/linux/arch/<arch>/boot`. The kernel image — the file that contains the kernel — is called `bzImage`.

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
```

If you have configured parts of your kernel to load as modules, launch the module compilation. Do this with `make modules`.

8.7 Installing the Kernel

After the kernel is compiled, it must be installed so it can be booted. The kernel must be installed in the directory `/boot`. Do this with the following command:

```
INSTALL_PATH=/boot make install
```

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 are 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

To enable GRUB to boot the old kernel (now `/boot/vmlinuz.old`), add the label `linux.old` as the boot image in the file `/boot/grub/menu.lst`. This procedure is described in detail in *Booting and Boot Managers* on page 185. GRUB does not need to be reinstalled.

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 compiled. If you get an error message like "System.map does not match current kernel", most likely you forgot to copy `System.map` to `/boot` following the compilation of the kernel.

8.8 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 are actually recompiled.

Special Features of SUSE LINUX

This chapter provides information about various software packages, the virtual consoles, and the keyboard layout. This is followed by a section about language and country-specific settings (I18N and L10N).

9.1	Hints for Special Software Packages	216
9.2	Virtual Consoles	224
9.3	Keyboard Mapping	224
9.4	Language and Country-Specific Settings	225

9.1 Hints for Special Software Packages

9.1.1 The Package bash and /etc/profile

The following is a list of all init files read by BASH when it is used as a login shell. BASH processes them in the order they appear in this list.

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 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.

9.1.2 The 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. In Example 9.1, root is entered. Package-specific tables, located in /etc/cron.d, have the same format. See man cron.

Example 9.1: 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. They are contained in the package `aaa_base`. `/etc/cron.daily` contains, for instance, the components `backup-rpmdb`, `clean-tmp`, or `clean-vi`.

9.1.3 Log Files: 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.

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 9.2: 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 also be configured here.

```

logrotate is controlled through CRON and 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

9.1.4 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 with 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.

9.1.5 The Command locate

`locate` for quickly finding files is not included in the standard scope of the installed software. If necessary, install the package (`find-locate`). The `updatedb` process is started automatically every night or about 15 minutes after booting the system.

9.1.6 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 9.1.

Table 9.1: `ulimit`: Setting Resources for the User

<code>-m</code>	maximum size of physical memory
<code>-v</code>	maximum size of virtual memory
<code>-s</code>	maximum size of the stack
<code>-c</code>	maximum size of the core files
<code>-a</code>	display of limits set

System-wide settings can be made in `/etc/profile`. There, enable creation of core files, needed by programmers for *debugging*. A normal user cannot increase the values specified in `/etc/profile` by the system administrator, but he can make special entries in his own `~/ .bashrc`.

Example 9.3: `ulimit`: Settings in `/.bashrc`

```
# Limits of physical memory:
ulimit -m 98304

# Limits of virtual memory:
ulimit -v 98304
```

Memory amounts must be specified in KB. For more detailed information, see `man bash`.

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

9.1.7 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 with access to a modern operating system, such as Linux, should not really need to worry much about memory. The concept of *available RAM* dates back to before the days of unified memory management. The slogan *free memory is bad memory* applies well to Linux. As a result, Linux has always made the effort to balance out caches without actually allowing free or unused memory.

Basically, the kernel does not have direct knowledge of any applications or user data. Instead, it manages applications and user data in a *page cache*. If memory runs short, parts of it are written to the swap partition or to files, from which they can initially be read with the help of the `mmap` command (see `man mmap`).

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`.

9.1.8 The File `/etc/resolv.conf`

Domain name resolution is handled through the file `/etc/resolv.conf`. Refer to *DNS — Domain Name System* on page 438.

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.

9.1.9 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`. To customize the program, copy `.gnu-emacs` to the home directory (with `cp /etc/skel/ .gnu-emacs ~/ .gnu-emacs`) and make the desired settings there.

`.gnu-emacs` defines the file `~/ .gnu-emacs-custom` as `custom-file`. If users make settings with the customize options, 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 divided into several packages:

- The base package `emacs`.
- `emacs-x11` (usually installed): the program *with* X11 support.
- `emacs-nox`: the program *without* X11 support.
- `emacs-info`: online documentation in info format.
- `emacs-el`: the uncompiled library files in Emacs Lisp. These are not required at run-time.

- Numerous add-on packages can be installed if needed: `emacs-auctex` (for LaTeX), `psgml` (for SGML and XML), `gnuserv` (for client and server operation), and others.

9.1.10 Brief Introduction to vi

Text editors are still used for many system administration tasks as well as for programming. In the world of Unix, `vi` stands out as an editor that offers comfortable editing functions and is more ergonomic than many editors with mouse support.

Operating Modes

Basically, `vi` makes use of three operating modes: *insert* mode, *command* mode, and *extended* mode. The keys have different functions depending on the mode. On start-up, `vi` is normally set to the *command* mode. The first thing to learn is how to switch between the modes:

Command Mode to Insert Mode There are many possibilities, including a for append, `i` for insert, or `o` for a new line under the current line.

Insert Mode to Command Mode Press `(Esc)` to exit the *insert* mode. `vi` cannot be terminated in *insert* mode, so it is important to get used to pressing `(Esc)`.

Command Mode to Extended Mode The *extended* mode of `vi` can be activated by entering a colon (`:`). The *extended* or *ex* mode is like an independent line-oriented editor that can be used for various simple and more complex tasks.

Extended Mode to Command Mode After executing a command in *extended* mode, the editor automatically returns to *command* mode. If you decide not to execute any command in *extended* mode, delete the colon with the backspace key. The editor returns to *command* mode.

It is not possible to switch directly from *insert* mode to *extended* mode without first switching to *command* mode.

`vi`, like other editors, has its own procedure for terminating the program. You cannot terminate `vi` while in *insert* mode. First, exit *insert* mode by pressing `(Esc)`. Subsequently, you have two options:

1. *Exit without saving*: To terminate the editor without saving the changes, enter `:q!` in *command* mode. The `!` causes `vi` to ignore any changes.

2. *Save and exit*: There are several possibilities to save your changes and terminate the editor. In *command* mode, use ZZ. To exit the program saving all changes from the *extended* mode, enter `:wq`. In *extended* mode, `w` stands for “write” and `q` for “quit”.

vi in Action

vi can be used as a normal editor. In *insert* mode, enter text and delete text with the backspace and delete keys. Use the arrow keys to move the cursor.

However, these control keys often cause problems, because there are many terminal types that use special key codes. This is where the *command* mode comes into play. Press `(Esc)` to switch from *insert* mode to *command* mode. In *command* mode, move the cursor with `(H)`, `(J)`, `(K)`, and `(L)`. The keys have the following functions:

- `(H)` move one character to the left
- `(J)` move one line down
- `(K)` move one line up
- `(L)` move one character to the right

The commands in *command* mode allow diverse variations. To execute a command several times, simply enter the number of repetitions before entering the actual command. For example, enter `(5)(L)` to move the cursor five characters to the right.

For More Information

vi supports a wide range of commands. It enables the use of macros, shortcuts, named buffers, and many other useful features. A detailed description of the various options would exceed the scope of this manual. SUSE LINUX comes with vim (vi improved), an improved version of vi. There are numerous information sources for this application:

- vimtutor is an interactive tutor for vim.
- In vim, enter the command `:help` to get help for many subjects.
- A book about vim is available online at <http://www.truth.sk/vim/vimbook-OPL.pdf>.

- The web pages of the vim project at <http://www.vim.org> feature all kinds of news, mailing lists, and other documentation.
- A number of vim sources are available on the Internet: <http://www.selflinux.org/selflinux/html/vim.html>, <http://www.linuxgazette.com/node/view/9039>, http://www.apmaths.uwo.ca/~xli/vim/vim_tutorial.html. See <http://linux-universe.com/HOWTO/Vim-HOWTO/vim-tutorial.html> for further links to tutorials.

Note

The `{protect}{dbapplication}{VIM}` License

vim is “charityware”, which means that the authors do not charge any money for the software but encourage you to support a non-profit project with a monetary contribution. This project solicits help for poor children in Uganda. More information is available online at <http://iccf-holland.org/index.html>, <http://www.vim.org/iccf/>, and <http://www.iccf.nl/>.

Note

9.2 Virtual Consoles

Linux is a multiuser and multitasking system. The advantages of these features can be appreciated even on a stand-alone PC system. In text mode, there are six virtual consoles available. Switch between them using `(Alt)-(F1)` to `(Alt)-(F6)`. The seventh console is reserved for X. More or fewer consoles can be assigned by modifying the file `/etc/inittab`.

To switch to a console from X without shutting it down, use `(Ctrl)-(Alt)-(F1)` to `(Ctrl)-(Alt)-(F6)`. `(Alt)-(F7)` then returns to X.

9.3 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.). Applications not shipped with SUSE LINUX should be adapted to these defaults.

Under X, the compose key (multikey) can be accessed using `Ctrl-Shift` (right). Also see the corresponding entry in `/usr/X11R6/lib/X11/Xmodmap`.

Further settings are possible over the “X Keyboard Extension” (XKB). This extension is also used by the desktop environments GNOME (`gswitchit`) and KDE (`kxkb`). Information about XKB is available in `/etc/X11/xkb/README` and the documents listed there.

Detailed information about the input of Chinese, Japanese, and Korean (CJK) is available at Mike Fabian’s page: <http://www.suse.de/~mfabian/suse-cjk/input.html>.

9.4 Language and Country-Specific Settings

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`).

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`.

RC_LC_ALL This variable (if set) overwrites the values of the variables mentioned above.

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.

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>]
```

9.4.1 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 (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 during installation. If you selected another language, that language is enabled but still with UTF-8 as the character encoding.

LANG=en_US.ISO-8859-1 This sets the language to English, country to United States, and the character set to ISO-8859-1. This character set does not support the Euro sign, but it can be useful sometimes for programs that have not been updated to support UTF-8. The string defining the charset (ISO-8859-1 in this case) is then evaluated by programs like Emacs.

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, include `LC_MESSAGES=es_ES` so messages are displayed in Spanish instead.

9.4.2 Settings for Language Support

Files in the category *Messages* are, as a rule, only stored in the corresponding language directory (like `en`) to have a fallback. If you set `LANG` to `en_US` and the *message* file in `/usr/share/locale/en_US/LC_MESSAGES` does not exist, it falls back to `/usr/share/locale/en/LC_MESSAGES`.

A fallback chain can also be defined, for example, for Breton to French or for Galician to Spanish to Portuguese:

```
LANGUAGE="br_FR:fr_FR"
```

```
LANGUAGE="gl_ES:es_ES:pt_PT"
```

If desired, use the Norwegian variants *nynorsk* and *bokmål* instead (with additional fallback to `no`):

```
LANG="nn_NO"
```

```
LANGUAGE="nn_NO:nb_NO:no"
```

or

```
LANG="nb_NO"
```

```
LANGUAGE="nb_NO:nn_NO:no"
```

Note that in Norwegian, `LC_TIME` is also treated differently.

Possible Problems

The thousands comma is not recognized. LANG is probably set to en, but the description glibc uses is located in /usr/share/lib/en_US/LC_NUMERIC. LC_NUMERIC, for example, must be set to en_US.

For More Information

- *The GNU C Library Reference Manual*, Chapter “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`.

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 current implementation is compatible with the *System Initialization* section of the LSB specification. Refer to *Standards and Specifications* on page 656 for information about LSB.

10.1	Booting with the Initial RAM Disk	230
10.2	The init Program	235
10.3	Runlevels	235
10.4	Changing Runlevels	237
10.5	Init Scripts	238
10.6	The YaST Runlevel Editor	241
10.7	SuSEconfig and /etc/sysconfig	243
10.8	The YaST sysconfig Editor	245

The kernel takes control of the system's hardware as soon as the simple message "Uncompressing Linux..." is printed on screen. 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 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.

10.1 Booting with the Initial RAM Disk

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.). If the root file system is already encrypted, 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 RAM disk*: running user space programs even before the root file system is mounted.

10.1.1 Concept of the Initial RAM Disk

The *initial RAM disk* (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 RAM disk. Special drivers are not required.

10.1.2 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 attempts to unmount `initrd`. If this does not work, the system is fully functional, but the memory taken up by `initrd` cannot be unlocked, so is no longer available.

linuxrc

The only requirements for `linuxrc` in the `initrd` are that it must have the special name `linuxrc`, it must be located in the root directory of the `initrd`, and it must 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`.

10.1.3 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
```

Because 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.

10.1.4 Using `initrd` in SUSE

Installing the System

The `initrd` has 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.

Booting the Installed System

In the past, YaST offered more than forty kernels for installing in the system. The main difference among 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 now used 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

Because 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

10.1.5 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 (so it is 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.

10.1.6 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`.

10.2 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 *Runlevels* on this page). 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.

10.3 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 10.1 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 work with the system without X. To enable a graphical environment, like GNOME, KDE, or any other window manager, use `init 5` instead. `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 10.1: 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. If the default runlevel is 3, the X Window System must be configured properly, as described in *The X Window System* on page 247, before the runlevel can be switched to 5. If this is done, 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

10.4 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 those for which there is no start script in the new runlevel. In this example, these are all the scripts that reside in `/etc/init.d/rc3.d` (old runlevel was 3) and start with a `K`. The number following `K` specifies the order to start, because there are some dependencies to consider.
4. The last things to start are the start scripts of the new runlevel. These are, in this example, in `/etc/init.d/rc5.d` and begin with an `S`. The same procedure regarding the order in which they are started is applied here.

When changing into the same runlevel as the current runlevel, `init` only checks `/etc/inittab` for changes and starts the appropriate steps, for example, for starting a `getty` on another interface.

10.5 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)`).

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 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 10.2.

Table 10.2: Possible `init` Script Options

Option	Description
<code>start</code>	Start service.
<code>stop</code>	Stop service.
<code>restart</code>	If the service is running, stop it then restart it. If it is not running, start it.
<code>reload</code>	Reload the configuration without stopping and restarting the service.
<code>force-reload</code>	Reload the configuration if the service supports this. Otherwise, do the same as if <code>restart</code> had been given.
<code>status</code>	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 `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. `blogd` writes any screen output to the log file `/var/log/boot.msg`, but only if and when `/var` is mounted read-write. Otherwise, `blogd` buffers all screen data until `/var` becomes available. Get further information about `blogd` 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 as the keyboard layout and initialization of the virtual consoles.

halt This script is only executed while changing into runlevel 0 or 6. Here, it is executed either as `halt` or as `reboot`. Whether the system shuts down or reboots depends on how `halt` is called.

rc This script calls the appropriate stop scripts of the current runlevel and the start scripts of the newly selected runlevel.

10.5.1 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 *Runlevels* on page 235.

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 10.1.

Example 10.1: 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 *The YaST Runlevel Editor* on this page.

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.

10.6 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).

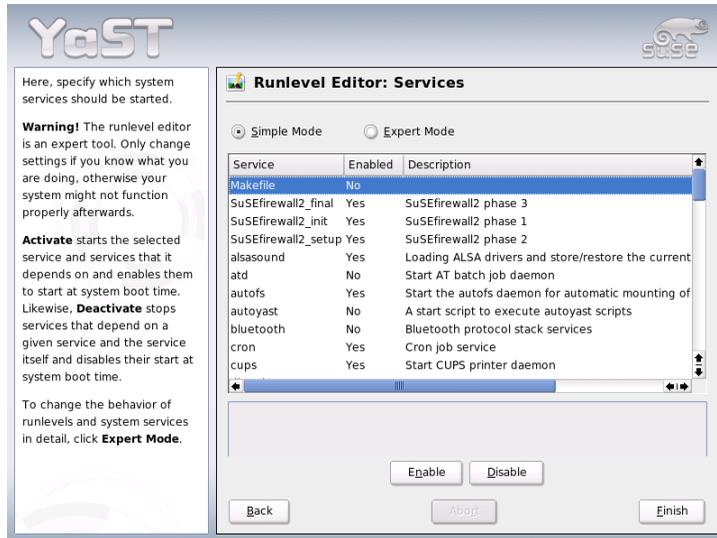


Figure 10.1: YaST Runlevel Editor

This YaST dialog allows the selection of one of the runlevels (as listed in Table 10.1 on page 236) 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 that you know about their consequences.

Caution

10.7 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, because 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, because 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 *The YaST sysconfig Editor* on the facing 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 automated system configuration by `SuSEconfig`, set the variable `ENABLE_SUSECONFIG` in `/etc/sysconfig/suseconfig` to `no`. Do not disable `SuSEconfig` if you want to use the SUSE installation support. It is also possible to disable the autoconfiguration partially.

Note

10.8 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

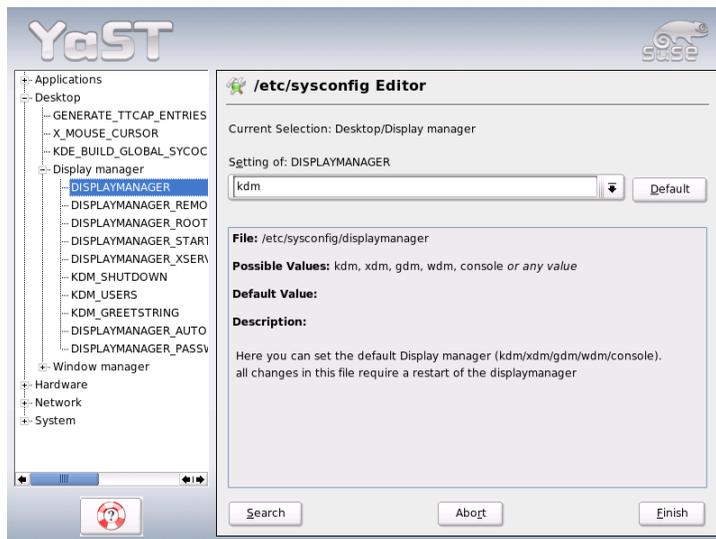


Figure 10.2: System Configuration Using the sysconfig Editor

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. X is network-based, enabling applications started on one host to be displayed on another host connected over any kind of network (LAN or Internet).

This chapter 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 (*Graphics Card and Monitor (SaX2)* on page 64).

11.1	Optimizing the X Configuration	248
11.2	Installing and Configuring Fonts	253
11.3	OpenGL — 3D Configuration	258

11.1 Optimizing the X Configuration

X.Org is an Open Source implementation of the X Window System. It is further developed by the X.Org Foundation, which is also responsible for the development of new technologies and standards of the X Window System.

To use the available hardware, including mouse, graphics card, monitor, and 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/Xorg` 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 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.

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`. A rough outline of the most important sections follows.

`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 available section types are listed in Table 11.1 on the facing page.

Table 11.1: 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 are <code>Driver</code> and the options defining the <code>Protocol</code> and <code>Device</code> .
Monitor	Describes the monitor used. The individual elements of this section are the name, which is referred to later in the <code>Screen</code> definition, the bandwidth, and the synchronization frequency limits (<code>HorizSync</code> and <code>VertRefresh</code>). Settings are given in MHz, kHz, and Hz. Normally, the server refuses any modeline that does not correspond with the specification of the monitor. This prevents too high frequencies from being sent to the monitor by accident.
Modes	The modeline parameters 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. Find details of the meaning of individual number values in the HOWTO file <code>/usr/share/doc/howto/en/XFree86-Video-Timings-HOWTO.gz</code> .
Device	This section defines a specific graphics card. It is referenced by its descriptive name.
Screen	This section puts together a <code>Monitor</code> and a <code>Device</code> to form all the necessary settings for X.Org. In the <code>Display</code> subsection, specify the size of the virtual screen (<code>Virtual</code>), the <code>Viewport</code> , and the <code>Modes</code> used with this screen.
ServerLayout	This section defines the layout of a single or multi-head configuration. This section binds the input devices <code>InputDevice</code> and the display devices <code>Screen</code> .

Monitor, Device, and Screen are explained in more detail below. Further information about the other sections can be found in the manual pages of X.Org 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.

11.1.1 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 Example 11.1.

Example 11.1: 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, the X server searches for a suitable `Modeline` in the `Modes` section. The `Modeline` depends on the capability of both the monitor and the graphics card. The `Monitor` settings determine the resulting `Modeline`.

The first resolution found is the `Default` mode. With `(Ctrl)-(Alt)+` (on the number pad), switch to the next resolution in the list to the right. With `(Ctrl)-(Alt)-` (on the number pad), switch to the 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.

11.1.2 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 example. 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`. `._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 X.Org`).

11.1.3 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, because this could cause severe damage to your monitor.

Caution

Those who try 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.

Manual specification of modelines is rarely required today. If you are using a modern multisync monitor, the allowed frequencies and optimal resolutions can, as a rule, be read directly from the monitor by the X server via DDC, as described in the SaX2 configuration section. If this is not possible for some reason, use one of the VESA modes included in the X server. This will function with practically all graphics card and monitor combinations.

11.2 Installing and Configuring Fonts

The installation of additional fonts in SUSE LINUX is very easy. Simply copy the fonts to any directory located in the X11 font path (see *X11 Core Fonts* on page 257). 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 *Xft* on the next 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 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 *CID-Keyed Fonts* on page 258.

X.Org contains two completely different font systems: the old *X11 core font system* and the newly designed *Xft and fontconfig* system. The following sections briefly describe these two systems.

11.2.1 Xft

From the outset, the programmers of Xft made sure that scalable fonts including antialiasing are supported well. If Xft is used, the fonts are rendered by the application using the fonts, not by the X server as in the X11 core font system. In this way, the respective application has access to the actual font files and full control of how the glyphs are rendered. This constitutes the basis for the correct display of text in a number of languages. Direct access to the font files is very useful for embedding fonts for printing to make sure that the printout looks the same as the screen output.

In SUSE LINUX, the two desktop environments KDE and GNOME, Mozilla, and many other applications already use Xft by default. Xft is already used by more applications than the old X11 core font system.

Xft uses the fontconfig library for finding fonts and influencing how they are rendered. The properties of fontconfig are controlled by the global configuration file `/etc/fonts/fonts.conf` and the user-specific configuration file `~/.fonts.conf`. Each of these fontconfig configuration files must begin with

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
```

and end with

```
</fontconfig>
```

To add directories to search for fonts, append lines such as the following:

```
<dir>/usr/local/share/fonts/</dir>
```

However, this is usually not necessary. By default, the user-specific directory `~/ . fonts` is already entered in `/etc/fonts/fonts.conf`. Accordingly, all you need to do to install additional fonts is to copy them to `~/ . fonts`.

You can also insert rules that influence the appearance of the fonts. For example, enter

```
<match target="font">
  <edit name="antialias" mode="assign">
    <bool>false</bool>
  </edit>
</match>
```

to disable antialiasing for all fonts or

```
<match target="font">
  <test name="family">
    <string>Luxi Mono</string>
    <string>Luxi Sans</string>
  </test>
  <edit name="antialias" mode="assign">
    <bool>false</bool>
  </edit>
</match>
```

to disable antialiasing for specific fonts.

By default, most applications use the font names `sans-serif` (or the equivalent `sans`), `serif`, or `monospace`. These are not real fonts but only aliases that are resolved to a suitable font, depending on the language setting.

Users can easily add rules to `~/ . fonts.conf` to resolve these aliases to their favorite fonts:

```
<alias>
  <family>sans-serif</family>
  <prefer>
    <family>FreeSans</family>
  </prefer>
</alias>
<alias>
  <family>serif</family>
  <prefer>
    <family>FreeSerif</family>
  </prefer>
</alias>
<alias>
```

```

<family>monospace</family>
<prefer>
  <family>FreeMono</family>
</prefer>
</alias>

```

Because nearly all applications use these aliases by default, this affects almost the entire system. Thus, you can easily use your favorite fonts almost everywhere without having to modify the font settings in the individual applications.

Use the command `fc-list` to find out which fonts are installed and available for use. For instance, the command `fc-list ""` returns a list of all fonts. To find out which of the available scalable fonts (`: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 11.2: Parameters of `fc-list`

Parameter	Meaning and Possible Values
<code>family</code>	Name of the font family, for example, <code>FreeSans</code> .
<code>foundry</code>	The manufacturer of the font, for example, <code>urw</code> .
<code>style</code>	The font style, such as <code>Medium</code> , <code>Regular</code> , <code>Bold</code> , <code>Italic</code> , <code>Heavy</code> .
<code>lang</code>	The language that the font supports, for example, <code>de</code> for German, <code>ja</code> for Japanese, <code>zh-TW</code> for traditional Chinese, or <code>zh-CN</code> for simplified Chinese.

<code>weight</code>	The font weight, such as 80 for regular, 200 for bold.
<code>slant</code>	The slant, usually 0 for none and 100 for italic.
<code>file</code>	The name of the file containing the font.
<code>outline</code>	true for outline fonts, false for other fonts.
<code>scalable</code>	true for scalable fonts, false for other fonts.
<code>bitmap</code>	true for bitmap fonts, false for other fonts.
<code>pixelsize</code>	Font size in pixels. In connection with <code>fc-list</code> , this option only makes sense for bitmap fonts.

11.2.2 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 languages 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 and fontconfig system should be used if at all possible.

For its operation, the X server needs to know what fonts it has available and where in the system it can find them. This is handled by a `FontPath` variable, which contains the path to all valid system font directories. In each of these directories, a file named `fonts.dir` lists the available fonts in this directory. The `FontPath` is generated by the X server at start-up. It searches for a valid `fonts.dir` file in each of the `FontPath` entries in the configuration file `/etc/X11/XF86Config`. These entries are found in the `Files` section. Display the actual `FontPath` with `xset q`. This path may also be changed at runtime with `xset`. To add an additional path, use `xset +fp <path>`. To remove an unwanted path, use `xset -fp <path>`.

If the X server is already active, newly installed fonts in mounted directories can be made available with the command `xset fp rehash`. This command is executed by `SuSEconfig --module fonts`. Because the command `xset` needs access to the running X server, this only works if `SuSEconfig --module fonts` is started from a shell that has access to the running X server. The easiest way to achieve this is to assume `root` permissions by entering `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-*`).

11.2.3 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 and `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

11.3 OpenGL — 3D Configuration

11.3.1 Hardware Support

SUSE LINUX includes several OpenGL drivers for 3D hardware support. Table 11.3 on the next page provides an overview.

Table 11.3: 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.

11.3.2 OpenGL Drivers

The OpenGL drivers nVidia and DRI can be configured easily with SaX2. For nVidia adapters, the nVidia driver must be installed first. 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 is 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.

11.3.3 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 X.Org 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 only see "done" messages on the screen.

11.3.4 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 with `direct rendering: Yes`.

11.3.5 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 X.Org log files.

Often, you will find the line `DRI is disabled` in the X.Org file `/var/log/Xorg.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, because 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.

11.3.6 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 (X Window System) does not include 3D hardware acceleration configuration. If you experience problems with 3D hardware acceleration, it is recommended to disable 3D support completely.

11.3.7 Additional Online Documentation

For information about DRI, refer to `/usr/X11R6/lib/X11/doc/README.DRI (xorg-x11-doc)`.

Printer Operation

This chapter provides general information about operating printers and helps find suitable solutions for operating printers in networks. Special emphasis is placed on CUPS operation. A detailed troubleshooting section outlines the most common pitfalls in printer operation and describes ways to avoid them.

12.1	Preparation and Other Considerations	264
12.2	Methods and Protocols for Connecting Printers	265
12.3	Installing the Software	266
12.4	Configuring the Printer	267
12.5	Special Features in SUSE LINUX	271
12.6	Troubleshooting	276

12.1 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 for reasons of compatibility.

Printers can be distinguished in terms of the interfaces, such as USB or 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 categorized on the basis of the following three classes of printer languages:

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

Standard Printer (languages like PCL and ESC/P)

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

Proprietary Printers (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. See *Printers without Standard Printer Language Support* on page 276 for more information.

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

- <http://cdb.suse.de/> or <http://hardwaredb.suse.de/> — the SUSE LINUX printer database
- <http://www.linuxprinting.org/> — the LinuxPrinting.org printer database
- <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.

12.2 Methods and Protocols for Connecting Printers

There are various possibilities for connecting a printer to the system. The configuration of the CUPS print system does not distinguish between a local printer and a printer connected to the system over the network. In Linux, local printers must be connected as described in the manual of the printer manufacturer. CUPS supports serial, USB, parallel, and SCSI connections. For more information about the printer connection, read the article *CUPS in a Nutshell* in the Support Database at <http://portal.suse.com>. Find the article by entering *cups* in the search dialog.

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

12.3 Installing the Software

PPD (PostScript printer description) is the computer language that describes the properties, like resolution, and options, for example, duplex unit, of PostScript printers. These descriptions are necessary to 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 to enable even printers without PostScript support to be used.

To configure a PostScript printer, the best approach is to get a suitable PPD file. Many such PPDs are available in the package `manufacturer-PPDs` which is automatically installed within the scope of the standard installation (see *PPD Files in Various Packages* on page 274 and *No Suitable PPD File Available for a PostScript Printer* on page 277).

New PPD files can be stored in the directory `/usr/share/cups/model/` or added to the print system with YaST (see *Manual Configuration* on page 61). Subsequently, the PPD file can be selected during the installation.

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

12.4 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. Because SUSE LINUX puts great emphasis on security, third-party tools often have difficulties with the security restrictions and end up causing more complications than benefits.

12.4.1 Local Printers

If an unconfigured local printer is detected when you log in, YaST starts for configuring it. The procedure is described in *Configuration with YaST* on page 61. To configure the printer manually with command-line tools (see *Configuring with Command-Line Tools* on page 269), you need a device URI (uniform resource identifier) consisting of a back-end, such as “usb”, and parameters, like “/dev/usb/lp1”. For example, the full URI could be `parallel:/dev/lp0` (printer connected to the first parallel port) or `usb:/dev/usb/lp1` (first detected printer connected to the USB port).

12.4.2 Network Printers

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

CUPS supports the `socket`, `LPD`, `IPP`, and `smb` protocols. Here is some detailed information about these protocols:

socket *Socket* refers to a connection in which the data is sent to an Internet socket without first performing a data handshake. Some of the socket port numbers that are commonly used are 9100 or 35. An example device URI is `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 ID of the printer queue, is sent before the actual print data is sent. Therefore, a printer queue must be specified when configuring the LPD protocol for the data transmission. The implementations of diverse printer manufacturers are flexible enough to accept any name as printer queue. If necessary, the printer manual should indicate what 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. An example device URI is `lpd://<host-printer>/LPT1`.

IPP (Internet printing protocol) IPP is a relatively new (1999) protocol based on the HTTP protocol. With IPP, more job-related data is transmitted than in the other protocols. CUPS uses IPP for internal data transmission. This is the preferred protocol for a forwarding queue between two CUPS servers. The name of the print queue is necessary to configure IPP correctly. The port number for IPP is 631. Example device URIs are `ipp://<host-printer>/ps` or `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 uses the port numbers 137, 138, and 139. Example for a device URI:

```
smb://<user>:<password>@<workgroup>/<server>/<printer>
or: smb://<user>:<password>@<host>/<printer> or:
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. For example:

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

12.4.3 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 *Administrator for CUPS Web Front-End* on page 272).

Configuring CUPS in the Network Using YaST

For guidelines for installation of CUPS in the network, read the article *CUPS in a Nutshell* in the Support Database at <http://portal.suse.com>. For configuring the queues for the printers belonging to the CUPS server on the print server, you have the following options:

Network Printers or Print Server Box via TCP socket with local filtering (default) or without local filtering

via LPD protocol with local filtering (default) or without local filtering

via IPP protocol with local filtering (default) or without local filtering

Queue on LPD Server (always via LPD protocol)

without local filtering (default) or with local filtering

Queue on IPP Server (always via IPP protocol)

without local filtering (default) or with local filtering

Queue on SMB Server (always via SMB protocol)

with local filtering (default) or without local filtering

Queue on IPX Server (always via Novell IPX)

with local filtering (default) or without local filtering

Queue via Other URI with local filtering (default) or without local filtering

When permitting access to the queues for the client computers, the default settings are usually sufficient. If you are not sure, refer to the portal article mentioned above.

To activate the transmission of browsing information to the client computer, start YaST, select 'Printer' and click 'Change...' then 'Advanced' and select 'CUPS server settings'. Next, go to 'Browse Addresses', click 'Add', and enter the broadcast IP address of the network or @LOCAL. Conclude the configuration by clicking 'OK', 'Next', 'Accept', then 'Finish'

Configuring with Command-Line Tools

Alternatively, CUPS can be configured with command-line tools. If the preparatory work has been done (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-levell.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 used) or specified later, for example, 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
```

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
```

12.5 Special Features in SUSE LINUX

A number of features of CUPS have been adapted for use with SUSE LINUX. Some of the most important changes are covered here.

12.5.1 CUPS Server and Firewall

There are several ways to configure CUPS as the client of a network server.

- For every queue on the network server, you can configure a local queue through which to forward all jobs to the corresponding network server. Usually, this approach is not recommended, because all client machines must be reconfigured whenever the configuration of the network server changes
- Print jobs can also be forwarded directly to one network server. For this type of configuration, do not run a CUPS daemon. `lpr` (or corresponding library calls of other programs) can send jobs directly to the network server. However, this configuration does not work if you want to print on a local printer.
- The CUPS daemon can listen to IPP broadcast packets that other network servers send to announce available queues. This is the best CUPS configuration for printing over remote CUPS servers. However, there is a risk that an attacker sends the daemon IPP broadcasts with queues and the local daemon accesses a counterfeit queue. If it then displays the queue with the same name as another queue on the local server and the IPP packet is received earlier, the owner of the job may believe the job is sent to a local server, while in reality it is sent to the attacker's server. To use this method, port 631/UDP must be open for incoming packets.

YaST knows two methods for finding CUPS servers:

1. Scanning all network hosts to see if they offer this service.
2. Listening to IPP broadcasts (same method as described above). This method is also used during the installation to find CUPS servers for the proposal.

The second method requires port 631/UDP to be open for incoming packets.

The default setting of the firewall shown in the proposal dialog is not to permit IPP broadcasts on any interface. Accordingly, the second method for detecting remote queues and the third method for accessing remote queues cannot work. Therefore, the firewall configuration must be modified by marking one of the interfaces as `internal`, which opens the port by default, or by explicitly opening the port of an `external` interface. For security reasons, none of the ports is open by default. Opening a port to configure access to remote queues using the second method can be a security risk because an attacker could broadcast a server that might be accepted by users.

The proposed firewall configuration must be modified to enable CUPS to detect remote queues during the installation and access various remote servers from the local system during normal operation. Alternatively, the user can detect CUPS servers by actively scanning the local network hosts or configure all queues manually. However, due to the reasons mentioned above, this method is not recommended.

12.5.2 Administrator for CUPS Web Front-End

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. Do this as `root` with the following command:

```
lppasswd -g sys -a root
```

If this is not done, administration with the web interface or with the administration tool is not possible, because the authentication fails if no CUPS administrator has been configured. Instead of `root`, any other user can also be appointed as CUPS administrator (see *Changes in the CUPS Print Service (cupsd)* on the current page).

12.5.3 Changes in the CUPS Print Service (cupsd)

For information about these changes, see the Support Database article “Printer Configuration from SUSE LINUX 9.0” at <http://portal.suse.com>. Find the article by entering *printer* in the search dialog.

cupsd Runs as the User lp

On start-up, cupsd changes from the user root to the user lp. This provides a much higher level of security, because the CUPS print service does not run with unrestricted permissions, but only with the permissions needed for the print service.

However, the authentication (more precisely: the password check) cannot be performed via /etc/shadow, as lp has no access to /etc/shadow. Instead, 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>
```

When cupsd runs as lp, /etc/printcap cannot be generated, because lp is not permitted to create files in /etc/. Therefore, cupsd generates /etc/cups/printcap. 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.

When cupsd runs as lp, port 631 cannot be opened. Therefore, cupsd cannot 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 Activated by Default

In a standard installation, `cupsd` is activated automatically, enabling comfortable access to the queues of CUPS network servers without any additional manual actions. The two first items are vital preconditions for this feature, as otherwise the security would not be sufficient for an automatic activation of `cupsd`.

12.5.4 PPD Files in Various Packages

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 system. To determine the suitable PPD files for the 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, receive the PPD files matching the vendor and model.

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

CUPS PPD Files in the cups Package

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

- `/usr/share/cups/model/Postscript-level1.ppd.gz`
- `/usr/share/cups/model/Postscript-level2.ppd.gz`

PPD Files in the cups-drivers Package

Normally, the Foomatic printer filter "foomatic-rip" is used together with Ghostscript for non-PostScript printers. Suitable Foomatic PPD files have the entries "`*NickName: ... Foomatic/(Ghostscript driver)`" and "`*cupsFilter: ... foomatic-rip`". These PPD files are located in the `cups-drivers` package.

YaST prefers a Foomatic PPD file if 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).

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

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

PPD Files from Printer Manufacturers in the manufacturer-PPDs Package

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

- The vendor and model determined during the hardware detection match the vendor and model in a PPD file from the `manufacturer-PPDs` package.
- The PPD file from the `manufacturer-PPDs` package is the only suitable PPD file for the printer model or there is a Foomatic PPD file with a "`*NickName: ... Foomatic/Postscript (recommended)`" entry that also matches the printer model.

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

- The PPD file from the `manufacturer-PPDs` package does not match the vendor and model. This may happen if the `manufacturer-PPDs` package contains only one PPD file for similar models, for example, if there is no separate PPD file for the individual models of a model series, but the model name is specified in a form like "Funprinter 1000 series" in the PPD file.
- The Foomatic PostScript PPD file is not "recommended". This may be because the printer model does not operate efficiently enough in PostScript mode, for example, the printer may be unreliable in this mode because it has too little memory or the printer is too slow because its processor is too weak). Furthermore, the printer may not support PostScript by default, for example, because PostScript support is only available as an optional module).

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

12.6 Troubleshooting

The following sections cover some of the most frequently encountered printer hardware and software problems and ways to solve or circumvent these problems.

12.6.1 Printers without Standard Printer Language Support

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

12.6.2 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.

12.6.3 Parallel Ports

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

- I/O address: 378 (hexadecimal)
- Interrupt: irrelevant
- Mode: Normal, SPP, or Output Only
- DMA: disabled

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

If interrupt 7 is free, it can be activated with the entry shown in Example 12.1. Before activating the interrupt mode, check the file `/proc/interrupts` to see which interrupts are already in use. Only the interrupts currently being used are displayed. This may change depending on which hardware components are active. The interrupt for the parallel port must not be used by any other device. If you are not sure, use the polling mode with `irq=none`.

Example 12.1: /etc/modprobe.conf: Interrupt Mode for the First Parallel Port

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

12.6.4 Network Printer Connections

Identifying Network Problems Connect the printer directly to the computer.

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

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

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

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

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

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

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

If the `lpd` does not respond, it may not be active or there may be basic network problems. If `lpd` responds, the response should show why printing is not possible on the queue on `host`. If you receive a response like that in Example 12.2, the problem is caused by the remote `lpd`.

Example 12.2: 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
```

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

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

If a broadcasting CUPS network server exists, the following output should be returned after forty seconds:

Example 12.3: 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 `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. Because this is caused by the spooler in the print server box, there is nothing you can do about it. As a workaround, circumvent the spooler in the print server box by addressing the printer connected to the print server box directly via TCP socket (see *Network Printers* on page 267).

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

Enter a command like

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

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

12.6.5 Defective Printouts without Error Message

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

12.6.6 Disabled Queues

If the data transfer to the recipient fails entirely after several attempts, the CUPS back-end, such as `usb` or `socket`, reports an error to the print system (more precisely: to `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` disables printing for the respective queue (`disable`). After eliminating the cause of the problem, the system administrator must reenables printing with the command `/usr/bin/enable`.

12.6.7 CUPS Browsing: Deleting Print Jobs

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

12.6.8 Defective Print Jobs and Data Transfer Errors

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

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

1. To stop printing, remove all paper from ink-jet printers or open the paper trays of laser printers. High-quality printers have a button for canceling the current printout.
2. The print job may still be in the queue, because jobs are only removed after they are sent completely to the printer. Use `lpstat -o` (or `lpstat -h <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 turn on the printer.

12.6.9 Debugging the CUPS Print System

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

1. Set `LogLevel debug` in `/etc/cups/cupsd.conf`.
2. Stop `cupsd`.
3. Remove `/var/log/cups/error_log*` to avoid having to search through very large log files.
4. Start `cupsd`.

5. Repeat the action that led to the problem.
6. Check the messages in `/var/log/cups/error_log*` to identify the cause of the problem.

Mobile Computing with Linux

This chapter provides an overview of the various aspects of using Linux for mobile computing. The various fields of use are briefly introduced and the essential features of the employed hardware are described. Software solutions for special requirements and options for maximum performance are covered along with possibilities to minimize power consumption. An overview of the most important sources of information about the subject concludes the chapter.

13.1	Laptops	286
13.2	Mobile Hardware	292
13.3	Cellular Phones and PDAs	293
13.4	For More Information	293

Most people associate mobile computing with laptops, PDAs, and cellular phones and the data exchange between them. This chapter extends the focus to mobile hardware components, such as external hard disks, flash drives, or digital cameras, which can be connected to laptops or desktop systems.

13.1 Laptops

13.1.1 Special Properties of Laptop Hardware

The hardware of laptops differs from that of a normal desktop system. This is because criteria like exchangeability, occupied space, and power consumption are relevant properties. The manufacturers of mobile hardware have developed the PCMCIA standard (*Personal Computer Memory Card International Association*). This standard covers memory cards, network interface cards, ISDN and modem cards, and external hard disks. How the support for such hardware is implemented in Linux, what needs to be taken into account during configuration, what software is available for the control of PCMCIA, and how to troubleshoot any possible problems is described in *PCMCIA* on page 295.

13.1.2 Power Conservation

The inclusion of energy-optimized system components when manufacturing laptops contributes to their suitability for use without access to the electrical power grid. Their contribution towards conservation of power is at least as important as that of the operating system. SUSE LINUX supports various methods that influence the power consumption of a laptop and have varying effects on the operating time under battery power. The following list is ordered in descending order of contribution towards power conservation:

- Throttling the CPU speed
- Switching off the display illumination during pauses
- Manual adjustment of the display illumination
- Disconnection of unused, hotplug enabled accessories (USB CD-ROM, external mouse, unused PCMCIA cards, etc.)
- Spinning down the hard disk when idling

Detailed background information about power management in SUSE LINUX and about operating the YaST power management module is provided in *Power Management* on page 313.

13.1.3 Integration in Changing Operating Environments

Your system needs to adapt to changing operating environments when used for mobile computing. A lot of services depend on the environment and the underlying clients must be reconfigured. SUSE LINUX takes over this job for you.

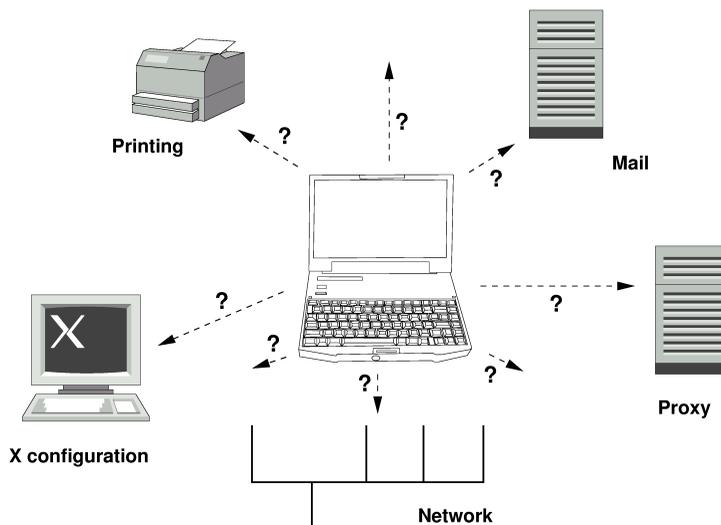


Figure 13.1: Integrating a Laptop in a Network

The services affected in the case of a laptop commuting back and forth between a small home network and an office network are:

Network Configuration This includes IP address allocation, name resolution, Internet connectivity, and connectivity to other networks.

Printing A current database of available printers and an available print server must be present, depending on the network.

E-Mail and Proxies As with printing, the list of the corresponding servers must be current.

Configuring X If your laptop is temporarily connected to a beamer or an external monitor, the different display configurations need to be kept available.

SUSE LINUX offers two ways of integrating a laptop into existing operating environments that can be combined.

SCPM SCPM (*system configuration profile management*) allows storage of arbitrary configuration states of a system into a kind of “snapshot” called a *profile*. Profiles can be created for different situations. They are useful when a system is operated in changing environments (home network, office network). It is always possible to switch between profiles. Information about SCPM can be found in *System Configuration Profile Management* on page 305. The kicker applet Profile Chooser in KDE allows switching between profiles. The application requires the root password before switching.

SLP The *service location protocol* (SLP) simplifies the connection of a laptop to an existing network. Without SLP, the administrator of a laptop usually requires detailed knowledge of the services available in a network. SLP broadcasts the availability of a certain type of service to all clients in a local network. Applications that support SLP can process the information dispatched by SLP and be configured automatically. SLP can even be used for the installation of a system and spare the effort of searching for a suitable installation source. Detailed information about SLP can be found in *SLP Services in the Network* on page 435.

The emphasis of SCPM lies on enabling and maintaining reproducible system conditions. SLP makes configuration of a networked computer a lot easier by automating much of it.

13.1.4 Software Options

There are various special task areas in mobile use that are covered by dedicated software: system monitoring (especially the battery charge), data synchronization, and wireless communication with peripherals and the Internet. The following sections cover the most important applications that SUSE LINUX provides for each task.

System Monitoring

Two KDE system monitoring tools are provided by SUSE LINUX. The pure status display of the rechargeable battery of the laptop is handled by the applet **KPowersave** in the kicker. Complex system monitoring is performed by **KSysguard**. When using GNOME, the described functions are provided by GNOME ACPI (as panel applet) and System Monitor.

KPowersave **KPowersave** is an applet that displays the state of the rechargeable battery in the control panel. The icon adjusts to represent the type of power supply. When working on AC power, a small plug icon is displayed. When working on batteries, the icon changes to a battery. The corresponding menu opens the YaST module for power management after requesting the root password. This allows setting the behavior of the system under different types of power supply. Information about power management and about the corresponding YaST module can be found in *Power Management* on page 313.

KSysguard **KSysguard** is an independent application that gathers all measurable parameters of the system into one monitoring environment. **KSysguard** has monitors for ACPI (battery status), CPU load, network, partitioning, and memory usage. It can furthermore watch and display all system processes. The presentation and filtering of the collected data can be customized. It is possible to monitor different system parameters in various data pages or collect the data of various machines in parallel over the network. **KSysguard** can also run as a daemon on machines without a KDE environment. More information about this program is provided in its integrated help function or in the SUSE help pages.

Synchronizing Data

When switching between working on a mobile machine disconnected from the network and working at a networked workstation in an office, it is necessary to keep the processed data synchronous across all instances. This could include e-mail folders, directories, and individual files that need to be present for work on the road as well as at the office. The solution in both cases is as follows:

Synchronization of E-Mail Use an IMAP account for storing your e-mails in the office network. The e-mails are then accessed from the workstation using any disconnected IMAP-enabled e-mail client, like Mozilla Thunderbird Mail, Evolution, or KMail as described in the *User Guide*. The e-mail client needs to be configured, so the same folder is always accessed

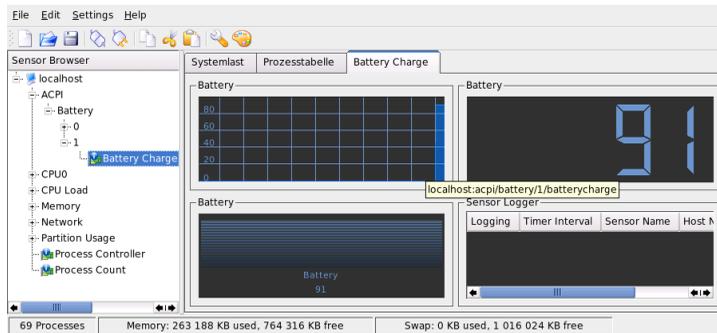


Figure 13.2: Monitoring the Battery State with KSysguard

for Sent messages. This ensures that all messages are available along with their status information after the synchronization process has completed. Use the SMTP service implemented in the mail client for sending messages instead of the system-wide MTA postfix or sendmail to receive reliable feedback about unsent mail.

Synchronizing Files and Directories There are several utilities suitable for synchronizing data between a laptop and a workstation. For detailed information, refer to *File Synchronization* on page 531.

Wireless Communication

As well as connecting to a home or office network with a cable, a laptop can also wirelessly connect to other computers, peripherals, cellular phones, or PDAs. Linux supports three types of wireless communication:

WLAN With the largest range of these wireless technologies, WLAN is the only one suitable for the operation of large and sometimes even spatially disjointed networks. Single machines can connect with each other to form an independent wireless network or access the Internet. Devices called access points act as base stations for WLAN-enabled devices and play intermediate for access to the Internet. A mobile user can switch between various access points, depending on location and which access point is offering the best connection. Like in cellular telephony, a large network is available to

WLAN users without binding them to a specific location for accessing it. Details about WLAN can be found in *Wireless LAN* on page 338.

Bluetooth Bluetooth has the broadest application spectrum of all wireless technologies. It can be used for communication between computers (laptop) and PDAs or cellular phones, as can IrDA. It can also be used to connect various computers within visible range. Bluetooth is also used to connect wireless system components, like a keyboard or mouse. The range of this technology is, however, not sufficient to connect remote systems to a network. WLAN is the technology of choice for communicating through physical obstacles like walls. More information about bluetooth, its applications, and configuration can be found in *Bluetooth* on page 346.

IrDA IrDA is the wireless technology with the shortest range. Both communication parties must be within viewing distance of each other. Obstacles like walls cannot be overcome. One possible application of IrDA is the transmission of a file from a laptop to a cellular phone. The short path from the laptop to the cellular phone is then covered using IrDA. The long range transport of the file to the recipient of the file is handled by the mobile network. Another application of IrDA is the wireless transmission of printing jobs in the office. More information about IrDA can be found in *Infrared Data Transmission* on page 355.

13.1.5 Data Security

Ideally, you protect data on your laptop against unauthorized access in multiple ways. Possible security measures can be taken in the following areas:

Protection against Theft Always physically secure your system against theft whenever possible. Various securing tools, like chains, are available in retail stores.

Securing Data on the System Important data should not only be encrypted during transmission, but also on the hard disk. This ensures its safety in case of theft. The creation of an encrypted partition with SUSE LINUX is described in *Encrypting Partitions and Files* on page 603.

Network Security Any transfer of data should be secured, no matter how it takes place. General security issues regarding Linux and networks can be found in *Security and Confidentiality* on page 605. Security measures related to wireless networking are provided in *Wireless Communication* on page 337.

13.2 Mobile Hardware

SUSE LINUX supports the automatic detection of mobile storage devices over firewire (IEEE 1394) or USB. The term mobile storage device comprises any kind of firewire or USB hard disk, USB flash drive, or digital camera. These devices are automatically detected and configured via hotplug as soon as they are connected with the system over the corresponding interface. `subfs` and `submount` ensure that the devices are mounted to the corresponding locations in the file system. The user is completely spared the manual mounting and unmounting that was found in previous versions of SUSE LINUX. Device can simply be disconnected as soon as no program accesses it.

External Hard Disks (USB and Firewire)

As soon as an external hard disk has been correctly recognized by the system, its icon appears in 'My Computer' (KDE) or 'Computer' (GNOME) in the list of mounted drives. Left-clicking the icon displays the contents of the drive. It is possible to create folders and files here and edit or delete them. To rename a hard disk from the name it had been given by the system, select the corresponding menu item from the menu that opens when the icon is right-clicked. This name change is limited to display in the file manager. The descriptor by which the device is mounted in `/media/usb-xxx` or `/media/ieee1394-xxx` remains unaffected by this.

USB Flash Drives These devices are handled by the system just like external hard disks. It is similarly possible to rename the entries in the file manager.

Digital Cameras (USB and Firewire) Digital cameras recognized by the system also appear as external drives in the overview of the file manager. KDE allows reading and accessing the pictures at the URL `camera: /`. The images can then be processed using `digikam` or `The GIMP`. When using GNOME, `Nautilus` displays the pictures in their own folder. A simple image processing and management utility is `GThumb`. Advanced photo processing is done with `The GIMP`. These programs are described in the *User Guide* with the exception of `GThumb`. There is also a chapter about digital cameras.

Note**Securing Mobile Data Drives**

Mobile hard disks or flash drives are as prone to theft as laptops. It is recommended to create an encrypted partition on them as described in *Encrypting Partitions and Files* on page 603 to prevent abuse by third parties.

Note

13.3 Cellular Phones and PDAs

A desktop system or a laptop can communicate with a cellular phone via bluetooth or IrDA. Some models support both protocols and some only one of the two. The usage areas for the two protocols and the corresponding extended documentation has already been mentioned in *Wireless Communication* on page 290. The configuration of these protocols on the cellular phones themselves is described in their manuals. The configuration of the Linux side is described in *Bluetooth* on page 346 and *Infrared Data Transmission* on page 355.

The support for synchronizing with handheld devices manufactured by Palm, Inc., is already built into *Evolution* and *Kontact*. Initial connection with the device is, in both cases, easily performed with the assistance of a wizard. Once the support for Palm Pilots is configured, it is necessary to determine which type of data should be synchronized (addresses, appointments, etc.). Both groupware applications are described in the *User Guide*.

The program *KPIlot* as integrated in *Kontact* is also available as an independent utility. It is described in the *User Guide*. The program *KitchenSync* is also available for synchronizing address data.

More information about *Evolution* and *Kontact* can be found in the *User Guide*.

13.4 For More Information

The central point of reference for all questions regarding mobile devices and Linux is <http://tuxmobil.org/>. Various sections of that web site deal with the hardware and software aspects of laptops, PDAs, cellular phones and other mobile hardware.

A similar approach to that of <http://tuxmobil.org/> is made by <http://www.linux-on-laptops.com/>. Information about laptops and handhelds can be found here.

SUSE maintains a mailing list in German dedicated to the subject of laptops. See <http://lists.suse.com/archive/suse-laptop/>. On this list, users and developers discuss all aspects of mobile computing with SUSE LINUX. Postings in English are answered, but the majority of the archived information is only available in German.

In the case of problems with power management with SUSE LINUX on laptops, it is advisable to read the file `README` in `/usr/share/doc/packages/powersave`. This directory often contains last minute feedback by testers and developers, so provides valuable hints for the solution of problems.

PCMCIA

This section covers special aspects of PCMCIA hardware and software as found in laptops. PCMCIA stands for *Personal Computer Memory Card International Association* and is used as a collective term for all related hardware and software.

14.1	Hardware	296
14.2	Software	296
14.3	Configuration	298
14.4	Utilities	300
14.5	Troubleshooting	300
14.6	For More Information	303

14.1 Hardware

The most important component is the PCMCIA card. There are two types of PCMCIA cards:

PC Cards These cards have been around since the dawn of PCMCIA. They use a 16-bit bus for the data transmission and are usually quite inexpensive. Some modern PCMCIA bridges have difficulties detecting these cards. Nevertheless, once they are detected, they usually run smoothly and do not cause any problems.

CardBus Cards This is a more recent standard. They use a 32-bit bus, which makes them faster but also more expensive. They are integrated in the system like PCI cards and also run smoothly.

If the PCMCIA service is active, the command `cardctl ident` reveals the type of the inserted card. A list of supported cards is available in the file `SUPPORTED.CARDS` in the directory `/usr/share/doc/packages/pcmcia`. The latest version of the PCMCIA HOWTO is also located in this directory.

The second important component is the PCMCIA controller or the PC card or CardBus bridge, which establishes the connection between the card and the PCI bus. All common models are supported. The controller type can be determined with the command `pcic_probe`. If it is a PCI device, the command `lspci -vt` provides further information.

14.2 Software

14.2.1 Base Modules

The required kernel modules are located in the kernel packages. Additionally, the `pcmcia` and `hotplug` packages are needed. When PCMCIA is started, the modules `pcmcia_core`, `yenta_socket`, and `ds` are loaded. In very rare cases, the module `tcic` is needed instead of `yenta_socket`. These modules initialize the existing PCMCIA controllers and provide the base functionality.

14.2.2 Card Manager

Because PCMCIA cards can be changed while the system is running, the activities in the slots must be monitored. This task is handled by the card services implemented in the base module. The initialization of an inserted card is handled by the card manager (for PC cards) or by the kernel's hotplug system (for Card-Bus cards). The card manager is started by the PCMCIA start script after the base modules are loaded. Hotplug is activated automatically.

If a card is inserted, the card manager or hotplug determines its type and function and loads the respective modules. After the modules have been loaded successfully, the card manager or hotplug launches certain initialization scripts, depending on the function of the card. The initialization scripts establish the network connection, mount partitions of external SCSI hard disks, or perform other hardware-specific actions. The scripts of the card manager are located in the directory `/etc/pcmcia`. The scripts for hotplug are located in `/etc/hotplug`. When the card is removed, the card manager or hotplug terminates all card activities with the same scripts. Subsequently, the modules that are no longer needed are unloaded.

These actions are referred to as hotplug events. Whenever hard disks or partitions are added (block events), the hotplug scripts use `subfs` to make the new media available for immediate use in `/media`. To mount media by means of the older PCMCIA scripts, `subfs` should be disabled in hotplug.

Both the start-up of PCMCIA and the card events are logged in the system log file (`/var/log/messages`). The modules that are loaded and the scripts that are executed are recorded in this log file.

Theoretically, a PCMCIA card can be removed without any additional actions. This works perfectly for network, modem, and ISDN cards, provided there are no more active network connections. However, this does not work for mounted partitions of an external hard disk or NFS directories. Such units must be synchronized and unmounted properly. Of course, this is not possible after the card has been taken out. If you are not sure, use the command `cardctl eject` to deactivate all cards that are still inserted in the laptop. To deactivate only one of the cards, specify the slot number, for example, `cardctl eject 0`.

14.3 Configuration

Using the YaST runlevel editor, determine whether PCMCIA should be started when the system is booted. This module can be started with 'System' → 'Runlevel Editor'.

The following three variables are defined in the file `/etc/sysconfig/pcmcia`:

PCMCIA_PCIC Contains the name of the module that controls the PCMCIA controller. Normally, the start script should determine the module automatically. If this fails, enter the module here. Otherwise, this variable should be left empty.

PCMCIA_CORE_OPTS This variable was designed for parameters for the `pcmcia_core` module. However, these parameters are rarely used. The options are described in the manual page of `pcmcia_core`. Because this manual page refers to the homonymous module from the `pcmcia-cs` package from David Hinds, it lists more parameters than the module from the kernel actually supports, namely all parameters beginning with `cb_` and `pc_debug`.

PCMCIA_BEEP Enables and disables the acoustic signals of the card manager.

The files `/etc/pcmcia/config` and `/etc/pcmcia/*.conf` contain the assignment of the drivers to PC cards. First, `config` is read and then `*.conf` in alphabetic order. The last entry found for a card is used. Details about the syntax of these files are available in the manual page of `pcmcia`.

The files designated as `/etc/sysconfig/hardware/hwcfg-
<configurationname>` contain the assignment of drivers to CardBus cards. These files are created by YaST when configuring a card. More information about the configuration names is available in `/usr/share/doc/packages/sysconfig/README` and in the manual page of `getcfg`.

14.3.1 Network Cards

Ethernet, wireless LAN, and TokenRing network cards can be configured with YaST like normal network cards. If your card was not detected, select the card type PCMCIA in the hardware settings. All other details regarding the configuration of the network are provided in *Network Integration* on page 422. Information about hotpluggable cards is provided in *Hotplug and PCMCIA* on page 433.

14.3.2 ISDN

Like other ISDN cards, ISDN PC cards can also be configured with YaST to a large extent. It does not matter which of the listed PCMCIA ISDN cards is selected, as long as it is a PCMCIA card. When configuring the hardware and selecting a provider, the operating mode must always be `hotplug`, not `onboot`. ISDN modems also exist in the form of PCMCIA cards. These are modem cards or multifunction cards with an additional ISDN connection kit. They are treated like modems.

14.3.3 Modem

Normally, there are no PCMCIA-specific settings for modem PC cards. As soon as a modem is inserted, it is available under `/dev/modem`. Some of the PCMCIA modem cards are softmodems that are not supported by Linux. If drivers are available for these cards, they must be installed in the system.

14.3.4 SCSI and IDE

The suitable driver module is loaded by the card manager or by hotplug. As soon as a SCSI or IDE card is inserted, the connected devices can be used. The device names are determined dynamically. Information about available SCSI and IDE devices is available under `/proc/scsi` and under `/proc/ide`.

External hard disks, CD-ROM drives, and similar devices must be switched on before the PCMCIA card is inserted in the slot. Use active termination for SCSI devices.

Caution

Removing SCSI or IDE Cards

Before a SCSI or IDE card is removed, all partitions of the connected devices must be unmounted with the command `umount`. If you forget to do this, you will only be able to access these devices after rebooting the system.

Caution

14.4 Utilities

The above-mentioned `cardctl` utility is the main tool for obtaining PCMCIA information or performing certain actions. Refer to the manual page of `cardctl` for details. Enter `cardctl` for a list of valid options. The graphical front-end `cardinfo` can be used to control the main features of `cardctl`. To use the graphical front-end, install the `pcmcia-cardinfo` package.

`ifport`, `ifuser`, `probe`, and `rcpcmcia` are some other utilities in the `pcmcia` package. However, these are not needed often. To find out which files the `pcmcia` package contains, enter the command `rpm -ql pcmcia`.

14.5 Troubleshooting

Most PCMCIA-related difficulties with laptops or certain cards can be identified and solved by approaching the problem systematically. First, find out if the problem is caused by the card or by the PCMCIA base system. To do this, boot the computer without inserting any card. If the base system seems to work smoothly, insert the card. All messages are logged in `/var/log/messages`. While searching for the cause of the problem, monitor this file with `tail -f /var/log/messages`. In this way, the problem can be narrowed down to one of the following two cases.

14.5.1 PCMCIA Base System Does Not Work

If the system hangs at the message "PCMCIA: Starting services" or other strange things happen when the system is booted, reboot the system and disable PCMCIA by entering `NOPCMCIA=yes` at the boot prompt. To isolate the error, manually load the three base modules of the PCMCIA system one by one.

To load the PCMCIA modules manually, execute the commands `modprobe pcmcia_core`, `modprobe yenta_socket`, and `modprobe ds` as the user `root`. In very rare cases, `tcic`, `i82365`, or `i82092` must be used instead of `yenta_socket`. The two modules that are loaded first are the critical modules.

If the error occurs while loading `pcmcia_core`, refer to the manual page of `pcmcia_core`. The options described in this manual page can first be tested with the command `modprobe`. For example, consider the testing of free I/O areas. Sometimes, this test can cause problems if it disrupts other hardware components. This can be avoided with the option `probe_io=0`:

```
modprobe pcmcia_core probe_io=0
```

If the selected option is successful, set the variable `PCMCIA_CORE_OPTS` in the file `/etc/sysconfig/pcmcia` to the value `probe_io=0`. Separate multiple options with spaces:

```
PCMCIA_CORE_OPTS="probe_io=0 setup_delay=10"
```

Errors that occur while loading the `yenta_socket` module indicate problems of a more basic nature, such as the resource allocation by ACPI.

The files `/etc/pcmcia/config` and `/etc/pcmcia/config.opts` are interpreted by the card manager. The settings they contain are partly used for starting `cardmgr` and partly for loading the driver modules for PC cards. IRQs, I/O ports, and memory address ranges can be included or excluded in the file `/etc/pcmcia/config.opts`. In rare cases, access to an incorrect I/O area can result in a system crash. If this is the case, try to limit these areas.

14.5.2 PCMCIA Card Does Not Work Properly

Basically, there are three types of errors: the card is not detected, the driver cannot be loaded, or the interface provided by the driver is misconfigured. It is important to know whether the card is controlled by the card manager or by hotplug. The card manager controls PC cards, and the hotplug controls CardBus cards.

No Reaction When Inserting a Card If the system does not react when inserting a card and the manual execution of the command `cardctl insert` does not help either, the interrupt allocation to PCI devices may be incorrect. If this is the case, other PCI devices such as the network card may also have problems. In this case, the boot parameter `pci=noacpi` or other PCI or ACPI parameters may be useful.

Card Not Detected If the card is not detected, the message "unsupported Card in Slot x" appears in `/var/log/messages`. This message merely indicates that the card manager is not able to assign a driver to the card. The file `/etc/pcmcia/config` or `/etc/pcmcia/*.conf` is needed for this assignment. This driver database can easily be extended by using existing entries as a template. Enter the command `cardctl ident` to find out details about the card. More information about this is available in the PCMCIA HOWTO (Section 6) and in the manual page of `pcmcia`. After editing `/etc/pcmcia/config` or `/etc/pcmcia/*.conf`, reload driver assignment with the command `rcpcmcia reload`.

Driver Not Loaded One reason for this may be that the driver database contains an incorrect assignment. This may happen if, for example, a manufacturer uses a different chip in an outwardly unmodified card model. Some models may only work or work better with drivers other than the preselected drivers. In this case, you need detailed information about the card. If necessary, post your problem on a mailing list or ask the Advanced Support.

For CardBus cards, the entry `HOTPLUG_DEBUG=yes` must be inserted in the file `/etc/sysconfig/hotplug`. Subsequently, check the system log for messages indicating whether the driver was loaded (successfully).

A resource conflict is another possible cause. For most PCMCIA cards, it does not matter with which IRQ, I/O port, or memory area they are operated, but there may be exceptions. If this is the case, test only one card at a time and disable other system components, such as the sound card, IrDA, modem, and printer, temporarily. View the resource allocation of the system with the command `lsdev` as the user `root`. It is perfectly acceptable if several PCI devices use the same IRQ.

One possible solution is to find a suitable option for the card driver module. Enter `modinfo <driver>` to list the options. A manual page is available for most modules. `rpm -ql pcmcia | grep man` lists all manual pages the `pcmcia` package contains. To test the options, the card drivers can be unloaded manually.

Once you have found a solution, you can generally permit or prohibit the use of a certain resource in `/etc/pcmcia/config.opts`. The options for card drivers can also be entered in this file. For example, to use the `pcnet_cs` module exclusively with IRQ 5, add the following entry:

```
module pcnet_cs opts irq_list=5
```

Misconfigured Interface In this case, carefully check the configuration of the interface and the name of the configuration with `getcfg` to eliminate configuration errors. The variable `DEBUG` in the file `/etc/sysconfig/network/config` and the variable `HOTPLUG_DEBUG` in the file `/etc/sysconfig/hotplug` should be set to `yes`. For other cards or if this does not help, you can add the line `set -vx` in the script executed by the card manager or by `hotplug` (see `/var/log/messages`). In this way, every single command of the script is logged in the system log. If you find a critical section in a script, enter and test the respective commands in a terminal.

14.6 For More Information

Practical information about specific laptop models is available at the Linux Laptop home page <http://linux-laptop.net>. Another source of useful information is the TuxMobil home page at <http://tuxmobil.org/>. These pages provide a laptop howto, an IrDA howto, and a lot of other interesting information. Moreover, the Support Database at <http://portal.suse.com> features several articles about the use of Linux on mobile devices. To find these articles, enter the keyword *notebook* or *laptop* in the search dialog.

System Configuration Profile Management

This chapter introduces SCPM (system configuration profile management). With the help of SCPM, adapt the configuration of your computer to different operating environments or hardware configurations. SCPM manages a set of system profiles for the different scenarios. SCPM enables easy switching between two system profiles, eliminating the need for manually reconfiguring the system.

15.1	Terminology	306
15.2	Configuration	307
15.3	Troubleshooting	310
15.4	For More Information	311

Some situations require a modified system configuration. 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. Restoring the original system configuration should be easy and the modification of the system configuration can be reproduced. With SCPM, any part of the system configuration can be kept in a customized profile.

The main field of application of SCPM 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 customized X server configuration for the multimedia projector at conferences, special power-saving settings for the road, or a different time zone at an overseas subsidiary.

15.1 Terminology

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, such as the use of hard disk 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 metadata, like the 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 of the configuration files of the desired service. SCPM ships with a selection of pre-configured resource groups that should be sufficient for most scenarios.

15.2 Configuration

Basically, two front-ends are available for configuring SCPM. The `scpm` package contains a command-line front-end. The YaST 'Profile Manager' can be used for the graphical configuration. Because both front-ends have the same functionality and an understanding of the command-line front-end greatly enhances understanding of the YaST module, the following instructions mainly focus on the command-line front-end. Special features of the YaST module are addressed where relevant.

15.2.1 Starting SCPM and Defining Resource Groups

SCPM must be activated before use. 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.

By default, SCPM handles network and printer settings as well as the X.Org configuration. To manage special services or configuration files, activate the respective 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.

```
scpm list_groups -a
```

```
nis           Network Information Service client
mail          Mail subsystem
ntpd          Network Time Protocol daemon
xf86          X-Server settings
autofs        Automounter service
network       Basic network settings
printer       Printer settings
```

Activate or deactivate a group with `scpm activate_group NAME` or `scpm deactivate_group NAME`. Replace `NAME` with the relevant group name. All resource groups can also be configured with the YaST profile manager.

15.2.2 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. Then store these modifications in the active profile with `scpm reload`. The `default` profile can be copied and renamed as the basis for new profiles.

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. You may want to modify the system configuration for special purposes and save the changes to 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`.

Profiles can be renamed or deleted with the commands `scpm rename x y` and `scpm delete z`. For example, to rename `work` to `project`, enter `scpm rename work project`. To delete `project`, enter `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.

15.2.3 Switching Configuration Profiles

The command `scpm switch work` switches to another profile (the profile `work`, in this case). Switch to the active profile to include modified settings of the system configuration in the profile. This corresponds to the command `scpm reload`.

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 switch -r work
```

```
Checking for modified resources
Checking for Resources to be started/shut down
Checking for dependencies
Restoring profile default
```

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, such as configuration files, are written, and the system services are restarted.

15.2.4 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

Insert 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 scripts must be executable and refer to the correct interpreter.

Caution

Integrating a Custom Script

Additional scripts to be executed by SCPM must be made readable and executable for the superuser (`root`). The access to these files must be blocked for all other users. Enter the commands `chmod 700 <filename>` and `chown root:root <filename>` to give `root` exclusive permissions to the files.

Caution

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 `" "`. 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`.

15.2.5 Selecting a Profile When Booting the System

To select a profile when booting the system, press `(F4)` during the boot screen to access a list of available profiles. Use the arrow keys to select a profile and confirm your selection with `(Enter)`. The selected profile is then used as a boot option.

15.3 Troubleshooting

15.3.1 Termination during the Switch Process

Sometimes, SCPM may stop working during a switch procedure. This may be caused by some outside effect, such as a user abort, a power failure, another similar problem, or even an error in SCPM itself. If this happens, 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.

15.3.2 Changing the Resource Group Configuration

To modify the configuration of the resource group when SCPM is already initialized, enter `scpm rebuild` after adding or removing groups. In this way, new resources are added to all profiles and the removed resources are deleted permanently. If the deleted resources are configured differently in the various profiles, this configuration data is lost, except for the current version in your system, which SCPM does not touch. If you modify the configuration with YaST, the rebuild command does not need to be entered, because this is handled by YaST.

15.4 For More Information

The latest documentation is available in the SCPM info pages, which you can view with tools like Konqueror or Emacs (`konqueror info:scpm`). In the console, enter `info` or `pinfo`. Information for developers is available in `/usr/share/doc/packages/scpm`.

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.

16.1	Power Saving Functions	314
16.2	APM	315
16.3	ACPI	316
16.4	Rest for the Hard Disk	323
16.5	The powersave Package	324
16.6	The YaST Power Management Module	332

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 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. Because APM largely consists of a function set implemented in the BIOS, the level of APM support may vary depending on the hardware. This is even more true of ACPI, which is even more complex. For this reason, it is virtually impossible to recommend one over the other. Simply test the various procedures on your hardware then select the technology that is best supported.

Note**Power Management for AMD64 Processors**

AMD64 processors with a 64-bit kernel only support ACPI.

Note

16.1 Power Saving Functions

Power saving functions are not only significant for the mobile use of laptops, but also for desktop systems. The following paragraphs briefly introduce the main functions and their use in the power management systems APM and ACPI:

Standby This operating mode turns off the display. On some computers, the processor performance is throttled. This function is not available in all APM implementations. This function corresponds to the ACPI state S1 or S2.

Suspend (to memory) This mode writes the entire system state to the RAM. Subsequently, the entire system except the RAM is put to sleep. In this state, the computer consumes very little power. The advantage of this state is the possibility to resume work at the same point within a few seconds without having to boot and restart applications. Devices using APM can usually be suspended by closing the lid and activated by opening it. This function corresponds to the ACPI state S3. The support of this state is still under development and therefore largely depends on the hardware.

Hibernation (suspend to disk) In this operating mode, the entire system state is written to the hard disk and the system is powered off. The reactivation from this state takes about thirty to ninety seconds. The state prior to the suspend is restored. Some manufacturers offer useful hybrid variants of this mode (such as RediSafe in IBM Thinkpads). The corresponding ACPI state is S4. In Linux, *suspend to disk* is performed by kernel routines that are independent from APM and ACPI.

Battery Monitor ACPI and APM check the battery charge status and provide information about the charge status. Additionally, both system coordinate actions to perform when a critical charge status is reached.

Automatic Power-Off Following a shutdown, the computer is powered off. This is especially important when an automatic shutdown is performed shortly before the battery is empty.

Shutdown of System Components Switching off the hard disk is the greatest single aspect of the power saving potential of the overall system. Depending on the reliability of the overall system, the hard disk can be put to sleep for some time. However, the risk of losing data increases with the duration of the sleep periods. Other components can be deactivated via ACPI (at least theoretically) or permanently in the BIOS setup.

Processor Speed Control In connection with the CPU, energy can be saved in three different ways: frequency and voltage scaling (also known as PowerNow! or Speedstep), throttling, and putting the processor to sleep (C states). Depending on the operating mode of the computer, these methods can also be combined.

16.2 APM

Some of the power saving functions are performed by the APM BIOS itself. On many laptops, standby and suspend states can be activated with key combinations or by closing the lid, without any special operating system function. However, to activate these modes with a command, certain actions must be triggered before the system is suspended. To view the battery charge level, you need special program packages and a suitable kernel.

SUSE LINUX kernels have built-in APM support. However, APM is only activated if ACPI is not implemented in the BIOS and an APM BIOS is detected.

To activate APM support, ACPI must be disabled with `acpi=off` at the boot prompt. Enter `cat /proc/apm` to check if APM is active. An output consisting of various numbers indicates that everything is OK. You should now be able to shut down the computer with the command `shutdown -h`.

BIOS implementations that are not fully standard-compliant can cause problems with APM. Some problems can be circumvented with special boot parameters. All parameters are entered at the boot prompt in the form `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.

The APM daemon (`apmd`) is no longer used. Its functionality is now handled by the new `powersaved` which also supports ACPI and CPU frequency scaling.

16.3 ACPI

ACPI (advanced configuration and power interface) was designed to enable the operating system to set up and control the individual hardware components. ACPI supersedes both PnP and APM. It delivers information about the battery, AC adapter, temperature, fan, and system events, like “close lid” or “battery low.”

The BIOS provides tables containing information about the individual components and hardware access methods. The operating system uses this information for tasks like assigning interrupts or activating and deactivating components. Because the operating system executes commands stored in the BIOS, the functionality depends on the BIOS implementation. The tables ACPI is able to detect and load are reported in `/var/log/boot.msg`. See *Troubleshooting* on page 321 for more information about troubleshooting ACPI problems.

16.3.1 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.

Subsequently, a number of modules must be loaded. This is done by the start script of the ACPI daemon. If any of these modules cause problems, the respective module can be excluded from loading or unloading in `/etc/sysconfig/powersave/common`. The system log (`/var/log/messages`) contains the messages of the modules, enabling you to see which components were detected.

`/proc/acpi` now contains a number of files that provide information about the system state or can be used to change some of the states. Some features do not work yet because they are still under development and the support of some functions largely depends on the implementation of the manufacturer.

All files (except `dsdt` and `fadt`) can be read with `cat`. In some files, settings can be modified with `echo`, for example, `echo X > <file>` to specify suitable values for `X`. Always use the command `powersave` to access this information and control options. The following describes the most important files:

`/proc/acpi/info` General information about ACPI.

`/proc/acpi/alarm` Here, specify when the system should wake from a sleep state. Currently, this feature is not fully supported.

`/proc/acpi/sleep` Provides information about possible sleep states.

`/proc/acpi/event` All events are reported here and processed by the Power-save daemon (`powersaved`). If no daemon accesses this file, events, such as a brief click on the **(Power)** button or closing the lid, can be read with `cat /proc/acpi/event` (terminate with **(Ctrl)-C**).

/proc/acpi/dsdt and /proc/acpi/fadt

These files contain the ACPI tables DSDT (*differentiated system description table*) and FADT (*fixed ACPI description table*). They can be read with `acpidmp`, `acpidisasm`, and `dmdecode`. These programs and their documentation are located in the package `pmtools`. For example, `acpidmp DSDT | acpidisasm`.

/proc/acpi/ac_adapter/AC/state

Shows whether the AC adapter is connected.

/proc/acpi/battery/BAT*/{alarm,info,state}

Detailed information about the battery state. The charge level is read by comparing the last full capacity from `info` with the remaining capacity from `state`. A more comfortable way to do this is to use one of the special programs introduced in *ACPI Tools* on page 321. 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. Activate or deactivate the fan manually by writing 0 (on) or 3 (off) into this file. However, both the ACPI code in the kernel and the hardware (or the BIOS) overwrite this setting when 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 value.

/proc/acpi/processor/CPU*/throttling

Can be used to set the throttling of the processor clock. Usually, throttling is possible in eight levels. This is independent from the frequency control of the CPU.

/proc/acpi/processor/CPU*/limit

If the performance (outdated) and the throttling are automatically controlled by a daemon, the maximum limits can be specified here. Some of the limits are determined by the system. Some can be adjusted by the user.

/proc/acpi/thermal_zone/ A separate subdirectory exists for every thermal zone. A thermal zone is an area with similar thermal properties whose number and names are designated by the hardware manufacturer. However, many of the possibilities offered by ACPI are rarely implemented. Instead, the temperature control is handled conventionally by the BIOS. The operating system is not given much opportunity to intervene, because the life span of the hardware is at stake. Therefore, some of the 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 is always ok.

/proc/acpi/thermal_zone/*/cooling_mode

Select the cooling method controlled by ACPI. Choose from passive (less performance, economical) or active cooling mode (full performance, fan noise).

/proc/acpi/thermal_zone/*/trip_points

Enables the determination of temperature limits for triggering specific actions, like passive or active cooling, suspension (*hot*), or a shutdown (*critical*). The possible actions are defined in the DSDT (device-dependent). The trip points determined in the ACPI specification are *critical*, *hot*, *passive*, *active1*, and *active2*. Even if not all of them are implemented, they must always be entered in this file in this order. For example, the entry `echo 90:0:70:0:0 > trip_points` sets the temperature for *critical* to 90 and the temperature for *passive* to 70 (all temperatures measured in degrees Celsius).

/proc/acpi/thermal_zone/*/polling_frequency

If the value in *temperature* is not updated automatically when the temperature changes, toggle the polling mode here. The command `echo X > /proc/acpi/thermal_zone/*/polling_frequency` causes the temperature to be queried every X seconds. Set X=0 to disable polling.

None of these settings, information, and events need to be edited manually. This can be done with the Powersave daemon (`powersaved`) and various applications, like `powersave`, `kpowersave`, and `wmpowersave`. See *ACPI Tools* on the next page. Because `powersaved` covers the functionalities of the older `acpid`, `acpid` is no longer needed.

16.3.2 Controlling the CPU Performance

The CPU can save energy in three ways. Depending on the operating mode of the computer, these methods can be combined. Saving energy also means that the system heats up less and the fans are activated less frequently.

Frequency and Voltage Scaling PowerNow! and Speedstep are the designations AMD and Intel use for this technology. However, this technology is also applied in processors of other manufacturers. The clock frequency of the CPU and its core voltage are reduced at the same time, resulting in more than linear energy savings. This means that when the frequency is halved (half performance), far less than half of the energy is consumed. This technology is independent from APM or ACPI and requires a daemon that adapts the frequency and the current need for performance. The settings can be made in the directory `/sys/devices/system/cpu/cpu*/cpufreq/`.

Throttling the Clock Frequency This technology omits a certain percentage of the clock signal impulses for the CPU. At 25% throttling, every fourth impulse is omitted, and at 87.5%, only every eighth impulse reaches the processor. However, the energy savings are a little less than linear. Normally, throttling is only used if frequency scaling is not available or to maximize power savings. This technology, too, must be controlled by a special process. The system interface is `/proc/acpi/processor/*/throttling`.

Putting the Processor to Sleep The operating system puts the processor to sleep whenever there is nothing to do. In this case, the operating system sends the CPU a `halt` command. There are three states: C1, C2, and C3. In the most economic state C3, even the synchronization of the processor cache with the main memory is halted. Therefore, this state can only be applied if no other device modifies the contents of the main memory via bus master activity. Some drivers prevent the use of C3. The current state is displayed in `/proc/acpi/processor/*/power`.

Frequency scaling and throttling are only relevant if the processor is busy, because the most economic C state is applied anyway when the processor is idle. If the CPU is busy, frequency scaling is the recommended power saving method. Often the processor only works with a partial load. In this case, it can be run with a lower frequency. Usually, dynamic frequency scaling controlled by a daemon, such as `powersaved`, is the best approach. A static setting to a low frequency is useful for battery operation or if you want the computer to be cool or quiet.

Throttling should be used as the last resort, for example, to extend the battery operation time despite a high system load. However, some systems do not run smoothly when they are throttled too much. Moreover, CPU throttling does not make sense if the CPU has little to do.

In SUSE LINUX these technologies are controlled by the Powersave daemon. The configuration is explained in *The powersave Package* on page 324.

16.3.3 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 `pmt001s`).

16.3.4 Troubleshooting

There are two different types of problems. On one hand, the ACPI code of the kernel may contain bugs that were not detected in time. In this case, a solution will be made available for download. More often, however, the problems are caused by the BIOS. Sometimes, deviations from the ACPI specification are purposely integrated in the BIOS to circumvent errors in the ACPI implementation in other widespread operating systems. Hardware components that have serious errors in the ACPI implementation are recorded in a blacklist that prevents the Linux kernel from using ACPI for these components.

The first thing to do when problems are encountered is to update the BIOS. If the computer does not boot at all, one of the following boot parameters may be helpful:

pci=noacpi Do not use ACPI for configuring the PCI devices.

acpi=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

Monitor the boot messages of the system with the command `dmesg | grep -2i acpi` (or all messages, because the problem may not be caused by ACPI) after booting. If an error occurs while parsing an ACPI table, the most important table — the DSDT — can be replaced with an improved version. In this case, the faulty DSDT of the BIOS is ignored. The procedure is described in *Troubleshooting* on page 329.

In the kernel configuration, there is a switch for activating ACPI debug messages. If a kernel with ACPI debugging is compiled and installed, experts searching for an error can be supported with detailed information.

If you experience BIOS or hardware problems, it is always advisable to contact the manufacturers. Especially if they do not always provide assistance for Linux, they should be confronted with the problems. Manufacturers will only take the issue seriously if they realize that an adequate number of their customers use Linux.

For More Information

Additional documentation and help on ACPI:

- <http://www.cpqlinux.com/acpi-howto.html> (detailed ACPI HOWTO, contains DSDT patches)
- <http://www.intel.com/technology/iapc/acpi/faq.htm> (ACPI FAQ @Intel)
- <http://acpi.sourceforge.net/> (the ACPI4Linux project at Sourceforge)
- <http://www.poupinou.org/acpi/> (DSDT patches by Bruno Ducrot)

16.4 Rest for the Hard Disk

In Linux, the hard disk can be put to sleep entirely if it is not needed, or it can be run in a more economic or quiet mode. On modern laptops, you do not need to switch off the hard disks manually, because they automatically enter an economic operating mode whenever they are not needed. However, if you want to maximize power savings, test some of the following methods. Most of the functionalities can be controlled with `powersaved`.

The `hdparm` application can be used to modify various hard disk settings. The option `-y` instantly switches the hard disk to the standby mode. `-Y` (caution) puts it to sleep. `hdparm -S <x>` causes the hard disk to be spun down after a certain period of inactivity. Replace $\langle x \rangle$ as follows: 0 disables this mechanism, causing the hard disk to run continuously. Values from 1 to 240 are multiplied by five seconds. Values from 241 to 251 correspond to one to eleven times thirty minutes.

Internal power saving options of the hard disk can be controlled with the option `-B`. Select a value from 0 to 255 for maximum saving to maximum throughput. The result depends on the hard disk used and is difficult to assess. To make a hard disk quieter, use the option `-M`. Select a value from 128 to 254 for quiet to fast.

Often, it is not so easy to put the hard disk to sleep. In Linux, numerous processes write to the hard disk, waking it up repeatedly. Therefore, it is important to understand how Linux handles data that needs to be written to the hard disk. First, all data is buffered in the RAM. This buffer is monitored by the kernel update daemon (`kupdated`). When the data reaches a certain age limit or when the buffer is filled to a certain degree, the buffer content is flushed to the hard disk. The buffer size is dynamic and depends on the size of the memory and the system load. By default, `kupdated` is set to short intervals to achieve maximum data integrity. It checks the buffer every 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.

Caution

Impairment of the Data Integrity

Changes to the kernel update daemon settings endanger the data integrity.

Caution

Apart from these processes, journaling file systems, like ReiserFS and Ext3, write their metadata independently from `bdflush`, which also prevents the hard disk from spinning down. To avoid this, a special kernel extension has been developed for mobile devices. See `/usr/src/linux/Documentation/laptop-mode.txt` for details.

Another important factor is the way active programs behave. For example, good editors regularly write hidden backups of the currently modified file to the hard disk, causing the disk to wake up. Features like this can be disabled at the expense of data integrity.

In this connection, the mail daemon `postfix` makes use of the variable `POSTFIX_LAPTOP`. If this variable is set to `yes`, `postfix` will access the hard disk far less frequently. However, this is irrelevant if the interval for `kupdated` was increased.

16.5 The powersave Package

The `powersave` package is responsible for the power saving function in laptops during battery operation. Some of its features are also useful for normal workstations and servers, such as `suspend`, `standby`, ACPI button functionality, and putting IDE hard disks to sleep.

This package contains all power management features of your computer. It supports hardware using ACPI, APM, IDE hard disks, and PowerNow! or SpeedStep technologies. The functionalities from the packages `apmd`, `acpid`, `ospm`, and `cpufreqd` (now `cpuspeed`) have been consolidated in the `powersave` package. Daemons from these packages should not be run concurrently with the `powersave` daemon.

Even if your system does not contain all the hardware elements listed above, use the `powersave` daemon for controlling the power saving function. Because ACPI and APM are mutually exclusive, you can only use one of these systems on your computer. The daemon automatically detects any changes in the hardware configuration.

Note

Information about powersave

Information about the `powersave` package is also available in `/usr/share/doc/packages/powersave`.

Note

16.5.1 Configuring the powersave Package

Normally, the configuration of `powersave` is distributed to several files:

`/etc/sysconfig/powersave/common`

This file contains general settings for the `powersave` daemon. For example, the amount of debug messages (in `/var/log/messages`) can be increased by increasing the value of the variable `POWERSAVE_DEBUG`.

`/etc/sysconfig/powersave/events`

The `powersave` daemon needs this file for processing system events. An event can be assigned external actions or actions performed by the daemon itself. For external actions, the daemon tries to run an executable file in `/usr/lib/powersave/scripts/`. Predefined internal actions:

- `ignore`
- `throttle`
- `dethrottle`
- `suspend_to_disk`
- `suspend_to_ram`
- `standby`
- `do_suspend_to_disk`
- `do_suspend_to_ram`
- `do_standby`

`throttle` slows down the processor by the value defined in `POWERSAVE_MAX_THROTTLING`. This value depends on the current scheme. `dethrottle` sets the processor to full performance. `suspend_to_disk`, `suspend_to_ram`, and `standby` trigger the system event for a sleep mode. These three actions are generally responsible for triggering the sleep mode, but they should always be associated with specific system events.

The directory `/usr/lib/powersave/scripts` contains scripts for processing events:

notify Notification about an event by way of the console, X window, or acoustic signal.

screen_saver Activates the screen saver.

switch_vt Useful if the screen is displaced after a suspend or standby.

wm_logout Saves the settings and logs out from GNOME, KDE, or other window managers.

wm_shutdown Saves the GNOME or KDE settings and shuts down the system.

If, for example, the variable `POWERSAVE_EVENT_GLOBAL_-SUSPEND2DISK="prepare_suspend_to_disk do_suspend_to_disk"` is set, the two scripts or actions are processed in the specified order as soon as the user gives `poweraved` the command for the sleep mode `suspend to disk`. The daemon runs the external script `/usr/lib/powersave/scripts/prepare_suspend_to_disk`. After this script has been processed successfully, the daemon runs the internal action `do_suspend_to_disk` and sets the computer to the sleep mode after the script has unloaded critical modules and stopped services.

The actions for the event of a `(sleep)` button could be modified as in `POWERSAVE_EVENT_BUTTON_SLEEP="notify suspend_to_disk"`. In this case, the user is informed about the suspend by the external script `notify`. Subsequently, the event `POWERSAVE_EVENT_GLOBAL_-SUSPEND2DISK` is generated, resulting in the execution of the above-mentioned actions and a secure system suspend mode. The script `notify` can be customized using the variable `POWERSAVE_NOTIFY_METHOD` in `/etc/sysconfig/powersave/common`.

/etc/sysconfig/powersave/cpufreq

Contains variables for optimizing the dynamic CPU frequency settings.

/etc/sysconfig/powersave/battery

Contains battery limits and other battery-specific settings.

/etc/sysconfig/powersave/sleep

In this file, activate the sleep modes and determine which critical modules should be unloaded and which services should be stopped prior to a suspend or standby event. When the system is resumed, these modules are reloaded and the services are restarted. You can even delay a triggered sleep mode (in order to be able to save files). The default settings mainly concern USB and PCMCIA modules. A failure of suspend or standby is usually caused by certain modules. See *Troubleshooting* on page 329 for more information about identifying the error.

`/etc/sysconfig/powersave/thermal`

Activates cooling and thermal control. Details about this subject are available in the file `/usr/share/doc/packages/powersave/README.thermal`.

`/etc/sysconfig/powersave/scheme_*`

These are the various schemes that adapt the power consumption to certain deployment scenarios. A number of schemes are preconfigured and can be used as they are. Custom schemes can be saved here.

16.5.2 Configuring APM and ACPI

Suspend and Standby

By default, the sleep modes are inactive, because they still do not work on some computers. There are three basic ACPI sleep modes and two APM sleep modes:

Suspend to Disk (ACPI S4, APM suspend)

Saves the entire memory content to the hard disk. The computer is switched off completely and does not consume any power.

Suspend to RAM (ACPI S3, APM suspend)

Saves the states of all devices to the main memory. Only the main memory continues consuming power.

Standby (ACPI S1, APM standby) Switches some devices off (manufacturer-dependent).

Make sure that the following default options are set in the file `/etc/sysconfig/powersave/events` for the correct processing of suspend, standby, and resume (default settings following the installation of SUSE LINUX):

```
POWERSAVE_EVENT_GLOBAL_SUSPEND2DISK=
    "prepare_suspend_to_disk do_suspend_to_disk"
POWERSAVE_EVENT_GLOBAL_SUSPEND2RAM=
    "prepare_suspend_to_ram do_suspend_to_ram"
POWERSAVE_EVENT_GLOBAL_STANDBY=
    "prepare_standby do_standby"
POWERSAVE_EVENT_GLOBAL_RESUME_SUSPEND2DISK=
    "restore_after_suspend_to_disk"
POWERSAVE_EVENT_GLOBAL_RESUME_SUSPEND2RAM=
    "restore_after_suspend_to_ram"
POWERSAVE_EVENT_GLOBAL_RESUME_STANDBY=
    "restore_after_standby"
```

Custom Battery States

In the file `/etc/sysconfig/powersave/battery`, define three battery charge levels (in percent) that trigger system alerts or specific actions when they are reached.

```
POWERSAVED_BATTERY_WARNING=20
POWERSAVED_BATTERY_LOW=10
POWERSAVED_BATTERY_CRITICAL=5
```

The actions or scripts to execute when the charge levels drop under the specified limits is defined in the configuration file `/etc/sysconfig/powersave/events`. The standard actions for buttons can be modified as described in *Configuring the powersave Package* on page 325.

```
POWERSAVE_EVENT_BATTERY_NORMAL="ignore"
POWERSAVE_EVENT_BATTERY_WARNING="notify"
POWERSAVE_EVENT_BATTERY_LOW="notify"
POWERSAVE_EVENT_BATTERY_CRITICAL="wm_shutdown"
```

Adapting Power Consumption to Various Conditions

The system behavior can be adapted to the type of power supply. The power consumption of the system should be reduced when the system is disconnected from the AC power supply and operated with the battery. Similarly, the performance should automatically increase as soon as the system is connected to the AC power supply. The CPU frequency, the power saving function of IDE, and a number of other parameters can be modified.

The actions to execute when the computer is disconnected from or connected to the AC power supply are defined in `/etc/sysconfig/powersave/events`. Select the schemes to use in `/etc/sysconfig/powersave/common`:

```
POWERSAVE_AC_SCHEME="performance"
POWERSAVE_BATTERY_SCHEME="powersave"
```

The schemes are stored in files in `/etc/sysconfig/powersave`. The file names are made in the format `scheme_<name of the scheme>`. The example refers to two schemes: `scheme_performance` and `scheme_powersave`. `performance`, `powersave`, `presentation`, and `acoustic` are preconfigured. Existing schemes can be edited, created, deleted, or associated with different power supply states with the help of the YaST power management module (see *The YaST Power Management Module* on page 332).

16.5.3 Additional ACPI Features

If you use ACPI, you can control the response of your system to *ACPI buttons* (power, sleep, lid open, and lid closed). Configure execution of the actions in `/etc/sysconfig/powersave/events`. Refer to this configuration file for an explanation of the individual options.

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_to_disk"

When the sleep button is pressed, the system is set to the suspend-to-disk 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`.

16.5.4 Troubleshooting

All error messages and alerts are logged in the file `/var/log/messages`. If you cannot find the needed information, increase the verbosity of the messages of `POWERSAVE` using `DEBUG` in the file `/etc/sysconfig/powersave/common`. Increase the value of the variable to 7 or even 15 and restart the daemon. The more detailed error messages in `/var/log/messages` should help you to find the error. The following sections cover the most common problems with `POWERSAVE`.

ACPI Activated with Hardware Support but Functions Do Not Work

If you experience problems with ACPI, use the 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. Ask the manufacturer to comply with the latest ACPI specification. If the errors persist after the BIOS update, proceed as follows to replace the faulty DSDT table in your BIOS with an updated DSDT:

1. Download the DSDT for your system from <http://acpi.sourceforge.net/dsdt/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 is integrated and loaded when the system is booted.

CPU Frequency Does Not Work

Refer to the kernel sources (`kernel-source`) to see if your processor is supported. You may need a special kernel module or module option to activate CPU frequency control. This information is available in `/usr/src/linux/Documentation/cpu-freq/*`. If a special module or module option is needed, configure it in the file `/etc/sysconfig/powersave/cpufreq` by means of the variables `CPUFREQD_MODULE` and `CPUFREQD_MODULE_OPTS`.

Suspend and Standby Do Not Work

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 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.

On ACPI and APM systems: When the system tries to unload faulty modules, the system is arrested or the suspend event is not triggered. The same can also happen if you do not unload modules or stop services that prevent a successful suspend. In both cases, try to identify the faulty module that prevented the sleep mode. The log files generated by the `powerd` daemon in `/var/log/<sleepmode>` are very helpful in this regard. If the computer does not enter the sleep mode, the cause lies in the last module unloaded. Manipulate the following settings in `/etc/sysconfig/powersave/sleep` to unload problematic modules prior to a suspend or standby.

```
POWERSAVE_UNLOAD_MODULES_BEFORE_SUSPEND2DISK=" "  
POWERSAVE_UNLOAD_MODULES_BEFORE_SUSPEND2RAM=" "  
POWERSAVE_UNLOAD_MODULES_BEFORE_STANDBY=" "  
POWERSAVE_SUSPEND2DISK_RESTART_SERVICES=" "  
POWERSAVE_SUSPEND2RAM_RESTART_SERVICES=" "  
POWERSAVE_STANDBY_RESTART_SERVICES=" "
```

If you use `suspend` or `standby` in changing network environments or in connection with remotely mounted file systems, such as Samba and NIS, use `automounter` to mount them or add the respective services, for example, `smbfs` or `nfs`, in the above-mentioned variable. If an application accesses the remotely mounted file system prior to a suspend or standby, the service cannot be stopped correctly and the file system cannot be unmounted properly. After resuming the system, the file system may be corrupt and must be remounted.

Using ACPI, Powersave Does Not Notice Battery Limits

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/sysconfig/powersave/battery` to `yes` to force battery polling.

16.6 The YaST Power Management Module

The YaST power management module can configure all power management settings described above. When starting the module from the YaST Control Center with ‘System’ → ‘Power Management’, the first dialog of the module opens. It is shown in Figure 16.1.

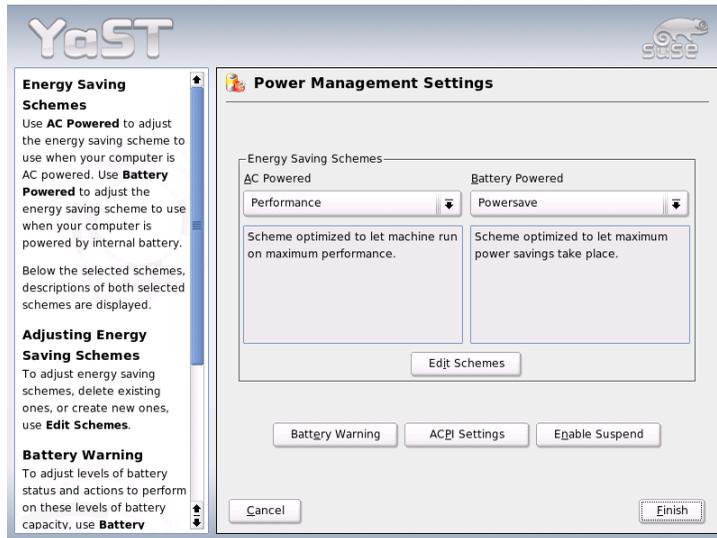


Figure 16.1: Scheme Selection

In this dialog, select the schemes to use for battery operation and AC operation. To add or modify the schemes, click 'Edit Schemes', which opens an overview of the existing schemes like that shown in Figure 16.2.

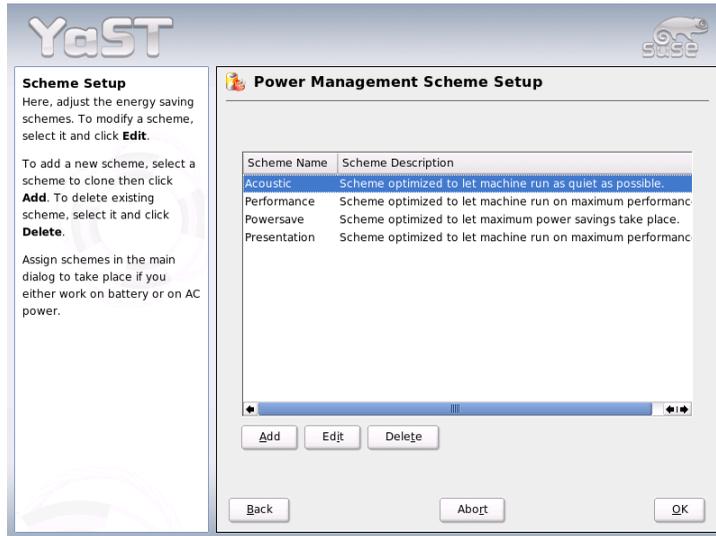


Figure 16.2: Overview of Existing Schemes

In the scheme overview, select the scheme to modify then click 'Edit'. To create a new scheme, click 'Add'. The dialog that opens is the same in both cases and is shown in Figure 16.3 on the next page.

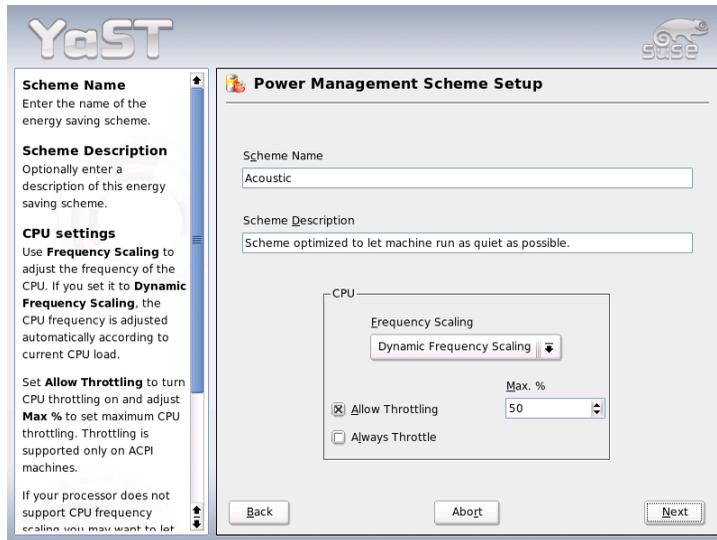


Figure 16.3: Adding a Scheme

First, enter a suitable name and description for the new or edited scheme. Determine if and how the CPU performance should be controlled for this scheme. Decide if and to what extent frequency scaling and throttling should be used. In the following dialog for the hard disk, define a 'Stand-by Policy' for maximum performance or for energy saving. The 'Acoustic Policy' controls the noise level of the hard disk (supported by few hard disks). The 'Cooling Policy' determines the cooling method to use. Unfortunately, this type of thermal control is rarely supported by the BIOS. Read `/usr/share/doc/packages/powersave/README.thermal` to learn how you can use the fan and passive cooling methods. Click 'Next' to proceed to the dialog for configuring the power saving mode of the connected display. Activate the check box 'Enable Screen Saver' to reduce the energy consumption of the display when the computer is not used. 'Enable Display Power Management' allows you to control the time after which the display should enter the standby, suspend, or power-off mode. Complete all settings for the scheme and click 'OK' to return to the start dialog. In the start dialog, assign the custom scheme to one of the two operating modes. To activate your settings, exit this dialog with 'OK'.

Global power management settings can also be made from the initial dialog using 'Battery Warnings', 'ACPI Settings', or 'Enable Suspend'. Click 'Battery Warnings' to access the dialog for the battery charge level, shown in Figure 16.4.

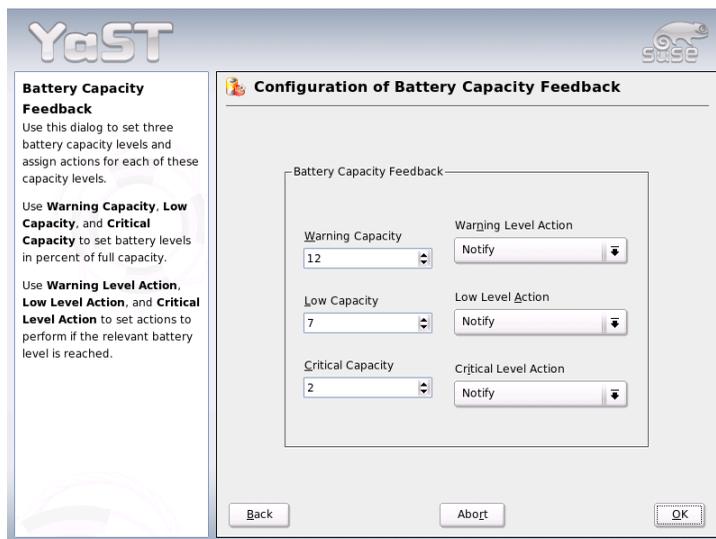


Figure 16.4: Battery Charge Level

The BIOS of your system notifies the operating system whenever the charge level drops under certain configurable limits. In this dialog, define three limits: 'Warning Capacity', 'Low Capacity', and 'Critical Capacity'. Specific actions are triggered when the charge level drops under these limits. Usually, the first two states merely trigger a notification to the user. The third critical level triggers a shut-down, because the remaining energy is not sufficient for continued system operation. Select suitable charge levels and the desired actions then click 'OK' to return to the start dialog.

Access the dialog for configuring the ACPI buttons using 'ACPI Settings'. It is shown in Figure 16.5 on the next page. The settings for the ACPI buttons determine how the system should respond to certain switches. Configure the system response to pressing the power button, pressing the sleep button, and closing the laptop lid. Click 'OK' to complete the configuration and return to the start dialog.

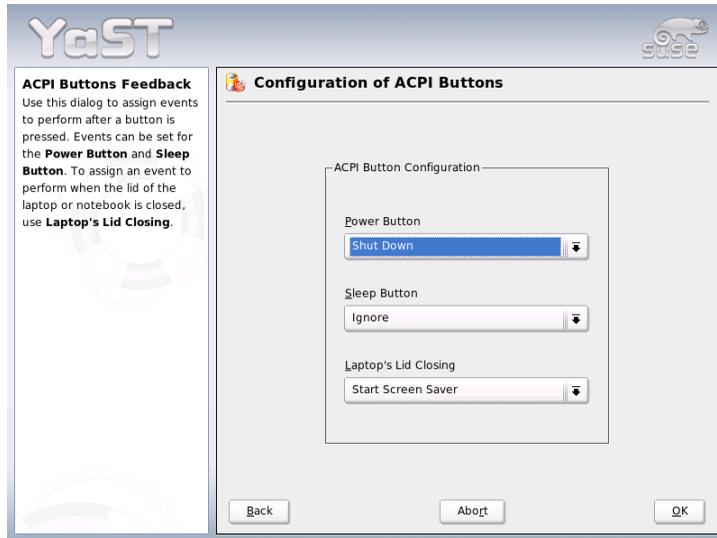


Figure 16.5: ACPI Settings

Click 'Enable Suspend' to enter a dialog in which to determine if and how users of this system may use the suspend or standby functionality. Click 'OK' to return to the main dialog. Click 'OK' again to exit the module and confirm your power management settings.

Wireless Communication

There are several possibilities for using your Linux system to communicate with other computers, cellular phones, or peripheral devices. WLAN (wireless LAN) can be used to network laptops. Bluetooth can be used to connect individual system components (mouse, keyboard), peripheral devices, cellular phones, PDAs, and individual computers with each other. IrDA is mostly used for communication with PDAs or cellular phones. This chapter introduces all three technologies and their configuration.

17.1	Wireless LAN	338
17.2	Bluetooth	346
17.3	Infrared Data Transmission	355

17.1 Wireless LAN

Wireless LANs have become an indispensable aspect of mobile computing. Today, most laptops have built-in WLAN cards. The 802.11 standard for the wireless communication of WLAN cards was prepared by the IEEE organization. Originally, this standard provided for a maximum transmission rate of 2 MBit/s. Meanwhile, several supplements have been added to increase the data rate. These supplements define details such as the modulation, transmission output, and transmission rates:

Table 17.1: Overview of Various WLAN Standards

Name	Band (GHz)	Maximum Transmission Rate (MBit/s)	Note
802.11	2.4	2	Outdated; virtually no end devices available
802.11b	2.4	11	Widespread
802.11a	5	54	Less common
802.11g	2.4	54	Backward-compatible with 11b

Additionally, there are proprietary standards, like the 802.11b variation of Texas Instruments with a maximum transmission rate of 22 MBit/s (sometimes referred to as 802.11b+). However, the popularity of cards using this standard is limited.

17.1.1 Hardware

802.11 cards are not supported by SUSE LINUX. Most cards using 802.11a, 802.11b, and 802.11g are supported. New cards usually comply with the 802.11g standard, but cards using 802.11b are still available. Normally, cards with the following chips are supported:

- Lucent/Agere Hermes
- Intel PRO/Wireless 2100
- Intersil Prism2/2.5/3

- Intersil PrismGT
- Atheros 5210, 5211, 5212
- Atmel at76c502, at76c503, at76c504, at76c506
- Texas Instruments ACX100

A number of older cards that are hardly used and no longer available are also supported. An extensive list of WLAN cards and the chips they use is available at the web site of *AbsoluteValue Systems*: http://www.linux-wlan.org/docs/wlan_adapters.html.gz. <http://wiki.uni-konstanz.de/wiki/bin/view/Wireless/ListeChipsatz> provides an overview of the various WLAN chips.

Some cards need a firmware image that must be loaded into the card when the driver is initialized. This is the case with Intel PRO/Wireless 2100 (Centrino), Intersil PrismGT, Atmel, and ACX100. The firmware can easily be installed with the YaST Online Update. More information about this subject is available in the installed system in `/usr/share/doc/packages/wireless-tools/README.firmware`.

17.1.2 Function

Operating Mode

Basically, wireless networks can be classified as managed networks and ad-hoc networks. Managed networks have a managing element: the access point. In this mode (also referred to as infrastructure mode), all connections of the WLAN stations in the network run over the access point, which may also serve as a connection to an ethernet. Ad-hoc networks do not have an access point. The stations communicate directly with each other. The transmission range and number of participating stations are greatly limited in ad-hoc networks. Therefore, an access point is usually more efficient. It is even possible to use a WLAN card as an access point. Most cards support this functionality.

Because a wireless network is much easier to intercept and compromise than a wired network, the various standards include authentication and encryption methods. In the original version of the IEEE 802.11 standard, these are described under the term WEP. However, because WEP has proved to be insecure (see *Security* on page 344), the WLAN industry (joined under the name *Wi-Fi Alliance*) has defined a new extension called WPA, which is supposed to eliminate the weaknesses of WEP. The later IEEE 802.11i standard (also referred to as WPA2, because WPA is based on a draft version 802.11i) includes WPA and some other authentication and encryption methods.

Authentication

To make sure that only authorized stations can connect, various authentication mechanisms are used in managed networks:

Open An open system is a system that does not require authentication. Any station can join the network. Nevertheless, WEP encryption (see *Encryption* on the current page) can be used.

Shared Key (according to IEEE 802.11)

In this procedure, the WEP key is used for the authentication. However, this procedure is not recommended, because it makes the WEP key more susceptible to attacks. All an attacker needs to do is to listen long enough to the communication between the station and the access point. During the authentication process, both sides exchange the same information, once in encrypted form and once in unencrypted form. Thus, the key can be reconstructed with suitable tools. Because this method makes use of the WEP key for the authentication and for the encryption, it does not enhance the security of the network. A station that has the correct WEP key can authenticate, encrypt, and decrypt. A station that does not have the key cannot be able to decrypt received packets. Accordingly, it cannot communicate, regardless of whether it had to authenticate itself.

WPA-PSK (according to IEEE 802.1x) WPA-PSK (PSK stands for Pre-Shared Key) works similarly to the Shared Key procedure. All participating stations as well as the access point need the same key. The key is 256 bits in length and is usually entered as a passphrase. This system does not need a complex key management like WPA-EAP and is more suitable for private use. Therefore, WPA-PSK is sometimes referred to as WPA “Home”.

WPA-EAP (according to IEEE 802.1x) Actually, WPA-EAP is not an authentication system but a protocol for transporting authentication information. WPA-EAP is used to protect wireless networks in enterprises. In private networks, it is scarcely used. For this reason, WPA-EAP is sometimes referred to as WPA “Enterprise”.

Encryption

There are various encryption methods to ensure that no unauthorized person can read the data packets that are exchanged in a wireless network or gain access to the network:

WEP (defined in IEEE 802.11) This standard makes use of the RC4 encryption algorithm, originally with a key length of 40 bits, later also with 104 bits. Often, the length is declared as 64 bits or 128 bits, depending on whether the 24 bits of the initialization vector are included or not. However, this standard has some weaknesses. Attacks against the keys generated by this system may be successful. Nevertheless, it is better to use WEP than not encrypt the network at all.

TKIP (defined in WPA/IEEE 802.11i) This key management protocol defined in the WPA standard uses the same encryption algorithm as WEP, but eliminates its weakness. As a new key is generated for every data packet, attacks against these keys are in vain. TKIP is used together with WPA-PSK.

CCMP (defined in IEEE 802.11i) CCMP describes the key management. Usually, it is used in connection with WPA-EAP, but it can also be used with WPA-PSK. The encryption takes place according to AES and is stronger than the RC4 encryption of the WEP standard.

17.1.3 Configuration with YaST

To configure your wireless network card, start the YaST 'Network Card' module. In 'Network Address Setup', select the device type 'Wireless' and click 'Next'.

In 'Wireless Network Card Configuration', shown in Figure 17.1 on the following page, make the basic settings for the WLAN operation:

Operating Mode A station can be integrated in a WLAN in three different modes. The suitable mode depends on the network in which to communicate: 'Ad-hoc' (peer-to-peer network without access point), 'Managed' (network is managed by an access point), or 'Master' (your network card is to be used as access point).

Network Name (ESSID) All stations in a wireless network need the same ESSID for communicating with each other. If nothing is specified, the card automatically selects an access point, which may not be the one you intended to use.

Authentication Mode Select a suitable authentication method for your network: 'Open', 'Shared Key', or 'WPA-PSK'. If you select 'WPA-PSK', a network name must be set.

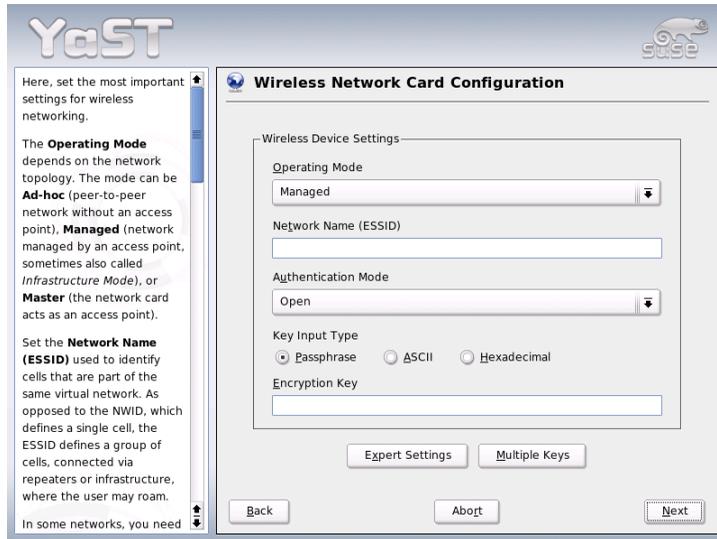


Figure 17.1: YaST: Configuring the Wireless Network Card

Expert Settings This button opens a dialog for the detailed configuration of your WLAN connection. A detailed description of this dialog is provided later.

After completing the basic settings, your station is ready for deployment in the WLAN.

Note

Security in Wireless Networks

Be sure to use one of the supported authentication and encryption methods to protect your network traffic. Unencrypted WLAN connections allow third parties to intercept all network data. Even a weak encryption (WEP) is better than none at all. Refer to *Encryption* on page 340 and *Security* on page 344 for information.

Note

Depending on the selected authentication method, YaST prompts you to fine-tune the settings in another dialog. For 'Open', there is nothing to configure, because this setting implements unencrypted operation without authentication.

WEP Keys Set a key input type. Choose from 'Passphrase', 'ASCII', or 'Hexadecimal'. You may keep up to four different keys to encrypt the transmitted data. Click 'Multiple Keys' to enter the key configuration dialog. Set the length of the key: '128 bit' or '64 bit'. The default setting is '128 bit'. In the list area at the bottom of the dialog, up to four different keys can be specified for your station to use for the encryption. Press 'Set as Default' to define one of them as the default key. Unless you change this, YaST uses the first entered key as the default key. If the standard key is deleted, one of the other keys must be marked manually as the default key. Click 'Edit' to modify existing list entries or create new keys. In this case, a pop-up window prompts you to select an input type ('Passphrase', 'ASCII', or 'Hexadecimal'). If you select 'Passphrase', enter a word or a character string from which a key is generated according to the length previously specified. 'ASCII' requests an input of five characters for a 64-bit key and thirteen characters for a 128-bit key. For 'Hexadecimal', enter ten characters for a 64-bit key or twenty-six characters for a 128-bit key in hexadecimal notation.

WPA-PSK To enter a key for WPA-PSK, select the input method 'Passphrase' or 'Hexadecimal'. In the 'Passphrase' mode, the input must be eight to sixty-three characters. In the 'Hexadecimal' mode, enter sixty-four characters.

Click 'Expert Settings' to leave the dialog for the basic configuration of the WLAN connection and enter the expert configuration. The following options are available in this dialog:

Channel The specification of a channel on which the WLAN station should work is only needed in 'Ad-hoc' and 'Master' modes. In 'Managed' mode, the card automatically searches the available channels for access points. In 'Ad-hoc' mode, select one of the twelve offered channels for the communication of your station with the other stations. In 'Master' mode, determine on which channel your card should offer access point functionality. The default setting for this option is 'Auto'.

Bit Rate Depending on the performance of your network, you may want to set a certain bit rate for the transmission from one point to another. In the default setting 'Auto', the system tries to use the highest possible data transmission rate. Some WLAN cards do not support the setting of bit rates.

Access Point In an environment with several access points, one of them can be preselected by specifying the MAC address.

Use Power Management When you are on the road, use power saving technologies to maximize the operating time of your battery. More information about power management is available in *Power Management* on page 313.

17.1.4 Utilities

hostap (hostap package) is used to run a WLAN card as an access point. More information about this package is available at the project home page (<http://hostap.epitest.fi/>).

kismet (kismet package) is a network diagnosis tool with which to listen to the WLAN packet traffic. In this way, you can also detect any intrusion attempts in your network. More information is available at <http://www.kismetwireless.net/> and in the manual page.

17.1.5 Tips and Tricks for Setting up a WLAN

Stability and Speed

The performance and reliability of a wireless network mainly depends on whether the participating stations receive a clean signal from the other stations. Obstructions like walls greatly weaken the signal. The more the signal strength sinks, the more the transmission slows down. During operation, check the signal strength with the iwconfig utility on the command line ('Link Quality' field) or with kwifiranager in KDE. If you have problems with the signal quality, try to set up the devices somewhere else or adjust the position of the antennas of your access points. Auxiliary antennas that substantially improve the reception are available for a number of PCMCIA WLAN cards. The rate specified by the manufacturer (e.g., 54 MBit/s) is a nominal value that represents the theoretic maximum. In practice, the maximum data throughput is no more than half this value.

Security

If you want to set up a wireless network, remember that anybody within the transmission range can easily access it if no security measures are implemented. Therefore, be sure to activate an encryption method. All WLAN cards and access points support WEP encryption. Although this is not entirely safe, it does

present an obstacle for a potential attacker. WEP is usually adequate for private use. WPA-PSK would be even better, but it is not implemented in older access points or routers with WLAN functionality. On some devices, WPA can be implemented by means of a firmware update. Furthermore, Linux does not support WPA on all hardware components. When this documentation was prepared, WPA only worked with cards using Atheros or Prism2/2.5/3 chips. On the latter, WPA only works if the `hostap` driver is used (see *Problems with Prism2 Cards* on this page). If WPA is not available, WEP is better than no encryption. In enterprises with advanced security requirements, wireless networks should only be operated with WPA.

17.1.6 Troubleshooting

If your WLAN card fails to respond, check if you have downloaded the needed firmware. Refer to *Hardware* on page 338. The following paragraphs cover some known problems.

Multiple Network Devices

Modern laptops usually have a network card and a WLAN card. If you configured both devices with DHCP (automatic address assignment), you may encounter problems with the name resolution and the default gateway. This is evident from the fact that you can ping the router but cannot surf the Internet. The Support Database at <http://portal.suse.com> features an article on this subject. To find the article, enter “DHCP” in the search dialog.

Problems with Prism2 Cards

Several drivers are available for devices with Prism2 chips. The various cards work more or less smoothly with the various drivers. With these cards, WPA is only possible with the `hostap` driver. If such a card does not work properly or not at all or you want to use WPA, read `/usr/share/doc/packages/wireless-tools/README.prism2`.

WPA

WPA support has been implemented for the first time in SUSE LINUX. In Linux, WPA support is still under development. Thus, YaST only allows the configuration of WPA-PSK. WPA does not work with many cards. To enable WPA, some of these cards need a firmware update. If you want to use WPA, read `/usr/share/doc/packages/wireless-tools/README.wpa`.

17.1.7 For More Information

The Internet pages of Jean Tourrilhes, who developed the *Wireless Tools* for Linux, present a wealth of useful information about wireless networks. See http://www.hp1.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html.

17.2 Bluetooth

Bluetooth is a wireless technology for connecting various devices, such as cellular phones, PDAs, peripheral devices, laptops, or system components like the keyboard or mouse. The name is derived from the Danish king Harold Bluetooth, who united various warring factions in Scandinavia. The Bluetooth logo is based on the runes for “H” (resembles a star) and “B”.

A number of important aspects distinguish Bluetooth from IrDA. First, the individual devices do not need to “see” each other directly and, second, several devices can be connected in a network. However, the maximum data rate is 720 Kbps (in the current version 1.2). Theoretically, Bluetooth can even communicate through walls. In practice, however, this depends on the properties of the wall and the device class. There are three device classes with transmission ranges between ten and a hundred meters.

17.2.1 Basics

Software

To be able to use Bluetooth, you need a Bluetooth adapter (either a built-in adapter or an external device), drivers, and a Bluetooth protocol stack. The Linux kernel already contains the basic drivers for using Bluetooth. The BlueZ system is used as protocol stack. To make sure the applications work with Bluetooth, the base packages `bluez-libs` and `bluez-utils` must be installed. These packages provide a number of needed services and utilities. Additionally, some adapters (Broadcom, AVM BlueFritz!) require the `bluez-firmware` package to be installed. The former packages `bluez-pan` and `bluez-sdp` have been integrated in the base packages. The `bluez-cups` package enables printing over Bluetooth connections.

General Interaction

A Bluetooth system consists of four interlocked layers that provide the desired functionality:

Hardware The adapter and a suitable driver for the support by the Linux kernel.

Configuration Files Used for controlling the Bluetooth system.

Daemons Services that are controlled by the configuration files and provide the functionality.

Applications The applications allow the functionality provided by the daemons to be used and controlled by the user.

When inserting a Bluetooth adapter, the respective driver is loaded by the hot-plug system. After the driver is loaded, the system checks the configuration files to see if Bluetooth should be started. If this is the case, it determines the services to start. Based on this information, the respective daemons are started. For security reasons, the Bluetooth system is deactivated in the default configuration.

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. Unfortunately, some manufacturers do not comply strictly with the definitions of the individual profiles. Despite this, communication between the devices usually works smoothly.

17.2.2 Configuration

Configuring Bluetooth with YaST

Use the YaST Bluetooth module, shown in Figure 17.2 on the following page, to configure Bluetooth support on your system. As soon as hotplug detects a Bluetooth adapter on your system, Bluetooth is automatically started with the settings configured in this module.

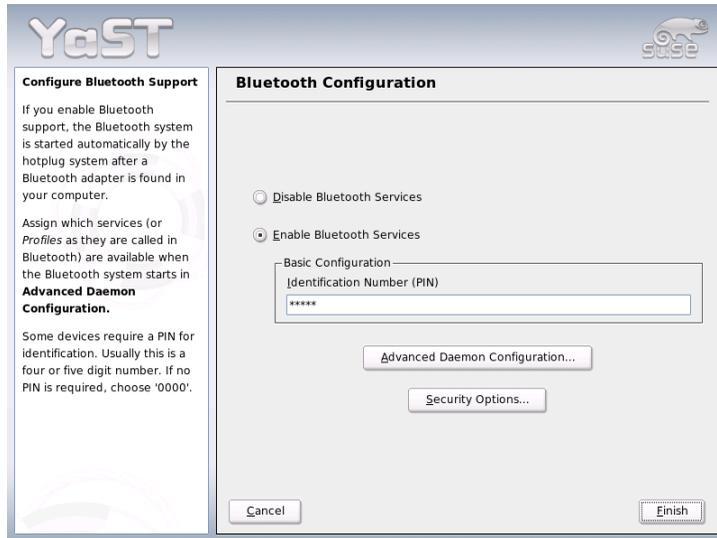


Figure 17.2: YaST Bluetooth Configuration

In the first step of the configuration, determine whether Bluetooth services should be started on your system. If a PIN is needed to establish a connection with the desired partner device, enter the respective number. Then click 'Advanced Daemon Configuration' to enter the dialog for selecting and configuring the available services (called *profiles* in Bluetooth). All available services are displayed in a list and can be enabled or disabled by clicking 'Activate' or 'Deactivate'. Click 'Edit' to open a dialog in which to specify additional arguments for the selected service (daemon). Do not change anything unless you are familiar with the service. After completing the configuration of the daemons, exit this dialog by clicking 'OK'.

From the main dialog, click 'Security Options' to enter the security dialog in which to specify encryption, authentication, and scan settings. Then exit the security dialog to return to the main dialog. After closing the main dialog with 'Finish', your Bluetooth system is ready for use.

If you want to use Bluetooth to set up a network, activate 'PAND' in the 'Advanced Daemon Configuration' dialog and set the mode of the daemon with 'Edit'. For a functional Bluetooth network connection, one `pand` must operate

in the 'Listen' mode and the peer in the 'Search' mode. By default, the 'Listen' mode is preset. Adapt the behavior of your local `pand`. Additionally, configure the `bnepX` interface (`X` stands for the device number in the system) in the YaST 'Network Card' module.

Configuring Bluetooth Manually

The configuration files for the individual components of the BlueZ system are located in the directory `/etc/bluetooth`. The only exception is the file `/etc/sysconfig/bluetooth` for starting the components, which is modified by the YaST module.

The configuration files described below 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. Usually, however, the default settings should be adequate.

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. Because multiple devices cannot be addressed with different PINs, 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 fully secure. By default, the authentication and encryption of Bluetooth connections is deactivated.

Note

Various settings, such as the device names and the security mode, can be changed in the configuration file `/etc/bluetooth/hcid.conf`. Usually, the default settings should be adequate. The file contains comments describing the options for the various settings.

Two sections in the included file are designated as `options` and `device`. The first contains general information that `hcid` uses for starting. The latter contains settings for the individual local Bluetooth devices. Local devices are those physically connected to the computer. All other devices that can only be accessed over wireless connections are referred to as remote devices.

One of the most important settings of the `options` section is `security auto; i`. If set to `auto`, `hcicd` tries to use the local PIN for incoming connections. If it fails, it switches to `none` and establishes the connection anyway. For increased security, this default setting should be set to `user` to make sure that the user is requested to enter a PIN every time a connection is established.

Set the name under which the computer is displayed on the other side in the `device` section. The device class (such `Desktop`, `Laptop`, or `Server`) is defined in this section. Authentication and encryption are also enabled or disabled here.

17.2.3 System Components and Utilities

The operability of Bluetooth depends on the interaction of various services. At least two background daemons are needed: `hcicd` (*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 `hcicd` and `sdpd` can be activated with the command `rcbluetooth start`. This command must be executed as `root`.

The following paragraphs briefly describe the most important shell tools that can be used for working with Bluetooth. Although various graphical components are now available for controlling Bluetooth, it can be worthwhile to check these programs.

Some of the commands can only be executed as `root`. This includes the command `l2ping <device_address>` for testing the connection to a remote device.

hcitool

`hcitool` can be used to determine whether local and remote devices are detected. The command `hcitool dev` lists the local devices. The output generates a line in the form `<interface_name> <device_address>` for every detected local device.

Search for remote devices with the command `hcitool inq`. Three values are returned for every detected device: the device address, the clock offset, and the device class. The device address is important, because other commands use it for identifying the target device. The clock offset mainly serves a technical purpose. The class specifies the device type and the service type as a hexadecimal value.

The command `hcitool name <device-address>` can be used to determine the device name of a remote device. In the case of a remote computer, the class and the device name correspond to the information in its `/etc/bluetooth/hcid.conf`. Local device addresses generate an error output.

hciconfig

The command `/usr/sbin/hciconfig` delivers further information about the local device. If `hciconfig` is executed without any arguments, the output shows device information, such as the device name (`hciX`), the physical device address (a 12-digit number in the form `00:12:34:56:78`), and information about the amount of transmitted data.

`hciconfig hci0 name` displays the name that is returned by your computer when it receives requests from remote devices. As well as querying the settings of the local device, `hciconfig` can be used for modifying these settings. For example, `hciconfig hci0 name TEST` sets the name to `TEST`.

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.

17.2.4 Graphical Applications

In Konqueror, enter the URL `sdp:/` to list local and remote Bluetooth devices. Double-click a device for an overview of the services provided by the device. If you move across one of the specified services with the mouse, the browser's status bar shows which profile is used for the service. If you click a service, a dialog opens, asking what to do: save, use the service (an application must be started to do this), or cancel the action. Mark a check box if you do not want the dialog to be displayed again but always want the selected action to be performed. For some services, support is not yet available. For others, additional packages may need to be installed.

17.2.5 Examples

Network Connection between Two Hosts

In the first example, a network connection is established between the hosts *H1* and *H2*. These two hosts have the Bluetooth device addresses *baddr1* and *baddr2* (determined on both hosts with the command `hcitool dev` as described above). The hosts should be identified with the IP addresses `192.168.1.3` (*H1*) and `192.168.1.4` (*H2*).

The Bluetooth connection is established with the help of `pand` (*Personal Area Networking*). The following commands must be executed by the user `root`. The description focuses on the Bluetooth-specific actions and does not provide a detailed explanation of the network command `ip`.

Enter the command `pand -s` to start `pand` on the host *H1*. Subsequently, a connection can be established on the host *H2* with the command `pand -c <baddr1>`. If you enter `ip link show` on one of the hosts to list the available network interfaces, the output should contain an entry like the following:

```
bnep0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000
link/ether 00:12:34:56:89:90 brd ff:ff:ff:ff:ff:ff
```

Instead of `00:12:34:56:89:90`, the output should contain the local device address *baddr1* or *baddr2*. Now this interface must be assigned an IP address and activated. On *H1*, this can be done with the following two commands:

```
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*, assuming *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.

Data Transfer from a Mobile Phone to the Computer

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 `sdpc`, 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 recognizes 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. Audio tracks can be transferred to the laptop in the same way.

17.2.6 Troubleshooting

If you have difficulties establishing a connection, proceed according to the following list. Remember that the error can be on either side of a connection or even on both sides. If possible, reconstruct the problem with another Bluetooth device to verify that the device is not defective.

Is the local device listed in the output of `hcitool dev`?

If the local device is not listed in this output, `hcid` is not started or the device is not recognized as Bluetooth device. This can have various causes: the device may be defective or the correct driver may be missing. Laptops with built-in Bluetooth often have an on and off switch for wireless devices like WLAN and Bluetooth. Check the manual of your laptop to see if your device has such a switch. Restart the Bluetooth system with the command `rcbluetooth restart` and check if any errors are reported in `/var/log/messages`.

Does your Bluetooth adapter need a firmware file?

If it does, install `bluez-bluefw` and restart the Bluetooth system with `rcbluetooth restart`.

Does the output of `hcitool inq` return other devices?

Test this command more than once. The connection may have interferences, because the frequency band of Bluetooth is also used by other devices.

Do the PINs match? Check if the PIN number of the computer (in `/etc/bluetooth/pin`) matches that of the target device.

Can the remote device “see” your computer?

Try to establish the connection from the remote device. Check if this device sees the computer.

Can a network connection be established (see example 1)?

The first example (network connection) may not work for several reasons. For example, one of the two computers may not support the `ssh` protocol. Try `ping 192.168.1.3` or `ping 192.168.1.4`. If this works, check if `sshd` is active. Another problem could be that one of the two devices already has network settings that conflict with the address `192.168.1.X` in the example. If this is the case, try different addresses, such as `10.123.1.2` and `10.123.1.3`.

Does the laptop appear as a target device (example 2)? 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 `opd` on the laptop. Is `opd` active? Do you have write access to the specified directory?

Does the second example work the other way around?

If the `obexftp` package is installed, this can be done with the command

`obexftp -b <device_address> -B 10 -p <image>` on some devices. Several Siemens and Sony Ericsson models have been tested and found to be functional. Refer to the documentation of the package in `/usr/share/doc/packages/obexftp`.

17.2.7 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/>. Other useful information and instructions:

- Official howto of the Bluetooth protocol stack integrated in the kernel: <http://bluetooth.sourceforge.net/howto/index.html>
- Connection to PalmOS PDA: <http://www.cs.ucl.ac.uk/staff/s.zachariadis/btpalmlinux.html>

17.3 Infrared Data Transmission

IrDA (*Infrared Data Association*) is an industry standard for wireless communication with infrared light. Many laptops sold today are equipped with an IrDA-compatible transceiver that enables communication with other devices, such as printers, modems, LANs, or other laptops. The transfer speed ranges from 2400 bps to 4 Mbps.

There are two IrDA operation modes. The standard mode, SIR, accesses the infrared port through a serial interface. This mode works on almost all systems and is sufficient for most requirements. The faster mode, FIR, requires a special driver for the IrDA chip. Not all chip types are supported in FIR mode because of a lack of appropriate drivers. Set the desired IrDA mode in the BIOS of your computer. The BIOS also shows which serial interface is used in SIR mode.

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 at <http://irda.sourceforge.net/>.

17.3.1 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.

17.3.2 Configuration

The IrDA system service is not started automatically when the system is booted. Use the YaST IrDA module for the activation. Only one setting can be modified in this module: the serial interface of the infrared device. The test window shows two outputs. One is the output of `irdadump`, which logs all sent and received IrDA packets. This output should contain the name of the computer and the names of all infrared devices in transmission range. An example for these messages is shown in *Troubleshooting* on the next page. All devices to which an IrDA connection exists are listed in the lower part of the window.

IrDA consumes a considerable amount of battery power, because a discovery packet is sent every few seconds to detect other peripheral devices. Therefore, IrDA should only be started when necessary if you depend on battery power. Enter the command `rcirda start` to activate it or `rcirda stop` to deactivate it. All needed kernel modules are loaded automatically when the interface is activated.

Manual configuration can be performed in the file `/etc/sysconfig/irda`. This file contains only one variable, `IRDA_PORT`, which determines the interface to use in SIR mode.

17.3.3 Usage

Data can be sent to the device file `/dev/ir1pt0` for printing. The device file `/dev/ir1pt0` acts just like the normal `/dev/lp0` cabled interface, except the printing data is sent wirelessly with infrared light. For printing, make sure that the printer is in visual range of the computer's infrared interface and the infrared support is started.

A printer that is operated over the infrared interface can be configured with the YaST Printer module. Because it is not detected automatically, configure it manually by clicking 'Other (not detected)'. In the following dialog, select 'IrDA printer'. Usually, `ir1pt0` is the right connection. Details about operating printers in Linux are available in *Printer Operation* on page 263.

Communication with other hosts and with mobile phones or other similar devices is conducted through the device file `/dev/ircomm0`. The Siemens S25 and Nokia 6210 mobile phones, for example, can dial and connect to the Internet with the `wvdial` application using the infrared interface. Synchronizing data with a Palm Pilot is also possible, provided the device setting of the corresponding application has been set to `/dev/ircomm0`.

If you want, you can address only devices that support the printer or IrCOMM protocols. Devices that support the IROBEX protocol, such as the 3Com Palm Pilot, can be accessed with special applications, like `irobexpalm` and `irobexreceive`. Refer to the *IR-HOWTO* (<http://tldp.org/HOWTO/Infrared-HOWTO/>) for information. The protocols supported by the device are listed in brackets after the name of the device in the output of `irdadump`. IrLAN protocol support is still a “work in progress.”

17.3.4 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 17.1 appears regularly when a Canon BJC-80 printer is in visible range of the computer:

Example 17.1: Output of irdadump

```
21:41:38.435239 xid:cmd 5b62bed5 > ffffffff S=6 s=0 (14)
21:41:38.525167 xid:cmd 5b62bed5 > ffffffff S=6 s=1 (14)
21:41:38.615159 xid:cmd 5b62bed5 > ffffffff S=6 s=2 (14)
21:41:38.705178 xid:cmd 5b62bed5 > ffffffff S=6 s=3 (14)
21:41:38.795198 xid:cmd 5b62bed5 > ffffffff S=6 s=4 (14)
21:41:38.885163 xid:cmd 5b62bed5 > ffffffff S=6 s=5 (14)
21:41:38.965133 xid:rsp 5b62bed5 < 6cac38dc S=6 s=5 BJC-80
                    hint=8804 [Printer IrCOMM ] (23)
21:41:38.975176 xid:cmd 5b62bed5 > ffffffff S=6 s=* earth
                    hint=0500 [ PnP Computer ] (21)
```

Check the configuration of the interface if there is no output or the other device does not reply. Verify that the correct interface is used. The infrared interface is sometimes located at `/dev/ttyS2` or at `/dev/ttyS3` and an interrupt other than IRQ 3 is sometimes used. These settings can be checked and modified in the BIOS setup menu of almost every laptop.

A simple video camera can also help in determining whether the infrared LED lights up at all. Most video cameras can see infrared light; the human eye cannot.

The Hotplug System

The hotplug system in SUSE LINUX was developed in connection with the *Linux Hotplug project*, but it has a few distinguishing features. The main difference in SUSE LINUX is that instead of using the event multiplexer `/etc/hotplug.d`, the hotplug scripts are executed directly. Whenever possible the scripts `/sbin/hwup` and `/sbin/hwdown` are used to initialize or stop hotplug devices.

18.1	Devices and Interfaces	360
18.2	Hotplug Events	361
18.3	Hotplug Agents	362
18.4	Automatic Module Loading	364
18.5	Hotplug with PCI	365
18.6	The Boot Scripts Coldplug and Hotplug	365
18.7	Error Analysis	366

The hotplug system is not only used for devices that can be inserted and removed during operation, but also for all devices that are only detected after the kernel has been booted. These devices and their interfaces 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, and keyboard, are initialized.

Normally, devices are detected by a driver, which triggers a hotplug event that is handled by suitable scripts. For devices that cannot be detected automatically, `coldplug` uses static configurations.

Apart from a few historic exceptions, most devices are now initialized when the system is booted or when the devices are connected. The initialization often results in the registration of an interface. The registration of the interface, in turn, triggers hotplug events that cause an automatic configuration of the respective interface. Formerly, a set of configuration data was used as the basis for initializing devices. Now, the system searches for suitable configuration data on the basis of existing devices. Thus, the initialization procedure has been reversed, enabling a more flexible use of hotplug devices.

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`. All variables are commented. The second file, `/proc/sys/kernel/hotplug`, contains the name of the executable program called by the kernel. Device configurations are located in `/etc/sysconfig/hardware`.

18.1 Devices and Interfaces

As well as devices, the hotplug system also administers interfaces. A device is linked to either a bus or an interface. An interface links devices to each other or to an application.

A device is always connected to an interface. A bus can be regarded as a multiple interface. There are also virtual devices, such as network tunnels. Every interface is connected to another device or to an application. The distinction of devices or interfaces is important for understanding the overall concept.

Devices entered in `sysfs` are found under `/sys/devices`. Interfaces are located under `/sys/class` or `/sys/block`. All interfaces in `sysfs` should have a link to their devices. However, there are still some drivers that do not automatically add this link.

Devices are addressed by means of a device description. This may be the device path in `sysfs` (`/sys/devices/pci0000:00/0000:00:1e.0/0000:02:00.0`), a description of the connection point (`bus-pci-0000:02:00.0`), an individual ID (`id-32311AE03FB82538`), or something similar. In the past, interfaces were addressed by means of their names. However, these names represent a simple numbering of the existing devices and may change when devices are added or removed. Therefore, interfaces can also be addressed by means of a description of the associated device. Usually, the context indicates whether the description refers to the device itself or to its interface. Typical examples of devices, interfaces, and descriptions include:

PCI Network Card A device that is connected to the PCI bus (`/sys/devices/pci0000:00/0000:00:1e.0/0000:02:00.0` or `bus-pci-0000:02:00.0`) and has a network interface (`eth0`, `id-00:0d:60:7f:0b:22` or `bus-pci-0000:02:00.0`). The network interface is used by network services or connected to a virtual network device, such as a tunnel or VLAN, which in turn has an interface.

PCI SCSI Controller A device (`/sys/devices/pci0000:20/0000:20:01.1/host1/1:0:0:0` or `bus-scsi-1:0:0:0`) 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` or `bus-scsi-1:0:0:0`) with several interfaces (`/sys/block/sda*`).

18.2 Hotplug Events

Every device and every interface has an associated *hotplug event*, which is processed by the responsible hotplug agent. Hotplug events are triggered by the kernel when a link is established to a device or when a driver registers an interface. A hotplug event is a program call, usually `/sbin/hotplug` if nothing else is specified in `/proc/sys/kernel/hotplug`. `/sbin/hotplug` searches for a hotplug agent that matches the type of event. If no suitable agent is found, the program is terminated.

Note

Ignoring Certain Hotplug Events

To ensure that events of a particular kind are ignored, edit the file `/etc/sysconfig/hotplug` and enter the names of the undesired events in the variable `HOTPLUG_SKIP_EVENTS`.

Note

18.3 Hotplug Agents

A hotplug agent is an executable program that performs suitable actions for an event. The agents for device events are located in `/etc/hotplug` and are designated as `<eventname>.agent`. For interface events, all programs in `/etc/dev.d` are executed by `udev`. More information on `udev` can be found in *Dynamic Device Nodes with udev* on page 369.

Usually, device agents load kernel modules, but occasionally they also call additional commands.

In SUSE LINUX, this is handled by `/sbin/hwup` or `/sbin/hwdown`. These programs search for a configuration suitable for the device in the directory `/etc/sysconfig/hardware` and apply it. To prevent a certain device from being initialized, create a suitable configuration file with the start mode `manual` or `off`. If `/sbin/hwup` does not find any configuration, modules are automatically loaded by the agent. For more information, see *Automatic Module Loading* on page 364. More information about `/sbin/hwup` is available in the file `/usr/share/doc/packages/sysconfig/README` and in the manual page of `hwup`.

Interface agents are started indirectly over `udev`. In this way, `udev` first generates a device node the system can access. `udev` enables the assignment of persistent names to interfaces. See *Dynamic Device Nodes with udev* on page 369 for details. Subsequently, the individual agents set up the interfaces. The procedures for some interfaces are described below.

18.3.1 Activating Network Interfaces

Network interfaces are initialized with `/sbin/ifup` and deactivated with `/sbin/ifdown`. Details are provided in the file `/usr/share/doc/packages/sysconfig/README` and in the manual page of the command `ifup`. Because Linux does not use device nodes for network interfaces, these are not managed by `udev`.

If a computer has several network devices with different drivers, the designations of the interface can change if another driver is loaded faster after the system is booted. For this reason, SUSE LINUX manages events for PCI network devices by means of a queue. This feature can be disabled with the variable `HOTPLUG_PCI_QUEUE_NIC_EVENTS=no` in the file `/etc/sysconfig/hotplug`.

The best way to achieve consistent interface designations is to specify the names of the individual interfaces in the configuration files. Details about this method are available in the file `/usr/share/doc/packages/sysconfig/README`.

18.3.2 Activating Storage Devices

Interfaces to storage devices must be mounted to be able to access them. This can be fully automated or preconfigured. The configuration takes place in the variables `HOTPLUG_DO_MOUNT`, `HOTPLUG_MOUNT_TYPE`, and `HOTPLUG_MOUNT_SYNC` in `/etc/sysconfig/hotplug` and in the file `/etc/fstab`.

Fully automated operation can be activated by setting the variable `HOTPLUG_DO_MOUNT=yes`. Use the variable `HOTPLUG_MOUNT_TYPE` to switch between two modes: `subfs` or `fstab`.

In the `HOTPLUG_MOUNT_TYPE=subfs` mode, a directory is created in the directory `/media`. The name of the subdirectory is derived from the device properties. When the medium is accessed, it is automatically mounted and unmounted by `submountd`. Devices in this mode can easily be removed when the access control light goes out.

In the `HOTPLUG_MOUNT_TYPE=fstab` mode, storage devices are mounted in the conventional way by means of a suitable entry in the file `/etc/fstab`. The variable `HOTPLUG_MOUNT_SYNC` can be set to enable access in synchronous or asynchronous mode. In the asynchronous mode, write access is faster, because the results are buffered. However, careless removal of the data medium can result in incomplete writing of data. In the synchronous mode, all data is written immediately, but the access takes longer. The device must be unmounted manually with `umount`.

Fully automated operation can be deactivated by setting the variable `HOTPLUG_DO_MOUNT=no`. In this case, the device must be manually mounted and unmounted.

In the two latter operating modes, the use of persistent device names is recommended, because traditional device names may change depending on the initialization sequence. Details about persistent device names is available in *Dynamic Device Nodes with udev* on page 369.

18.4 Automatic Module Loading

If a device cannot be initialized with `/sbin/hwup`, the agent searches the *module maps* for a suitable driver. First, it searches 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`, because this is the first file read.

The USB agent also searches for user-mode drivers in the files `/etc/hotplug/usb.usermap` and `/etc/hotplug/usb/*.usermap`. User-mode drivers are programs that control access to a device instead of a kernel module. 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, the agent looks in `pci.handmap` and the kernel map. Because `hwinfo` has already looked there, the inquiry must fail. `hwinfo` has an additional database for driver mappings. However, it also loads `pci.handmap` to make sure that any individual mapping in this file is applied.

The agent `pci.agent` can be limited to devices of a certain type or driver modules from a certain subdirectory of `/lib/modules/<kernelversion>/kernel/drivers`. In the first case, the PCI device classes found at the end of the file `/usr/share/pci.ids` can be entered in the variables `HOTPLUG_PCI_CLASSES_WHITELIST` and `HOTPLUG_PCI_CLASSES_BLACKLIST` in the file `/etc/sysconfig/hotplug`. For the second case, specify one or several directories in the variables `HOTPLUG_PCI_DRIVERTYPE_WHITELIST` and `HOTPLUG_PCI_DRIVERTYPE_BLACKLIST`. Modules from the excluded directories are not loaded. In both cases, an empty whitelist implies that all possibilities except for those excluded in the blacklist are permitted. Therefore, enter the modules that should never be loaded by an agent in the file `/etc/hotplug/blacklist`. Write every 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 modules, set the variable `HOTPLUG_LOAD_MULTIPLE_MODULES=yes`. However, it is better to create a separate device configuration `/etc/sysconfig/hardware/hwcfg-*` for this device.

Modules loaded with `hwup` are not affected by this. Automatic module loading only takes place in exceptional cases and will be further limited in future versions of SUSE LINUX.

18.5 Hotplug with PCI

Some computers also allow hotplug for PCI devices. To make full use of this, special kernel modules must be loaded. However, these modules can cause problems on non-PCI hotplug computers. Unfortunately, hotplug PCI slots cannot be autodetected, so configure this function manually. To do this, set the variable `HOTPLUG_DO_REAL_PCI_HOTPLUG` in file `/etc/sysconfig/hotplug` to `yes`.

18.6 The Boot Scripts Coldplug and Hotplug

`boot.coldplug` is responsible for all devices that are not autodetected and for which no hotplug events are generated. It merely calls `hwup` for every static device configuration designated as `/etc/sysconfig/hardware/hwcfg-static-*`. This can also be used to initialize built-in devices in a different order than would be the case with hotplug, because `coldplug` is executed before hotplug.

`boot.hotplug` activates the processing of hotplug events. Because the boot parameter `khelper_max=0` prevents the delivery of hotplug events in the early boot phase, the generated events are still in a queue in the kernel. The number of events that are issued concurrently is set by `boot.hotplug` in the file `/etc/sysconfig/hotplug`. In this way, no hotplug events are lost.

18.7 Error Analysis

18.7.1 Log Files

Unless otherwise specified, `hotplug` only sends a few important messages to `syslog`. To obtain more information, set the variable `HOTPLUG_DEBUG` in the file `/etc/sysconfig/hotplug` to `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. Because `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 available in `/etc/sysconfig/hotplug`.

18.7.2 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. Due to the deactivation of `hotplug`, the kernel does not issue any `hotplug` events. In the running system, you can activate `hotplug` by entering the command `/etc/init.d/boot.hotplug start`. All events generated up to that time are then issued and processed. To reject the queued events, first enter `/bin/true` in `/proc/sys/kernel/hotplug` and reset the entry to `/sbin/hotplug` after some time. Because of the deactivation of `coldplug`, the static configurations are not applied. To apply the static configurations, you can later enter `/etc/init.d/boot.coldplug 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 to load are then listed on the screen before they are actually loaded after `<N>` seconds. You cannot intervene while this is going on.

18.7.3 The Event Recorder

The script `/sbin/hotplugeventrecorder` is executed for every event by `/sbin/hotplug`. If a directory `/events` exists, all `hotplug` events are stored as individual files in this directory. Thus, events can be regenerated for test purposes. If this directory does not exist, nothing is recorded.

18.7.4 Excessive System Load or Too Slow While Booting

When hotplug is started, the value of the variable `HOTPLUG_MAX_EVENTS` in `/etc/sysconfig/hotplug` is passed to the kernel. This value determines how many hotplug events can be processed concurrently. If hotplug causes an excessive system load when the system is booted, reduce this value. If hotplug is too slow, increase this value.

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`.

19.1	Creating Rules	370
19.2	Automization with NAME and SYMLINK	371
19.3	Regular Expressions in Keys	371
19.4	Key Selection	372
19.5	Consistent Names for Mass Storage Devices	373

Traditionally, device nodes were stored in the `/dev` directory on Linux systems. There was a node for every possible type of device, regardless 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, because 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. It provides basic information about devices connected to the system. `sysfs` is mounted under `/sys`.

It is not absolutely necessary for the user to create rules. If a device is connected, 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`. 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`.

19.1 Creating Rules

Before `udev` creates device nodes under `/dev`, it reads the file `/etc/udev/udev.rules`. The first rule that fits a device is used, even if other rules would also apply. Comments are introduced with a hash (`#`). Rules take the following form:

```
key, [key,...] NAME [, SYMLINK]
```

At least one key must be specified, because rules are assigned to devices on the basis of these keys. It is also essential to specify a name. 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 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` compares the serial number to the serial number of the device that is connected to the USB bus. 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`.

19.2 Automization with NAME and SYMLINK

The parameters `NAME` and `SYMLINK` allow the use of operators for automatic assignments. These operators refer to kernel data on the corresponding device. A simple example illustrates 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, such as `camera0` or `camera1`. Another useful operator is `%k`, which is replaced by the standard device name of the kernel, for example, `hda1`. Find a list of all the operators in the man page for `udev`.

19.3 Regular Expressions in Keys

Regular expressions, such as wild cards, can be used in the shell. 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 about the use of regular expressions in `udev` rules can be found in the man page `man udev`.

19.4 Key Selection

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 the kernel uses

ID device number on the bus (for example, PCI bus ID)

PLACE physical point where the device is connected (like on USB)

The keys `ID` and `PLACE` can be useful, but usually the keys `BUS`, `KERNEL`, and `SYSFS{ . . . }` are used. The udev configuration also provides keys that call external scripts and evaluate their results. Further details about this can be found in `man udev`.

The file system `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. 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` that 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, `udevinfo` can again reduce the amount of work you need 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 command `udevinfo -a -p /sys/block/sda`. The two commands can also be combined, as in `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 the output information, look for suitable keys that will not change. Remember that you cannot normally use keys from different directories.

19.5 Consistent Names for Mass Storage Devices

SUSE LINUX comes with scripts that help 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 (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 all the available hard disks with the kernel. Each of them triggers a hotplug block event that calls `udev`. First, `udev` reads the rules to determine whether a symlink needs to be created.

If the driver is loaded via `initrd`, 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 must be activated through the YaST runlevel editor or with the command `insserv boot.udev`.

Note

There are a number of tools and programs that 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, these programs do not work. YaST relies on these tools, so only works with the kernel device designations.

Note

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.

20.1	Terminology	376
20.2	Major File Systems in Linux	376
20.3	Some Other Supported File Systems	382
20.4	Large File Support in Linux	383
20.5	For More Information	384

20.1 Terminology

metadata A file system–internal data structure that assures all the data on disk is properly organized and accessible. Essentially, it is “data about the data.” Almost every file system has its own structure of metadata, which is 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.

20.2 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 change partitions and file systems can be performed using the YaST module.

Note

20.2.1 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. However, because Ext2 does not maintain a journal and uses significantly less memory, it is sometimes faster than other file systems.

Easy Upgradability The code for Ext2 is the strong foundation on which Ext3 could become a highly-acclaimed next-generation file system. Its reliability and solidity were elegantly combined with the advantages of a journaling file system.

20.2.2 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

Because Ext3 is based on the Ext2 code and shares its on-disk format as well as its metadata format, upgrades from Ext2 to Ext3 are incredibly easy. Unlike transitions to other journaling file systems, such as ReiserFS, 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, because recreating an entire file system from scratch might not work flawlessly. Considering the number of existing Ext2 systems that await an upgrade to a journaling file system, you can easily figure out why Ext3 might be of some importance to many system administrators. Downgrading from Ext3 to Ext2 is as easy as the upgrade. Just perform a clean unmount of the Ext3 file system and remount it as an Ext2 file system.

Reliability and Performance Other journaling file systems follow the “metadata-only” journaling approach. This means your metadata is always kept in a consistent state but the same cannot be automatically guaranteed for the file system data itself. Ext3 is designed to take care of both metadata and data. The degree of “care” can be customized. Enabling Ext3 in the `data=journal` mode offers maximum security (data integrity), but can slow down the system because both metadata and data are journaled. A relatively new approach is to use the `data=ordered` mode, which ensures both data and metadata integrity, but uses journaling only for metadata. The file system driver collects all data blocks that correspond to one metadata update. These 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.

20.2.3 Converting an Ext2 File System into Ext3

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`.

20.2.4 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 because small files can be stored directly in the B^{*} tree leaf nodes instead of being stored elsewhere and just maintaining a pointer to the actual disk location. In addition to that, storage is not allocated in chunks of 1

or 4 kB, but in portions of the exact size needed. Another benefit lies in the dynamic allocation of inodes. This keeps the file system more flexible than traditional file systems, like Ext2, where the inode density must be specified at file system creation time.

Better Disk Access Performance For small files, 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.

20.2.5 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.

20.2.6 XFS

Originally intended as the file system for their IRIX OS, SGI started XFS development in the early 1990s. The idea behind XFS was to create a high-performance 64-bit journaling file system to meet the extreme computing challenges of today. XFS is very good at manipulating large files and performs well on high-end hardware. However, even XFS has a drawback. Like ReiserFS, XFS takes great care of metadata integrity, but less of data integrity.

A quick review of XFS's key features explains why it may prove a strong competitor for other journaling file systems in high-end computing.

High Scalability through the Use of Allocation Groups

At the creation time of an XFS file system, the block device underlying the file system is divided into eight or more linear regions of equal size. Those are referred to as *allocation groups*. Each allocation group manages its own inodes and free disk space. Practically, allocation groups can be seen as file systems in a file system. Because allocation groups are rather independent of each other, more than one of them can be addressed by the kernel simultaneously. This feature is the key to XFS's great scalability. Naturally, the concept of independent allocation groups suits the needs of multiprocessor systems.

High Performance through Efficient Management of Disk Space

Free space and inodes are handled by B⁺-trees inside the allocation groups. The use of B⁺-trees greatly contributes to XFS's performance and scalability. 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 because the contents of a file are not distributed all over the file system.

20.3 Some Other Supported File Systems

Table 20.1 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 20.1: File System Types in Linux

<code>cramfs</code>	<i>Compressed ROM file system</i> : A compressed read-only file system for ROMs.
<code>hpfs</code>	<i>High Performance File System</i> : The IBM OS/2 standard file system — only supported in read-only mode.
<code>iso9660</code>	Standard file system on CD-ROMs.
<code>minix</code>	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.
<code>msdos</code>	<i>fat</i> , the file system originally used by DOS, is today used by various operating systems.
<code>ncpfs</code>	File system for mounting Novell volumes over networks.
<code>nfs</code>	<i>Network File System</i> : Here, data can be stored on any machine in a network and access may be granted via a network.
<code>smbfs</code>	<i>Server Message Block</i> is used by products such as Windows to enable file access over a network.
<code>sysv</code>	Used on SCO UNIX, Xenix, and Coherent (commercial UNIX systems for PCs).
<code>ufs</code>	Used by BSD, SunOS, and NeXTstep. Only supported in read-only mode.
<code>umsdos</code>	<i>UNIX on MSDOS</i> : Applied on top of a normal <i>fat</i> file system, achieves UNIX functionality (permissions, links, long file names) by creating special files.
<code>vfat</code>	<i>Virtual FAT</i> : Extension of the <i>fat</i> file system (supports long file names).
<code>ntfs</code>	<i>Windows NT file system</i> , read-only.

20.4 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 20.2 offers an overview of the current limitations of Linux files and file systems.

Table 20.2: Maximum Sizes of File Systems (On-Disk Format)

File System	File Size (Bytes)	File System Size (Bytes)
Ext2 or Ext3 (1 kB block size)	2^{34} (16 GB)	2^{41} (2 TB)
Ext2 or Ext3 (2 kB block size)	2^{38} (256 GB)	2^{43} (8 TB)
Ext2 or Ext3 (4 kB block size)	2^{41} (2 TB)	2^{44} (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 (starting with 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 20.2 on the preceding page 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

20.5 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/xfst/>

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.

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.

21.1	Structure of a PAM Configuration File	386
21.2	The PAM Configuration of sshd	388
21.3	Configuration of PAM Modules	389
21.4	For More Information	391

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.

21.1 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).

21.2 The PAM Configuration of sshd

To show how the theory behind PAM works, consider the PAM configuration of sshd as a practical example:

Example 21.1: PAM Configuration for sshd

```

#%PAM-1.0
auth required pam_unix2.so # set_secrcp
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.

21.3 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`.

21.3.1 `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 21.2 on the next page.

Example 21.2: 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`.

21.3.2 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 21.3.

Example 21.3: 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`.

21.3.3 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 21.4 is an example:

Example 21.4: 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`.

21.3.4 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.

21.4 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

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.

22.1	Introduction to TCP/IP	396
22.2	IPv6 — The Next Generation Internet	404
22.3	Manual Network Configuration	412
22.4	Network Integration	422
22.5	Routing in SUSE LINUX	434
22.6	SLP Services in the Network	435
22.7	DNS — Domain Name System	438
22.8	NIS — Network Information Service	459
22.9	LDAP — A Directory Service	464
22.10	NFS — Shared File Systems	486
22.11	DHCP	491
22.12	Time Synchronization with xntp	500

22.1 Introduction to TCP/IP

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 22.1 are provided for the purpose of exchanging data between two 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 22.1: Several Protocols in the TCP/IP Protocol Family

Protocol	Description
TCP	Transmission Control Protocol: A connection-oriented secure protocol. The data to transmit is first sent by the application as a stream of data then converted by the operating system to the appropriate format. The data arrives at the respective application on the destination host in the original data stream format in which it was initially sent. TCP determines whether any data has been lost during the transmission and that there is no mix-up. TCP is implemented wherever the data sequence matters.

UDP	User Datagram Protocol: A connectionless, insecure protocol. The data to transmit is sent in the form of packets generated by the application. The order in which the data arrives at the recipient is not guaranteed and data loss is a possibility. UDP is suitable for record-oriented applications. It features a smaller latency period than TCP.
ICMP	Internet Control Message Protocol: Essentially, this is not a protocol for the end user, but a special control protocol that issues error reports and can control the behavior of machines participating in TCP/IP data transfer. In addition, a special echo mode is provided by ICMP that can be viewed using the program <code>ping</code> .
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.

22.1.1 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 22.1 on the next page.

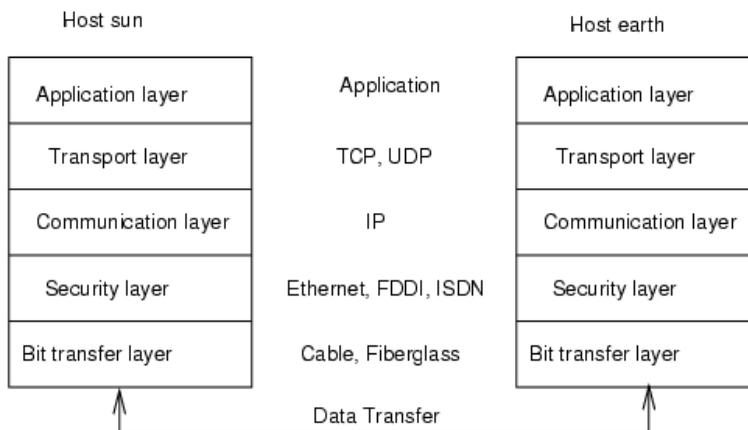


Figure 22.1: Simplified Layer Model for TCP/IP

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 22.2.

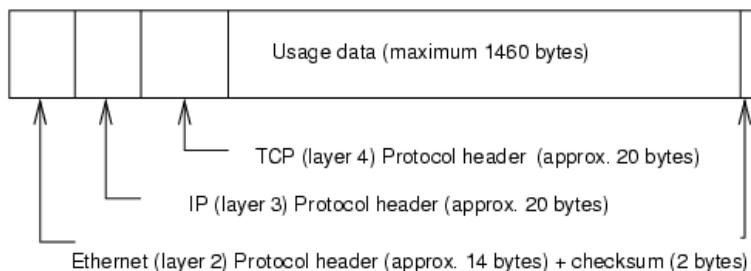


Figure 22.2: TCP/IP Ethernet Packet

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 transmitted, as long as packets are in the correct format.

22.1.2 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 *IPv6 — The Next Generation Internet* on page 404.

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 22.1.

Example 22.1: 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* (CIDR, classless interdomain 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 (“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 are 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 22.2.

Example 22.2: 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 configured.

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) expires.

Table 22.2: Specific Addresses

Address Type	Description
Base network address	This is the netmask AND any address in the network, as shown in Table 22.2 on the preceding page under <code>Result</code> . This address cannot be assigned to any hosts.
Broadcast address	This basically says, "Access all hosts in this subnet-network." 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 22.3 on the next page.

Table 22.3: 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

22.1.3 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 as a *fully qualified domain name* (FQDN), 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 quite confusing for historical reasons. Traditionally, three-letter domain names are used in the USA. In the rest of the world, the two-letter ISO national codes are the standard. In addition to that, 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. 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 *DNS — Domain Name System* on page 438.

The protocol `whois` is closely related to DNS. With this program, quickly find out who is responsible for any given domain.

22.2 IPv6 — The Next Generation Internet

Due to the emergence of the WWW (World Wide Web), the Internet has experienced explosive growth with an increasing number of computers communicating via TCP/IP in the 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 a hundred 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 example, you need a subnetwork with 256 IP addresses, from which only 254 are usable, because two IP addresses are needed for the structure of the subnetwork itself: the broadcast and the base network address.

Under the current IPv4 protocol, DHCP or NAT (network address translation) are the typical mechanisms used to circumvent the potential address shortage. Combined with the convention to keep private and public address spaces separate, these methods can certainly mitigate the shortage. The problem with them lies in their configuration, which is a chore to set up and a burden to maintain. To set up a host in an IPv4 network, you need a number of address items, such as the host's own IP address, the subnetmask, the gateway address, and maybe a name server address. All these items need to be known and cannot be derived from somewhere else.

With IPv6, both the address shortage and the complicated configuration should be a thing of the past. The following sections tell more about the improvements and benefits brought by IPv6 and about the transition from the old protocol to the new one.

22.2.1 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 *The IPv6 Address System* on the next 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 would be impossible to switch the entire Internet from IPv4 to IPv6 at one time. Therefore, it is crucial that both protocols are able to coexist not only on the Internet, but also on one system. This is ensured by compatible addresses 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 *Coexistence of IPv4 and IPv6* on page 411). 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*).

22.2.2 The IPv6 Address System

As mentioned, the current IP protocol is lacking in two important aspects: there is an increasing shortage of IP addresses and configuring the network and maintaining the routing tables is becoming a more complex and burdensome task. IPv6 solves the first problem by expanding the address space to 128 bits. The second one is countered by introducing a hierarchical address structure, combined with sophisticated techniques to allocate network addresses, as well as *multihoming* (the ability to 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 Example 22.3, where all three lines represent the same address.

Example 22.3: Sample IPv6 Address

```
fe80 : 0000 : 0000 : 0000 : 0000 : 10 : 1000 : 1a4
fe80 :    0 :    0 :    0 :    0 : 10 : 1000 : 1a4
fe80 :                               : 10 : 1000 : 1a4
```

Each part of an IPv6 address has a defined function. The first bytes form the prefix and specify the type of address. The center part is the network portion of the address, but it may be unused. The end of the address forms the host part. With IPv6, the netmask is defined by indicating the length of the prefix after a slash at the end of the address. An address, as shown in Example 22.4 on the following page, 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 22.4: 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 22.4.

Table 22.4: Various IPv6 Prefixes

Prefix (hex)	Definition
00	IPv4 addresses and IPv4 over IPv6 compatibility addresses. These are used to maintain compatibility with IPv4. Their use still requires a router able to translate IPv6 packets into IPv4 packets. Several special addresses, (such as the one for the loopback device, have this prefix as well.
2 or 3 as the first digit	Aggregatable global unicast addresses. As is the case with IPv4, an interface can be assigned to form part of a certain subnetwork. Currently, there are the following address spaces: 2001::/16 (production quality address space) and 2002::/16 (6to4 address space).
fe80::/10	Link-local addresses. Addresses with this prefix should not be routed and should therefore only be reachable from within the same subnetwork.
fec0::/10	Site-local addresses. These may be routed, but only within the network of the organization to which they belong. In effect, they are the IPv6 equivalent of the current private network address space (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 to deliver the packet.

Interface ID The third part identifies the interface to which to deliver the packet. This also allows for the MAC to form part of the address. Given that the MAC is a globally unique, fixed identifier coded into the device by the hardware maker, the configuration procedure is substantially simplified. In fact, the first 64 address bits are consolidated to form the EUI-64 token, with the last 48 bits taken from the MAC, and the remaining 24 bits containing special information about the token type. This also makes it possible to assign an EUI-64 token to interfaces that do not have a MAC, such as those based on PPP or ISDN.

On top of this basic structure, IPv6 distinguishes between five different types of unicast addresses:

:: (unspecified) This address is used by the host as its source address when the interface is initialized for the first time — when the address cannot yet be determined by other means.

:::1 (loopback) The address of the loopback device.

IPv4 Compatible Addresses The IPv6 address is formed by the IPv4 address and a prefix consisting of 96 zero bits. This type of compatibility address is used for tunneling (see *Coexistence of IPv4 and IPv6* on page 411) to allow IPv4 and IPv6 hosts to communicate with others operating in a pure IPv4 environment.

IPv4 Addresses Mapped to IPv6 This type of address specifies a pure IPv4 address in IPv6 notation.

Local Addresses There are two address types for local use:

link-local This type of address can only be used in the local subnetwork. Packets with a source or target address of this type should not be routed to the Internet or other subnetworks. These addresses contain a special prefix ($\text{fe80}::/10$) 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 sub-networks, 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 ($\text{fec0}::/10$), 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 additional addresses that belong to the foreign networks where it is roaming. These are called *care-of* addresses. The home network has a facility that forwards any packets destined to the host when it is roaming outside. In an IPv6 environment, this task is performed by the *home agent*, which takes all packets destined to the home address and relays them through a tunnel. On the other hand, those packets destined to the care-of address are directly transferred to the mobile host without any special detours.

22.2.3 Coexistence of IPv4 and IPv6

The migration of all hosts connected to the Internet from IPv4 to IPv6 is a gradual process. Both protocols will coexist for some time to come. The coexistence on one system is guaranteed where there is a *dual stack* implementation of both protocols. That still leaves the question of how an IPv6 enabled host should communicate with an IPv4 host and how IPv6 packets should be transported by the current networks, which are predominantly IPv4 based. The best solutions offer tunneling and compatibility addresses (see *Structure of an IPv6 Address* on page 407).

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

22.2.4 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).

22.3 Manual Network Configuration

Manual configuration of the network software should always be the last alternative. Using YaST is recommended. The background information about the network configuration can assist in your work with YaST.

All built-in network cards as well as hotplug network cards (PCMCIA, USB, some PCI cards) are detected and configured via hotplug. The following information will help you to understand this procedure:

Various Views of Network Cards The system sees a network card in two different ways: first as a physical device and second as an interface. The insertion or detection of a device triggers a hotplug event. This hotplug event triggers the initialization of the device over the script `/sbin/hwup`. When the network card is initialized as a new network interface, the kernel generates another hotplug event that triggers the setup of the interface with `/sbin/ifup`.

Assignment of Interface Names by the Kernel

The kernel numbers interface names according to the temporal order of their registration. The initialization sequence is decisive for the assignment of names. If one of several network card fails, the numbering of all subsequently initialized cards is shifted. For real hotpluggable cards, the order in which the devices are connected is what matters.

To achieve a flexible configuration, the configuration of the device (hardware) and the interface has been separated and the mapping of configurations to devices and interfaces is no longer managed on the basis of the interface names. The device configurations are located in `/etc/sysconfig/hardware/hwcfg-*`. The interface configurations are located in `/etc/sysconfig/network/ifcfg-*`. The names of the configurations are assigned in such a way that they describe the devices and interfaces with which they are associated. Because the former mapping of drivers to interface name requires static interface names, this mapping can no longer take place in `/etc/modprobe.conf`. In the new concept, alias entries in this file would cause undesirable side effects.

The configuration names — everything after `hwcfg-` or `ifcfg-` — can describe the devices by means of the slot, a device-specific ID, or the interface name. For example, the configuration name for a PCI card could be `bus-pci-0000:02:01.0` (PCI slot) or `vpid-0x8086-0x1014-0x0549` (vendor and product ID). The name of the associated interface could be `bus-pci-0000:02:01.0` or `wlan-id-00:05:4e:42:31:7a` (MAC address).

To assign a certain network configuration to any card of a certain type (of which only one is inserted at a time) instead of a certain card, select less specific configuration names. For example, `bus-pcmcia` would be used for all PCMCIA cards. On the other hand, the names can be limited by a preceding interface type. For example, `wlan-bus-usb` would be assigned to WLAN cards connected to a USB port.

The system always uses the configuration that best describes an interface or the device providing the interface. The search for the most suitable configuration is handled by `/sbin/getcfg`. The output of `getcfg` delivers all information that can be used for describing a device. Details regarding the specification of configuration names are available in the manual page of `getcfg`.

With the described method, a network interface is configured with the correct configuration even if the network devices are not always initialized in the same order. However, the name of the interface still depends on the initialization sequence. There are two ways to ensure reliable access to the interface of a certain network card.

- `/sbin/getcfg-interface <configuration name>` returns the name of the associated network interface. Therefore, the configuration name, such as `firewall`, `dhcpd`, `routing`, or various virtual network interfaces (tunnels), can be entered instead of the interface name, which is not persistent, in some (unfortunately not yet in all) configuration files.
- Persistent interface names can be assigned to all interfaces whose configurations do not bear the interface names. This can be done by means of entries `PERSISTENT_NAME=<pname>` names in an interface configuration (`ifcfg-*`). However, the persistent name *<pname>* should not be the same as the name that would automatically be assigned by the kernel. Therefore, `eth*`, `tr*`, `wlan*`, `qeth*`, `iucv*`, and so on are not permitted. Instead, use `net*` or descriptive names like `external`, `internal`, or `dmz`. A persistent name can only be assigned to an interface immediately after its registration, which means that the driver of the network card must be reloaded or `hwup <device description>` must be executed. The command `rcnetwork restart` is not sufficient for this purpose.

Note

Using Persistent Interface Names

The use of persistent interface names has not been tested in all areas. Therefore, some applications may not be able to handle freely selected interface names. If you experience a problem of this kind, inform us using <http://feedback.suse.de>.

Note

`ifup` requires an existing interface, because it does not initialize the hardware. The initialization of the hardware is handled by the command `hwup` (executed by `hotplug` or `coldplug`). When a device is initialized, `ifup` is automatically executed for the new interface via `hotplug` and the interface is set up if the start mode is `onboot`, `hotplug`, or `auto` and the network service was started. Formerly, the command `ifup <interfacename>` triggered the hardware initialization. Now the procedure has been reversed. First, a hardware component is initialized then all other actions follow. In this way, a varying number of devices can always be configured in the best way possible with an existing set of configurations.

The following table summarizes the most important scripts involved in the network configuration. Where possible, the scripts are distinguished by hardware and interface:

Table 22.5: Manual Network Configuration Scripts

Configuration Stage	Command	Function
Hardware	<code>hw{up,down,status}</code>	The <code>hw*</code> scripts are executed by the <code>hotplug</code> subsystem to initialize a device, undo the initialization, or query the status of a device. More information is available in the manual page of <code>hwup</code> .
Interface	<code>getcfg</code>	<code>getcfg</code> can be used to query the interface name associated with a configuration name or a hardware description. More information is available in the manual page of <code>getcfg</code> .
Interface	<code>if{up,down,status}</code>	The <code>if*</code> scripts start existing network interfaces or return the status of the specified interface. More information is available in the manual page of <code>ifup</code> .

More information about `hotplug` and persistent device names is available in *The Hotplug System* on page 359 and *Dynamic Device Nodes with udev* on page 369.

22.3.1 Configuration Files

This section provides an overview of the network configuration files and explains their purpose and the format used.

/etc/sysconfig/hardware/hwcfg-*

These files contain the hardware configurations of network cards and other devices. They contain the needed parameters such as the kernel module, start mode, and script associations. Refer to the manual page of `hwup` for details. Regardless of the existing hardware, the `hwcfg-static-*` configurations are applied when `coldplug` is started.

/etc/sysconfig/network/ifcfg-*

These files contain the configurations for network interface. They include information such as the start mode and the IP address. Possible parameters are described in the manual page of `ifup`. Additionally, all variables from the files `dhcp`, `wireless`, and `config` can be used in the `ifcfg-*` files, if a general setting should be used for only one interface.

/etc/sysconfig/network/config, dhcp, wireless

The file `config` contains general settings for the behavior of `ifup`, `ifdown`, and `ifstatus`. `dhcp` contains settings for DHCP and `wireless` for wireless LAN cards. The variables in all three configuration files are commented and can also be used in `ifcfg-*` files, where they are treated with higher priority.

/etc/sysconfig/network/routes,ifroute-*

The static routing of TCP/IP packets is determined here. In these files, the destination of the route is entered in the first column, the gateway in the second, the network mask of the destination in the third, and an optional network interface in the fourth. Special options can be entered in the fifth column and additional columns. Empty columns are marked with a `-`. Details are provided in the manual page of `routes` and in *Routing in SUSE LINUX* on page 434.

If the network interface is omitted, the system tries to implement the route for every interface. However, this only works for the suitable interface. For example, this can be used for the default route. Of course, configuration names can be used instead of interface names.

To use a route only with a specific interface configuration, it can be entered in `ifroute-<configurationname>` instead of `routes`. In this way, different default routes can be configured. The default route of the last implemented network interface is used.

/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`. Precede comments with `#` signs. YaST enters the specified name server in this file. Example 22.5 shows what `/etc/resolv.conf` could look like.

Example 22.5: /etc/resolv.conf

```
# Our domain
search example.com
#
# We use sun (192.168.0.20) as nameserver
nameserver 192.168.0.20
```

Some services, like `pppd` (`wvdial`), `ippd` (`isdn`), `dhcpcd` and `dhclient`), `pcmcia`, and `hotplug`, modify the file `/etc/resolv.conf` by means of the script `modify_resolvconf`. If the file `/etc/resolv.conf` has been temporarily modified by this script, it contains a predefined comment giving information about the service that modified it, the location where the original file has been backed up, and how to turn off the automatic modification mechanism. If `/etc/resolv.conf` is modified several times, the file includes modifications in a nested form. These can be reverted in a clean way even if this reversal takes place in an order different from the order in which modifications were introduced. Services that may need this flexibility include `isdn`, `pcmcia`, and `hotplug`.

If a service was not terminated in a normal, clean way, `modify_resolvconf` can be used to restore the original file. Also, on system boot, a check is performed to see whether there is an uncleaned, modified `resolv.conf`, for example, after a system crash, in which case the original (unmodified) `resolv.conf` is restored.

YaST uses the command `modify_resolvconf` check to find out whether `resolv.conf` has been modified and subsequently warns the user that changes will be lost after restoring the file. Apart from this, YaST does not rely on `modify_resolvconf`, which means that the impact of changing `resolv.conf` through YaST is the same as that of any manual change. In both cases, changes have a permanent effect. Modifications requested by the above-mentioned services are only temporary.

/etc/hosts

In this file (see Example 22.6), 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, enter a line consisting of the IP address, the fully qualified host name, and the host name (such as `earth`) into the file. The IP address must be at the beginning of the line, the entries separated by blanks and tabs. Comments are always preceded by the `#` sign.

Example 22.6: /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 Example 22.7).

Example 22.7: /etc/networks

```
loopback    127.0.0.0
localnet    192.168.0.0
```

/etc/host.conf

Name resolution — the translation of host and network names via the *resolver* library — is controlled by this file. This file is only used for programs linked to `libc4` or `libc5`. For current `glibc` programs, refer to the settings in `/etc/nsswitch.conf`. A parameter must always stand alone in its own line. Comments are preceded by a `#` sign. Table 22.6 on the facing page shows the parameters available. An example for `/etc/host.conf` is shown in Example 22.8 on the next page.

Table 22.6: *Parameters for /etc/host.conf*

<code>order hosts, bind</code>	Specifies in which order the services are accessed for the name resolution. Available arguments are (separated by blank spaces or commas): <i>hosts</i> : Searches the <code>/etc/hosts</code> file <i>bind</i> : Accesses a name server <i>nis</i> : Via NIS
<code>multi on/off</code>	Defines if a host entered in <code>/etc/hosts</code> can have multiple IP addresses.
<code>nospoof on spoofalert on/off</code>	These parameters influence the name server <i>spoofing</i> , but, apart from that, do not exert any influence on the network configuration.
<code>trim domainname</code>	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 <code>/etc/hosts</code> file, but should still be recognized with the attached domain names.

Example 22.8: */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 details.

The order for queries is defined in the file `/etc/nsswitch.conf`. An example of `nsswitch.conf` is shown in Example 22.9 on the following 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 *DNS — Domain Name System* on page 438).

Example 22.9: /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 22.7. In addition, `automount`, `bootparams`, `netmasks`, and `publickey` are expected in the near future. The configuration options for NSS databases are listed in Table 22.8 on the next page.

Table 22.7: Databases Available via /etc/nsswitch.conf

<code>aliases</code>	Mail aliases implemented by <code>sendmail</code> ; see man 5 <code>aliases</code> .
<code>ethers</code>	Ethernet addresses.
<code>group</code>	For user groups, used by <code>getgrent</code> . See also the man page for <code>group</code> .
<code>hosts</code>	For host names and IP addresses, used by <code>gethostbyname</code> and similar functions.
<code>netgroup</code>	Valid host and user lists in the network for the purpose of controlling access permissions; see man 5 <code>netgroup</code> .
<code>networks</code>	Network names and addresses, used by <code>getnetent</code> .
<code>passwd</code>	User passwords, used by <code>getpwent</code> ; see man 5 <code>passwd</code> .
<code>protocols</code>	Network protocols, used by <code>getprotoent</code> ; see man 5 <code>protocols</code> .
<code>rpc</code>	Remote procedure call names and addresses, used by <code>getrpcbyname</code> and similar functions.

<code>services</code>	Network services, used by <code>getservent</code> .
<code>shadow</code>	Shadow passwords of users, used by <code>getspnam</code> ; see <code>man 5 shadow</code> .

Table 22.8: Configuration Options for NSS Databases

<code>files</code>	directly access files, for example, <code>/etc/aliases</code>
<code>db</code>	access via a database
<code>nis</code>	NIS, see also <i>NIS — Network Information Service</i> on page 459
<code>nisplus</code>	
<code>dns</code>	can only be used as an extension for <code>hosts</code> and <code>networks</code>
<code>compat</code>	can only be used as an extension for <code>passwd</code> , <code>shadow</code> , and <code>group</code>

`/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`. This is important for the performance of directory services, like NIS and LDAP, because otherwise the network connection needs to be used for every access to names or groups. `hosts` is not cached by default, because the mechanism in `nscd` to cache `hosts` 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. Reduce this waiting time by restarting `nscd` with the command `rnscd 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.

22.3.2 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 22.9).

Table 22.9: Some Start-up Scripts for Network Programs

<code>/etc/init.d/network</code>	This script handles the configuration of the network interfaces. The hardware must already have been initialized by <code>/etc/init.d/coldplug</code> (via <code>hotplug</code>). If the <code>network</code> service was not started, no network interfaces will be implemented when they are inserted via <code>hotplug</code> .
<code>/etc/init.d/inetd</code>	Starts <code>xinetd</code> . <code>xinetd</code> can be used to make server services available on the system. For example, it can start <code>vsftpd</code> whenever an FTP connection is initiated.
<code>/etc/init.d/portmap</code>	Starts the portmapper needed for the RPC server, such as an NFS server.
<code>/etc/init.d/nfsserver</code>	Starts the NFS server.
<code>/etc/init.d/sendmail</code>	Controls the <code>sendmail</code> process.
<code>/etc/init.d/ypserv</code>	Starts the NIS server.
<code>/etc/init.d/ypbind</code>	Starts the NIS client.

22.4 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`).

22.4.1 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 `ip address list eth0`. The output should list all information about the network device `eth0` or display an error message.

If the kernel support for the network card is implemented as a module, default for the SUSE kernel, the name of the module must be entered in `/etc/sysconfig/hardware/hwcfg-*`. If nothing is specified there, `hotplug` automatically selects a driver. Regardless of the network card type (hotpluggable or built-in), `hotplug` assigns a driver.

22.4.2 Configuring the Network Card with YaST

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 is 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:

Network Configuration Set the device type of the interface and the configuration name. Select the device type from the options provided. Specify a configuration name according to your needs. Usually, the default settings are useful and can be accepted. Information about the naming conventions for configuration names is available in the manual page of `getcfg`.

Kernel Module 'Hardware Configuration Name' specifies the name of the `/etc/sysconfig/hardware/hwcfg-*` file containing the hardware settings of your network card, for example, the name of the suitable kernel module. Usually, YaST proposes useful names for PCMCIA and USB hardware. For other hardware, 0 usually only makes sense if the card is configured with `hwcfg-static-0`.

If the network card is a PCMCIA or USB device, activate the respective check boxes and exit this dialog with 'Next'. If not, select your network card model from 'Select from List'. YaST automatically selects the suitable kernel module. Exit this dialog with 'Next'.

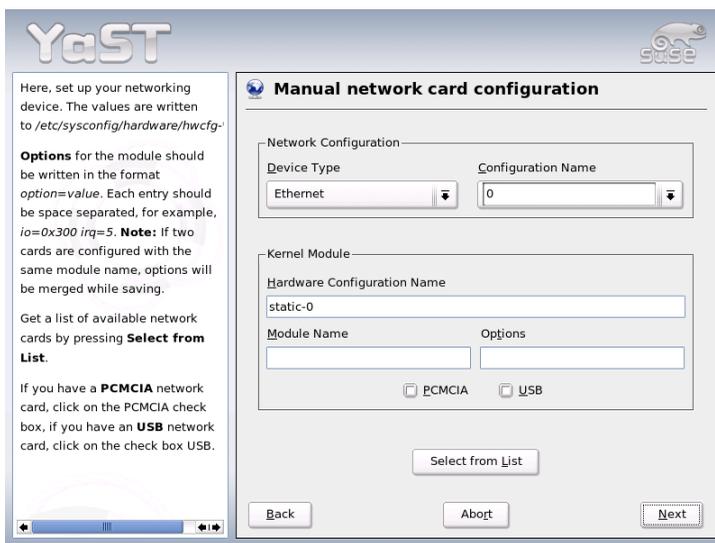


Figure 22.3: Configuration of the Network Card

Setting the Network Address

Set the device type of the interface and the configuration name. Select the device type from those provided. Specify a configuration name according to your needs. Usually, the default settings are useful and can be accepted. Information about the naming conventions for configuration names is available in the manual page of `getcfg`.

If you selected 'Wireless' as the device type of the interface, configure the operating mode, the network name (ESSID), and the encryption in the next dialog, 'Wireless Network Card Configuration'. Click 'OK' to complete the configuration of your card. A detailed description of the configuration of WLAN cards is provided in *Configuration with YaST* on page 341. For all other interface types, proceed with the network address setup:

‘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, configure the details after selecting ‘DHCP Client Options’. Specify whether the DHCP server should always honor broadcast requests and any identifier to use. By default, DHCP servers use the card’s hardware address to identify an interface. If you have a virtual host setup where different hosts communicate through the same interface, an identifier is necessary to distinguish them.

‘Static Address Setup’ If you have a static address, enable the corresponding check box. Then enter the address and subnet mask for your network. The preset subnet mask should match the requirements of a typical home network.

Leave this dialog by selecting ‘Next’ or proceed to configure the host name, name server, and routing details (see *Host Name and DNS* on page 80 and *Routing* on page 83).

‘Advanced...’ enables you to specify more complex settings. Under ‘Details’, use ‘User Controlled’ to delegate the control over the network card from the administrator (`root`) to the normal user. For mobile operation, this allows the user to adapt changing network connections in a more flexible way, because he can control the activation or deactivation of the interface. The MTU (Maximum Transmission Unit) and the type of ‘Device Activation’ can also be set in this dialog.

Cable Modem

In some countries (Austria, US), it is quite common to access the Internet through the TV cable network. The TV cable subscriber usually gets a modem that is connected to the TV cable outlet on one side and to a computer network card on the other (using a 10Base-TG twisted pair cable). The cable modem then provides a dedicated Internet connection with a fixed IP address.

Depending on the instructions provided by your ISP, when configuring the network card either select ‘Automatic Address Setup (via DHCP)’ or ‘Static Address Setup’. Most providers today use DHCP. A static IP address often comes as part of a special business account.

For further information about the configuration of cable modems, read the Support Database article on the topic, which is available online at <http://sdb.suse.de/en/sdb/html/cmodem8.html>.

22.4.3 Modem

In the YaST Control Center, access the modem configuration under 'Network Devices'. If your modem was not automatically detected, open the dialog for manual configuration. In the dialog that opens, enter the interface to which the modem is connected under 'Modem Device'.

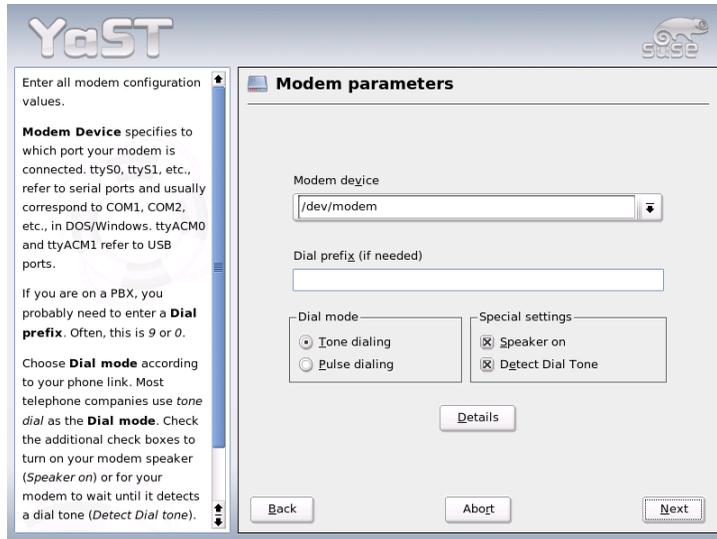


Figure 22.4: Modem Configuration

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. Also select whether to use tone or pulse dialing, whether the speaker should be on, and whether the modem should wait until it detects a dial tone. The latter option should not be enabled if the modem is connected to an exchange.

Under 'Details', set the baud rate and the modem initialization strings. Only change these settings if your modem was not autodetected or if it requires special settings for data transmission to work. This is mainly the case with ISDN terminal adapters. Leave this dialog by clicking 'OK'. To delegate the control over the modem to the normal user without root permissions, activate 'User Controlled'.

In this way, a user without administrator permissions can activate or deactivate an interface. Under 'Dial prefix regex', specify a regular expression. The 'Dial Prefix' in KInternet, which can be modified by the normal user, must match this regular expression. If this field is left empty, the user cannot set a different 'Dial Prefix' without administrator permissions.

In the next dialog, select the ISP (Internet service provider). To choose from a pre-defined list of ISPs operating in your country, select 'Countries'. Alternatively, click 'New' to open a dialog in which to provide the data for your own ISP. This includes a name for the dial-up connection and for the ISP and the login and the password as provided by your ISP. Enable 'Always Ask for Password', to be prompted for the password each time you connect.

The last dialog allows specification of additional connection options:

'Dial on Demand' If you enable dial on demand, specify at least one name server.

'Modify DNS when Connected' This check box 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', disable this and also provide a fixed name server address.

Automatically Retrieve DNS If the provider does not transmit its domain name server after connecting, disable this option and enter the DNS data manually.

'Stupid Mode' This option is enabled by default. It has the effect that input prompts sent by the ISP's 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)' With this option, specify a period of network inactivity after which the modem disconnects automatically.

IP Details This opens the address configuration dialog. If your ISP does not assign a dynamic IP address to your host, disable 'Dynamic IP Address' then enter your host's local IP address and the remote IP address. Ask your ISP for this information. Leave 'Default Route' enabled and close the dialog by selecting 'OK'.

Selecting 'Next' returns to the original dialog, which displays a summary of the modem configuration. Close this dialog with 'Finish'.

22.4.4 DSL

To configure your DSL device, select the 'DSL' module from the YaST 'Network Devices' section. This YaST module consists of several dialogs in which to set the parameters of DSL links based on one of the following protocols:

- PPP over Ethernet (PPPoE)
- PPP over ATM (PPPoATM)
- CAPI for ADSL (Fritz Cards)
- Point-to-Point Tunneling Protocol (PPTP) — Austria

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 *Configuring the Network Card with YaST* on page 423). In the case of a DSL link, addresses may be assigned automatically but not via DHCP, which is why you should not enable the option 'Automatic address setup (via DHCP)'. Instead, enter a static dummy address for the interface, such as 192 . 168 . 22 . 1. In 'Subnet Mask', enter 255 . 255 . 255 . 0. If you are configuring a stand-alone workstation, 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

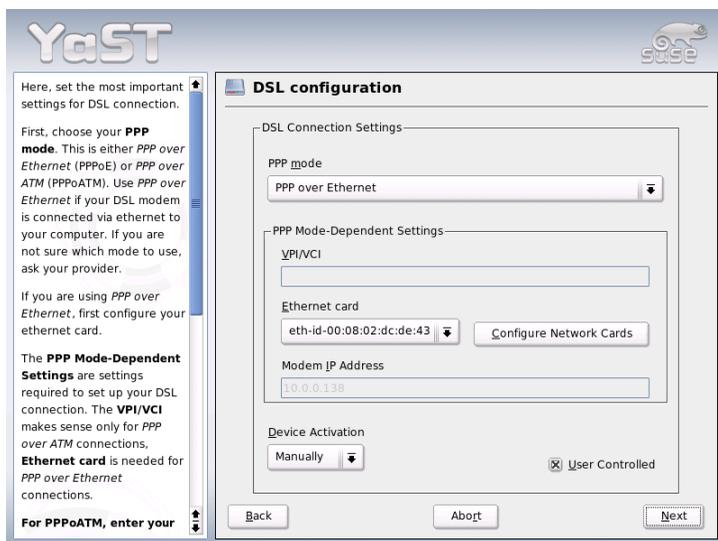


Figure 22.5: DSL Configuration

To begin the DSL configuration (see Figure 22.5), first select the PPP mode and the ethernet card to which the DSL modem is connected (in most cases, this is `eth0`). Then use 'Device Activation' to specify whether the DSL link should be established during the boot process. Click 'User Controlled' to authorize the normal user without root permissions to activate or deactivate the interface over KIn-ternet. 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 depend on the options set so far, which is why they are only briefly mentioned in the following paragraphs. For details on the available options, read the detailed help available from the dialogs.

To use 'Dial on Demand' on a stand-alone workstation, also specify the name server (DNS server). Most ISPs support dynamic DNS — the IP address of a name server is sent by the ISP each time you connect. For a single workstation, however, provide a placeholder address like `192.168.22.99`. If your ISP does not support dynamic DNS, enter the name server IP address provided by your ISP.

'Idle Time-out (seconds)' defines a period of network inactivity after to terminate the connection automatically. A reasonable time-out value is 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 remains in a standby mode and is reestablished automatically as soon as a program requests some kind of data traffic. If 'Dial on Demand' is disabled, the connection is completely terminated and must be reestablished manually when needed. It may then be useful to set the time-out to zero to prevent automatic hang-up.

Note

The configuration of T-DSL is very similar to the DSL setup. Just select 'T-Online' as your provider and YaST opens the T-DSL configuration dialog. In this dialog, provide some additional information required for T-DSL — the line ID, the T-Online number, the user code, and your password. All of these should be included in the information you received after subscribing to T-DSL.

22.4.5 ISDN

Use this module to configure one or several ISDN cards for your system. If YaST did not autodetect your ISDN card, manually select it. Multiple interfaces are possible, but several ISPs can be configured for one interface. In the subsequent dialogs, set the ISDN options necessary for the proper functioning of the card.

In the next dialog, shown in Figure 22.6 on the next page, select the protocol to use. The default is 'Euro-ISDN (EDSS1)' (see points 1. and 2.a below), but for older or larger exchanges, select '1TR6' (see point 2.b below). If you are in the US, select 'NI1'. Select your country in the relevant field. The corresponding country code then appears in the field next to it. Finally, provide your 'Area Code' and the dial prefix (if necessary).

'Start Mode' defines how the ISDN interface should be started. 'OnBoot' causes the ISDN driver to be initialized each time during the boot process. 'Manual' requires you to load the ISDN driver as `root` with the command `rcisdn start`.

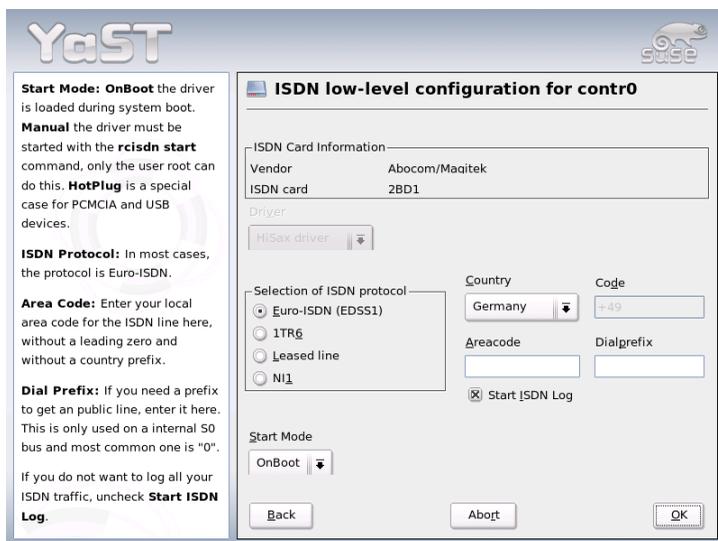


Figure 22.6: ISDN Configuration

‘Hotplug’, used for PCMCIA or USB devices, loads the driver after the device is plugged in. When finished with all these settings, select ‘OK’.

In the next dialog, specify the interface type for your ISDN card and add ISPs to an existing interface. Interfaces may be either the `SynCPpp` or the `RawIP` type, but most ISP operate in the `SynCPpp` mode, which is described below.

The number to enter for ‘My Phone Number’ varies depending on your particular setup:

ISDN card directly connected to phone outlet

A standard ISDN line provides three phone numbers (called multiple subscriber numbers, or MSNs). If the subscriber asked for more, there may be up to ten. One of these MSNs must be entered here, but without your area code. If you enter the wrong number, your phone operator automatically falls back to the first MSN assigned to your ISDN line.

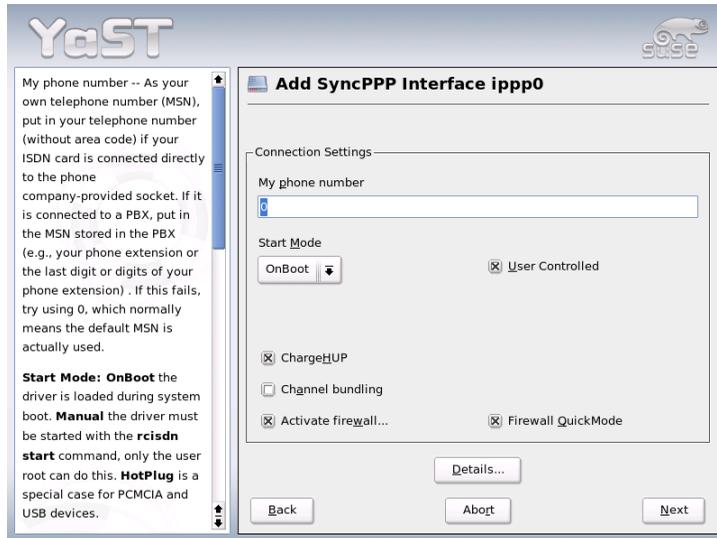


Figure 22.7: ISDN Interface Configuration

ISDN card connected to a phone exchange

Again, the configuration may vary depending on the equipment installed:

1. 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 enabled for direct outward dialing. If this does not work, try a single zero. For further information, consult the documentation that came with your phone exchange.

2. Larger phone exchanges designed for businesses normally use the 1TR6 protocol for internal calls. Their MSN is called EAZ and usually corresponds to the direct-dial number. For the configuration under Linux, it should be sufficient to enter the last digit of the EAZ. As a last resort, try each of the digits from 1 to 9.

For the connection to be terminated just before the next charge unit is due, enable 'ChargeHUP'. However, remember that may not work with every ISP. You can also enable channel bundling (multilink PPP) by selecting the corresponding check box. Finally, you can enable SuSEfirewall2 for your link by selecting 'Activate Firewall'. To enable the normal user without administrator permissions to activate or deactivate the interface, select the 'User Controlled'.

'Details...' opens a dialog in which to implement more complex connection schemes, which is not relevant for normal home users. Leave this dialog by selecting 'Next'.

In the next dialog, make IP address settings. If you have not been given a static IP by your provider, select 'Dynamic IP address'. Otherwise, use the fields provided to enter your host's local IP address and the remote IP address according to the specifications of your ISP. If the interface should be the default route to the Internet, select 'Default Route'. Each host can only have one interface configured as the default route. Leave this dialog by selecting 'Next'.

The following dialog allows you to set your country and to select an ISP. The IPs included in the list are call-by-call providers only. If your ISP is not in the list, select 'New'. This opens the 'Provider Parameters' dialog in which to enter all the details for your ISP. When entering the phone number, 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 finished, select 'Next'.

To use 'Dial on Demand' on a stand-alone workstation, also specify the name server (DNS server). Most ISPs support dynamic DNS, which means the IP address of a name server is sent by the ISP each time you connect. For a single workstation, however, you still need to provide a placeholder address like 192.168.22.99. If your ISP does not support dynamic DNS, specify the name server IP addresses of the ISP. If desired, specify a time-out for the connection — the period of network inactivity (in seconds) after which the connection should be automatically terminated. Confirm your settings with 'Next'. YaST displays a summary of the configured interfaces. To make all these settings active, select 'Finish'.

22.4.6 Hotplug and PCMCIA

Hotplug devices are no longer treated in a special way, because all devices are initialized by hotplug. Nevertheless, physical hotplug is characterized by some special aspects. When the system is booted, built-in devices are always initialized in the same order and receive the same interface names from the kernel. Interface names are assigned dynamically by the kernel. As soon as an interface

is registered, it is assigned the next free name. Because hotplug devices can be inserted at random, they do not always receive the same interface names. Nevertheless, they are assigned the same configurations, because the configurations do not depend on the interface names. If you prefer persistent interface names, enter `PERSISTENT_NAME=<name>` in the respective interface configuration file (`/etc/sysconfig/network/ifcfg-*`). This setting will be adopted the next time the card is initialized (inserted).

22.4.7 Configuring IPv6

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.

22.5 Routing in SUSE LINUX

The routing table is set up 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

22.6 SLP Services in the Network

The *service location protocol* (SLP) was developed with the aim of simplifying the configuration of networked clients within a local network. To configure a network client, including all required services, the administrator traditionally needs detailed knowledge of the servers available in the network. SLP is used to make the availability of a certain service known to all clients in the local network. Applications that support SLP can use the information distributed and be configured automatically.

22.6.1 SLP Support in SUSE LINUX

SUSE LINUX supports installation using installation sources provided 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.

Registering Your Own Services

Many applications under SUSE LINUX already have integrated SLP support through the use of the `libslp` library. If a service has not been compiled with SLP support, use one of the following methods to make it available with SLP:

Static Registration via `/etc/slp.reg.d`

Create a separate registration file for each new service. The following is an example of a file for registering a scanner service:

```
## Register a saned service on this system
## en means english language
## 65535 disables the timeout, so the service registration does
## not need refreshes
service:scanner.sane://$HOSTNAME:6566,en,65535
watch-port-tcp=6566
description=SANE scanner daemon
```

The most important line in this file is the *service URL*, which begins with `service:`. This contains the service type (`scanner.sane`) and the address under which the service is available on the server. `<$HOSTNAME>` is automatically replaced with the full host name. The name of the TCP port on which the relevant service can be found follows, separated by a colon. Then enter the language in which the service should appear and the duration of registration in seconds. These should be separated from the service URL by commas. Set the value for the duration of registration between 0 and 65535. 0 prevents registration. 65535 removes all restrictions.

The registration file also contains the two variables `watch-tcp-port` and `description`. 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 that is displayed in suitable browsers.

Static Registration with `/etc/slp.reg`

The only difference from the procedure described above is the grouping of all services within a central file.

Dynamic Registration with `slptool` If a service should be registered for SLP from proprietary scripts, use the `slptool` command line front-end.

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 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 at `slp: /`. Click the icons in the main window to obtain more detailed information about the relevant service.

If you use Konqueror with `service: /`, click the relevant icon once in the browser window to set up a connection with the selected service.

Activating SLP

Note

Activating `slpd`

`slpd` must run on your system if you want to offer 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. To activate it for the duration of a session, run `rcslpd start` as root to start it and `rcslpd stop` to stop it. Perform a restart or status check with `restart` or `status`. If `slpd` should be active by default, run the `insserv slpd` command once as root. This automatically includes `slpd` in the set of services to start when a system boots.

22.6.2 For More information

The following sources are available for further information about SLP:

RFC 2608, 2609, 2610 RFC 2608 generally deals with the definition of SLP. RFC 2609 deals with the syntax of the service URLs used in greater detail and RFC 2610 deals with DHCP via SLP.

<http://www.openslp.com> The home page 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 consult its supplied *Programmers Guide*.

22.7 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 *Domain Name System* on page 403. The following configuration examples refer to BIND.

22.7.1 Starting the Name Server BIND

On a SUSE LINUX system, the name server BIND (*Berkeley Internet name domain*) comes preconfigured so it can be started right after installation without any problem. If you already have a functioning Internet connection and have entered 127.0.0.1 as the name server address for `localhost` in `/etc/resolv.conf`, you normally already have a working name resolution without needing to know the DNS of the provider. BIND carries out 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 in `/usr/share/doc/packages/bind/sample-config`.

Note**Automatic Adaptation of the Name Server Information**

Depending on the type of Internet connection or the network connection, the name server information can automatically be adapted to the current conditions. To do this, set the variable `MODIFY_NAMED_CONF_DYNAMICALLY` in the file `/etc/sysconfig/network/config` to `yes`.

Note

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 `rndc 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 `rndc 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 Example 22.10 are just examples. Adjust these entries according to your own setup.

Example 22.10: 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 `rndc reload`. Achieve the same by stopping and restarting the name server with `rndc restart`. Stop the server at any time by entering `rndc stop`.

22.7.2 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.

`/etc/named.conf` is roughly divided into two areas. One is the `options` section for general settings and the other consists of zone entries for the individual domains. A `logging` section and `acl` (access control list) entries are optional. Comment lines begin with a `"#" sign or "/". A minimal /etc/named.conf is shown in Example 22.11.`

Example 22.11: 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

22.7.3 Important Configuration Options

directory "*filename*"; Specifies the directory in which BIND can find the files containing the zone data. Usually, this is `/var/lib/named`.

forwarders { *ip-address*; }; Specifies the name servers (mostly of the provider) to which DNS requests should be forwarded if they cannot be resolved directly. Replace *ip-address* with an IP address like `10.0.0.1`.

forward first; Causes DNS requests to be forwarded before an attempt is made to resolve them via the root name servers. Instead of `forward first`, `forward only` can be written to have all requests forwarded and none sent to the root name servers. This makes sense for firewall configurations.

listen-on port 53 { 127.0.0.1; *ip-address*; };
Tells BIND on which network interfaces and port to accept client queries. `port 53` does not need to be specified explicitly, because 53 is the default port. Enter `127.0.0.1` to permit requests from the local host. If you omit this entry entirely, all interfaces are used by default.

listen-on-v6 port 53 {any;}; Tells BIND on which port it should listen for IPv6 client requests. The only alternative to `any` is `none`. As far as IPv6 is concerned, the server only accepts a wild card address.

query-source address * port 53; This entry is necessary if a firewall is blocking outgoing DNS requests. This tells BIND to post requests externally from port 53 and not from any of the high ports above 1024.

query-source-v6 address * port 53; Tells BIND which port to use for IPv6 queries.

allow-query { 127.0.0.1; <net>; } Defines the networks from which clients can post DNS requests. Replace <net> with address information like 192.168.1/24. The /24 at the end is an abbreviated expression for the netmask, in this case, 255.255.255.0.

allow-transfer !*;; controls which hosts can request zone transfers. In the example, such requests are completely denied with ! *. Without this entry, zone transfers can be requested from anywhere without restrictions.

statistics-interval 0; In the absence of this entry, BIND generates several lines of statistical information per hour in `/var/log/messages`. 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 sixty 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 sixty minutes.

notify no; no prevents other name servers from being informed when changes are made to the zone data or when the name server is restarted.

22.7.4 The Configuration Section Logging

What, how, and where logging takes place can be extensively configured in BIND. Normally, the default settings should be sufficient. Example 22.12 shows the simplest form of such an entry and completely suppresses any logging.

Example 22.12: Entry to Disable Logging

```
logging {  
    category default { null; };  
};
```

22.7.5 Zone Entry Structure

Example 22.13: Zone Entry for my-domain.de

```
zone "my-domain.de" in {
    type master;
    file "my-domain.zone";
    notify no;
};
```

After `zone`, specify the name of the domain to administer, `my-domain.de`, followed by `in` and a block of relevant options enclosed in curly braces, as shown in Example 22.13. To define a *slave zone*, switch the `type` to `slave` and specify a name server that administers this zone as `master` (which, in turn, may be a slave of another master), as shown in Example 22.14.

Example 22.14: 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. It must be used together with `masters`.

type hint; The zone `.` of the `hint` type is used to set the root name servers. This zone definition can be left as is.

file my-domain.zone or file "slave/other-domain.zone";

This entry specifies the file where zone data for the domain is located. This file is not required for a slave, as this data is fetched from another name server. To differentiate master and slave files, use the directory `slave` for the slave files.

masters { <server-ip-address>; } This entry is only needed for slave zones. It specifies from which name server the zone file should be transferred.

allow-update {! *}; This option controls external write access, which would allow clients to make a DNS entry — something not normally desirable for security reasons. Without this entry, zone updates are not allowed at all. The above entry achieves the same because `! *` effectively bans any such activity.

22.7.6 Structure of Zone Files

Two types of zone files are needed. One assigns 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. Complete host names specified with a full domain name must end with a `.` to avoid having the domain added to it again. A missing or wrongly placed dot is probably the most frequent cause of name server configuration errors.

Note

The first case to consider is the zone file `world.zone`, responsible for the domain `world.cosmos`, shown in Example 22.15.

Example 22.15: File `/var/lib/named/world.zone`

```
1 $TTL 2D
2 world.cosmos. IN SOA      gateway root.world.cosmos. (
3                   2003072441 ; serial
4                   1D         ; refresh
5                   2H         ; retry
6                   1W         ; expiry
```

```
7          2D )          ; minimum
8
9          IN NS          gateway
10         IN MX          10 sun
11
12  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, `@` can be entered here, in which case the zone would be extracted from the corresponding entry in `/etc/named.conf`.
- After `IN` SOA is the name of the name server in charge as master for this zone. The name is expanded from `gateway` to `gateway.world.cosmos`, because it does not end with a `.`.
- An e-mail address of the person in charge of this name server follows. Because the `@` sign already has a special meaning, `.` is entered here instead. For `root@world.cosmos` the entry must read `root.world.cosmos.` The `.` must be included at the end to prevent the zone from being added.
- The `(` includes all lines up to `)` into the SOA record.

Line 3: The `serial` number is an arbitrary number that is increased each time this file is changed. It is needed to inform the secondary name servers (slave servers) of changes. For this, a 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 are informed of the changes made to the zone data.
- Line 10:** The MX record specifies the mail server that accepts, processes, and forwards e-mails for the domain `world.cosmos`. In this example, this is the host `sun.world.cosmos`. The number in front of the 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 is made with the next higher value.
- Lines 12–17:** These are the actual address records where one or more IP addresses are assigned to 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 `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` means *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 Example 22.16 on the next page.

Example 22.16: Reverse Lookup

```
1
2 $TTL 2D
3 1.168.192.in-addr.arpa. IN SOA gateway.world.cosmos. root.world.cosmos. (
4     2003072441      ; serial
5     1D              ; refresh
6     2H              ; retry
7     1W              ; expiry
8     2D )           ; minimum
9
10                    IN NS      gateway.world.cosmos.
11
12 1                  IN PTR    gateway.world.cosmos.
13 2                  IN PTR    earth.world.cosmos.
14 3                  IN PTR    mars.world.cosmos.
```

Line 1: \$TTL defines the standard TTL that applies to all entries here.

Line 2: The configuration file should activate reverse lookup for the network 192.168.1.0. Given that the zone is called 1.168.192.in-addr.arpa, should not be added to the 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.

22.7.7 Secure Transactions

Secure transactions can be made with the help of transaction signatures (TSIGs) based on shared secret keys (also called TSIG keys). This section describes how to generate and use such keys.

Secure transactions are needed for 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.

Generate a TSIG key with the following command (for details, see `man dnssec-keygen`):

```
dnssec-keygen -a hmac-md5 -b 128 -n HOST host1-host2
```

This creates two files with names similar to these:

```
Khost1-host2.+157+34265.private Khost1-host2.+157+34265.key
```

The key itself (a string like `ejIkuCyyGJwwuN3xAteKgg==`) is found in both files. To use it for transactions, the second file (`Khost1-host2.+157+34265.key`) must be transferred to the remote host, preferably in a secure way (using SCP, for 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 `{}/protect{}/path{/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 this example), the server's `/etc/named.conf` must include the following rule:

```
server 192.168.2.3 {
    keys { host1-host2. ; };
};
```

Analogous entries must be included in the configuration files of `host2`.

In addition to any ACLs (Access Control Lists — not to be confused with filesystem 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`.

22.7.8 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 *Secure Transactions* on page 447.

22.7.9 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`.

22.7.10 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 that 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, see the dialog shown in Figure 22.8 on the next page. It allows you to decide whether the PPP daemon should provide a list of forwarders on dial-up via DSL or ISDN ('PPP Daemon Sets Forwarders') or whether you want to supply your own list ('Set Forwarders Manually').

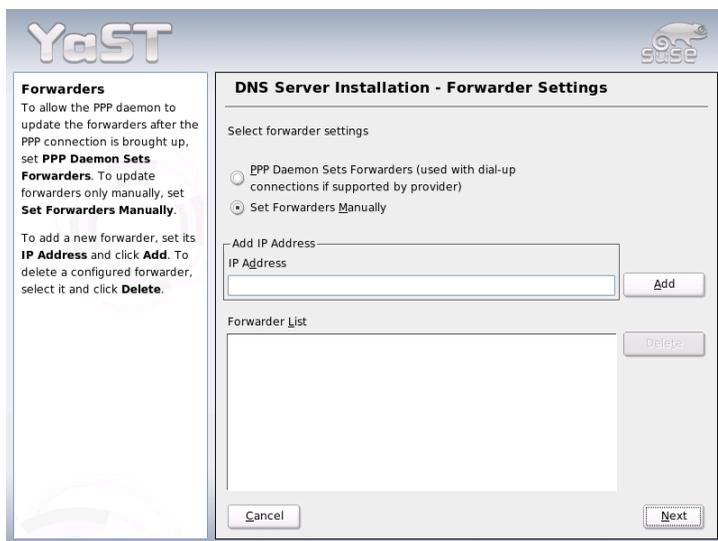


Figure 22.8: DNS Server Installation: Forwarder Settings

DNS Server Installation: DNS Zones The individual entries shown are explained in the discussion of the expert configuration (see *DNS Server: DNS Zones* on page 453).

DNS Server Installation: Finish Wizard

In the final dialog, you can open the DNS port (port 53) in the firewall that is activated during the installation and decide whether DNS should be started. The expert configuration can also be accessed from this dialog. See Figure 22.9 on the next page).

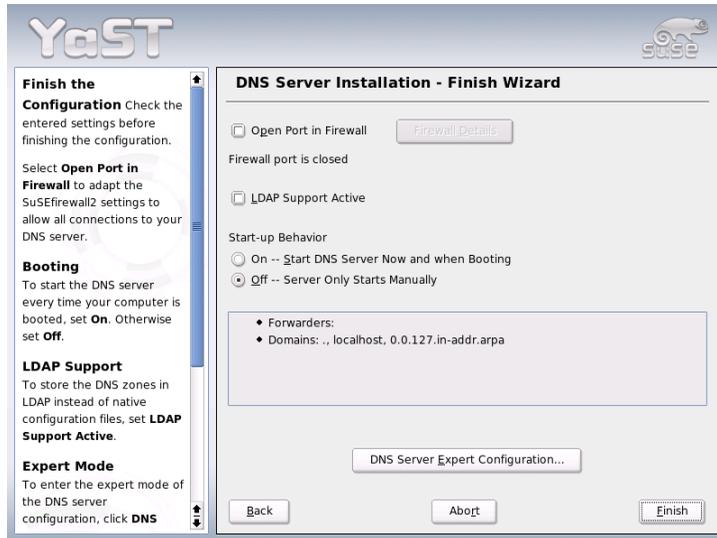


Figure 22.9: DNS Server Installation: Finish Wizard

Expert Configuration

After starting the module, YaST opens a window displaying several configuration options. Completing it results in a DNS server configuration with the basic functions in place:

DNS Server: Start-up Under 'Booting', define whether the DNS server should be 'On' or 'Off' by default. To start the DNS server right away, select 'Start DNS Server Now'. To stop the DNS server, select 'Stop DNS Now'. To save the current settings, select 'Save Settings and Restart DNS Server Now'.

You can open the DNS port in the firewall with 'Open Port in Firewall' and modify the firewall settings with 'Firewall Details'.

DNS Server: Forwarders This is the same dialog as the one opened after starting the wizard configuration (see *DNS Server Installation: Forwarder Settings* on page 450).

DNS Server: Logging This section allows you to set what the DNS server should log and how. Under 'Log Type', specify where the DNS server should write the log data. Use the system-wide log file `/var/log/` messages by selecting 'Log to System Log' or specify a different file by selecting 'Log to File'. In the latter case, additionally specify the maximum file size in megabytes and the number of log files to store.

Further options are available under 'Additional Logging': Enabling 'Log Named Queries' causes *every* query to be logged, in which case the log file could grow extremely large. For this reason, it is not a good idea to enable this option for other than debugging purposes. To log the data traffic during zone updates between DHCP and DNS server, enable 'Log Zone Updates'. To log the data traffic during a zone transfer from master to slave, enable 'Log Zone Transfer'. See Figure 22.10.

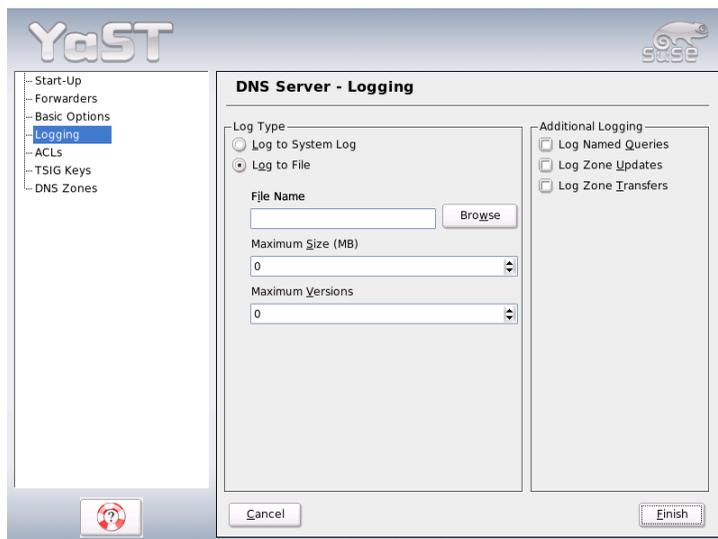


Figure 22.10: DNS Server: Logging

DNS Server: DNS Zones This dialog consists of several parts and is responsible for the management of zone files (see *Structure of Zone Files* on page 444). For a new zone, provide a name for it in 'Zone Name'. To add a reverse zone, the name must end in `.in-addr.arpa`. Finally, select the 'Zone Type' (master or slave). See Figure 22.11. Click 'Edit Zone ...' to configure other settings of an existing zone. To remove a zone, click 'Delete Zone'.

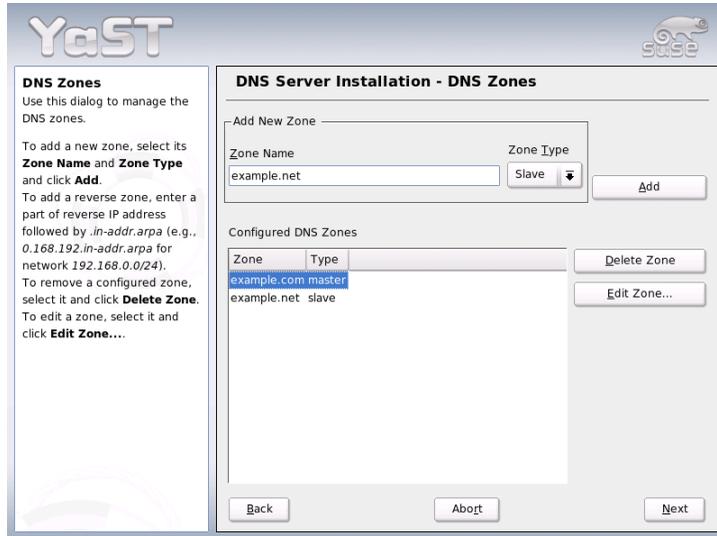


Figure 22.11: DNS Server: DNS Zones

DNS Server: Slave Zone Editor This dialog opens if you select the zone type ‘Slave’ in the step described in *DNS Server: DNS Zones* on page 453. 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 22.12.

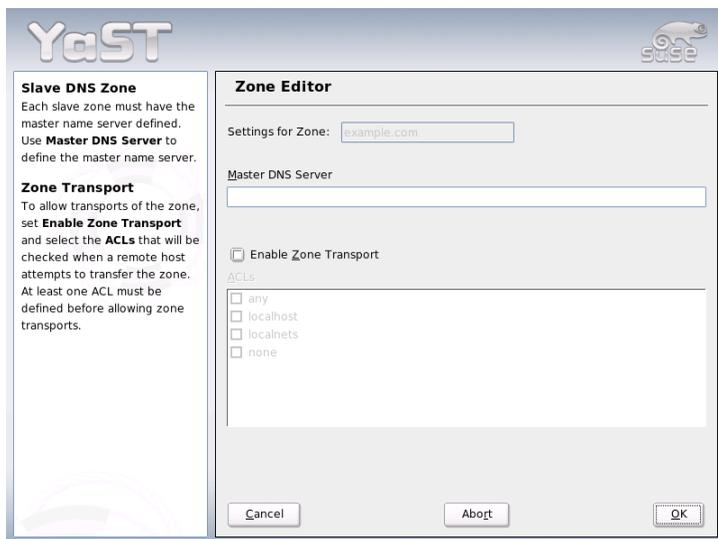


Figure 22.12: DNS Server: Slave Zone Editor

DNS Server: Master Zone Editor This dialog is opened if you selected the zone type ‘Master’ in the step described in *DNS Server: DNS Zones* on page 453. The dialog comprises several pages: Basic (the one opened first), NS Records, MX Records, SOA, and Records. For each of these pages, find a description in the following paragraphs.

The dialog shown in Figure 22.13 lets you define settings for dynamic DNS and access options for zone transfers to clients and slave name servers. To permit the dynamic update of zones, select 'Allow Dynamic Updates' as well as the corresponding TSIG key. The key must have been defined before the update action starts.

To enable zone transfers, select the corresponding ACLs. ACLs must have been defined already.

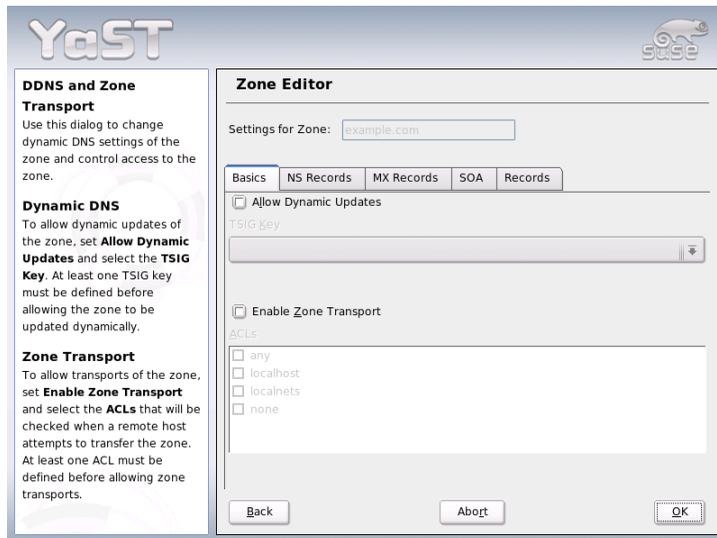


Figure 22.13: DNS Server: Zone Editor (Basic)

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' then confirm with 'Add'. See Figure 22.14 on the facing page.

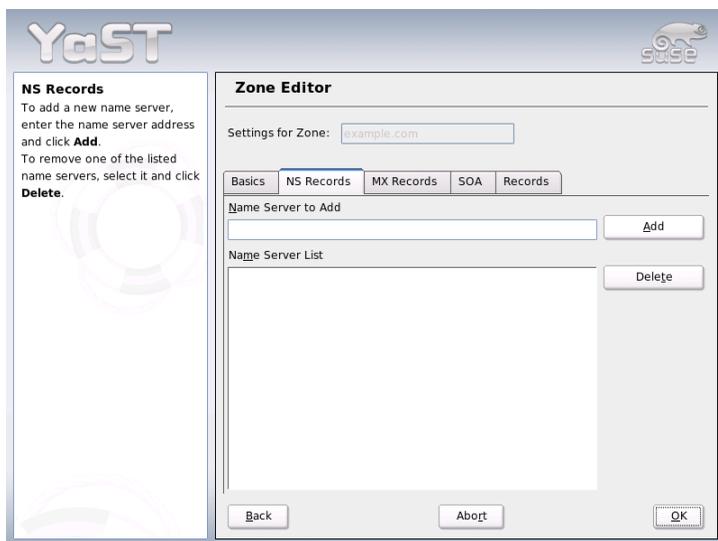


Figure 22.14: DNS Server: Zone Editor (NS Records)

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 22.15 on the next page.

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 22.15 on page 444. Please note that the changing SOA records is not supported for dynamic zones managed via LDAP.

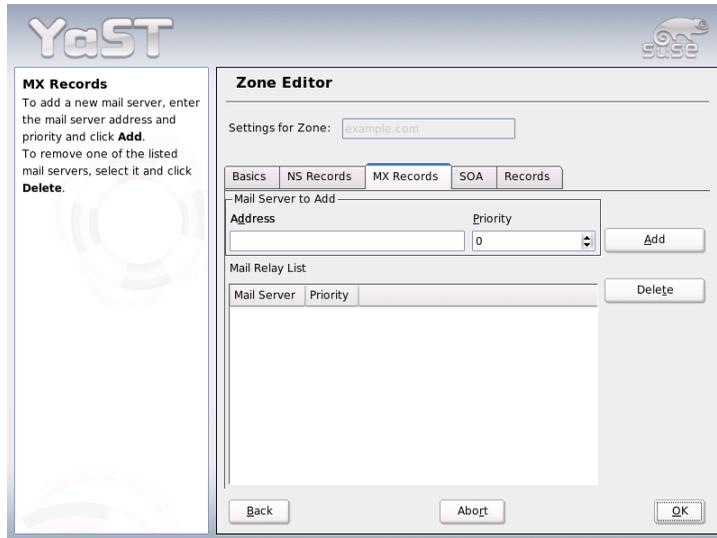


Figure 22.15: DNS Server: Zone Editor (MX Records)

DNS Server: Zone Editor (Records) This dialog lets you manage a list of IP addresses and the corresponding names as assigned to them. In 'Name', enter the host name then select its type. 'A-Record' represents the main entry. 'CNAME' is an alias. Under 'MX-Relay' the entry (name) is replaced with its value.

22.7.11 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.

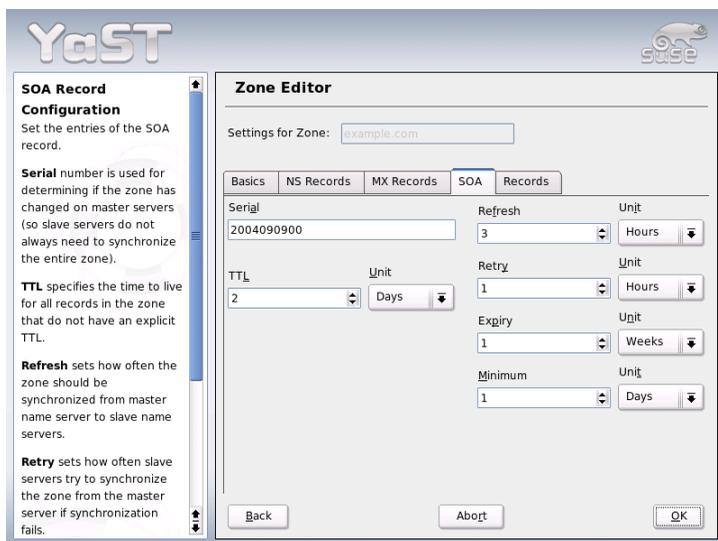


Figure 22.16: DNS Server: Zone Editor (SOA)

22.8 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 always finds 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 *NFS — Shared File Systems* on page 486.

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 as *YP*, which simply stands for the idea of the network's "yellow pages."

22.8.1 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 22.17. With the check box, define whether the host should also be a NIS client, enabling users to log in and access data from the NIS server.

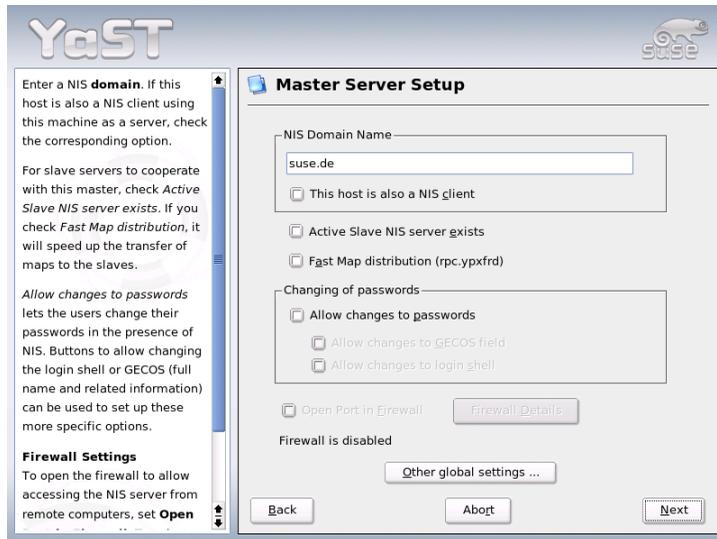


Figure 22.17: YaST: NIS Server Configuration Tool

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.

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 22.18, 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’.

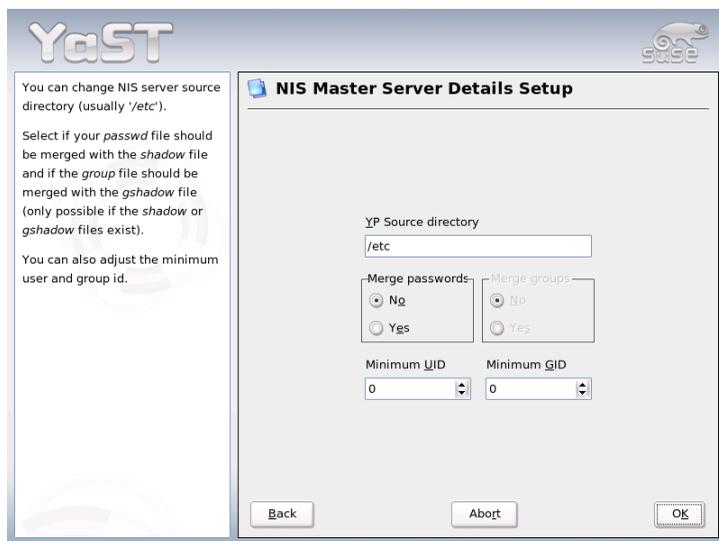


Figure 22.18: YaST: Changing the Directory and Synchronizing Files for a NIS Server

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.

'Next' continues to the last dialog, shown in Figure 22.19. 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.

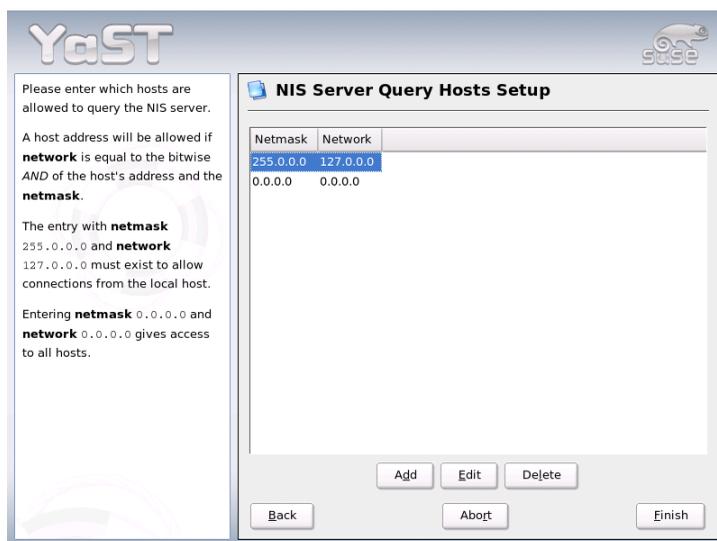


Figure 22.19: YaST: Setting Request Permissions for a NIS Server

Note

Automatic Firewall Configuration

If a firewall is active on your system (SuSEfirewall2), YaST adapts its configuration for the NIS server by enabling the `portmap` service when 'Open Ports in Firewall' is selected.

Note

22.8.2 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 *DHCP* on page 491. If a static IP address is used, specify the NIS domain and the NIS server manually (see Figure 22.20). ‘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`.

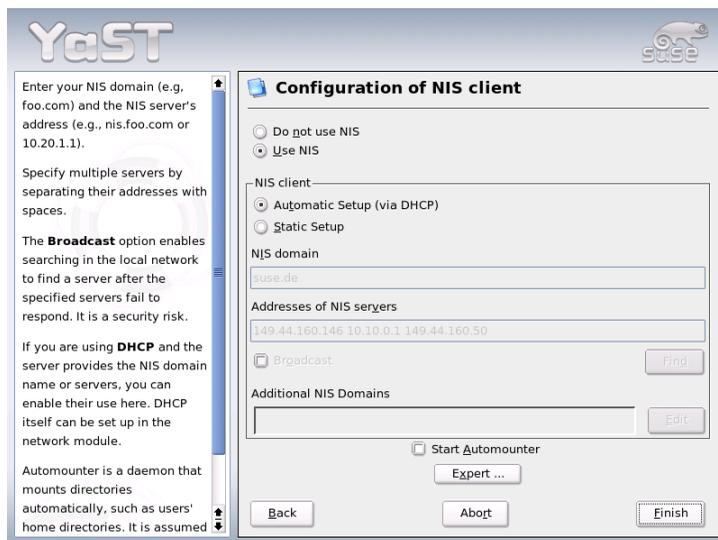


Figure 22.20: Setting Domain and Address of NIS Server

22.9 LDAP — A Directory Service

It is crucial within a networked environment to keep important information structured and quickly available. This can be done with a directory service that, like the common yellow pages, keeps information available in a well-structured, quickly searchable form.

In the ideal case, a central server keeps the data in a directory and distributes it to all clients using a certain protocol. The data is structured in a way that allows a wide range of applications to access it. That way, it is not necessary for every single calendar tool and e-mail client to keep its own database — a central repository can be accessed instead. This notably reduces the administration effort for the information. The use of an open and standardized protocol like LDAP (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, write access is limited to a small number of updates by the administrator. Conventional databases are optimized for accepting the largest possible data volume in a short time.
- Because write accesses can only be executed in a restricted fashion, a directory service is 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 example, the figures administered in accounting.
- When static data is administered, updates of the existing data sets are very rare. When working with dynamic data, especially when data sets like bank accounts or accounting are concerned, the consistency of the data is of primary importance. If an amount should be subtracted from one place to be added to another, both operations must happen concurrently, within a *transaction*, to ensure 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 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 a trimmed down version of the DAP. Without losing the X.500 entry hierarchy, profit from LDAP's cross-platform capabilities and save resources. 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`

22.9.1 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, unlike NIS. The clearly-defined hierarchical structure of the data eases the administration of large amounts of data, because it can be searched better.

22.9.2 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 22.21 on the next page.

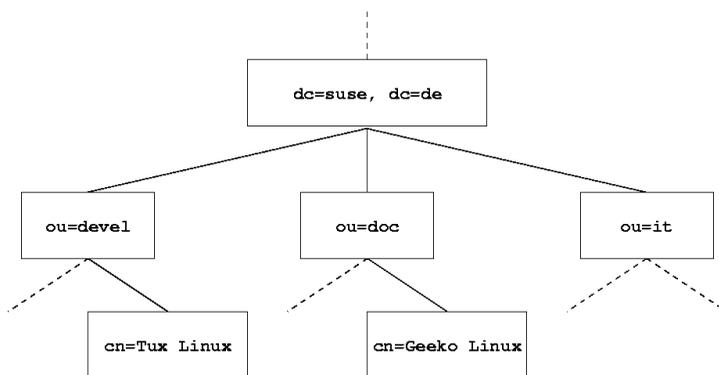


Figure 22.21: Structure of an LDAP Directory

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`.

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 22.10 on the following page 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 22.10: Commonly Used Object Classes and Attributes

Object Class	Meaning	Example Entry	Compulsory Attributes
dcObject	<i>domainComponent</i> (name components of the domain)	suse	dc
organizationalUnit	<i>organizationalUnit</i> (organizational unit)	doc	ou
inetOrgPerson	<i>inetOrgPerson</i> (person-related data for the intranet or Internet)	Geeko Linux	sn and cn

Example 22.17 shows an excerpt from a scheme directive with explanations.

Example 22.17: 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 permitted in conjunction with this object class.

A very good introduction to the use of schemes can be found in the documentation of OpenLDAP. When installed, find it in `/usr/share/doc/packages/openldap2/admin-guide/index.html`.

22.9.3 Server Configuration with `slapd.conf`

Your installed system contains a complete configuration file for your LDAP server at `/etc/openldap/slapd.conf`. The single entries are briefly described here and necessary adjustments are explained. Entries prefixed with a hash (`#`) are inactive. This comment character must be removed to activate them.

Global Directives in `slapd.conf`

Example 22.18: `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 22.18, 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 22.19: *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 22.20: *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 22.20 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 to this directory. Access control regulation in LDAP is a highly complex process. The following tips can help:

- Every access rule has the following structure:

```
access to <what> by <who> <access>
```

- *<what>* is a placeholder for the object or attribute to which access is granted. Individual directory branches can be protected explicitly with separate rules. It is also possible to process regions of the directory tree with one rule by using regular expressions. `slapd` evaluates all rules in the order in which they are listed in the configuration file. More general rules should be listed after more specific ones — the first rule `slapd` regards as valid is evaluated and all following entries are ignored.

- *<who>* determines who should be granted access to the areas determined with *<what>*. Regular expressions may be used. `slapd` again aborts the evaluation of *who* after the first match, so more specific rules should be listed before the more general ones. The entries shown in Table 22.11 are possible.

Table 22.11: 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
<code>dn.regex=<regex></code>	all users matching the regular expression

- *<access>* specifies the type of access. Use the options listed in Table 22.12.

Table 22.12: 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 22.21 on the following page shows a simple example for a simple access control that can be arbitrarily developed using regular expressions.

Example 22.21: 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 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 22.22: 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 22.22 on the preceding page). 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 of 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 in *Data Handling in the LDAP Directory* on this page, 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 *The YaST Runlevel Editor* on page 241, can be used to have the server started and stopped automatically on boot and halt of the system. It is also possible to create the corresponding links to the start and stop scripts with the `insserv` command from a command prompt as described in *Adding init Scripts* on page 240.

22.9.4 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/ldapd.conf` is correct and ready to go (it features appropriate entries for `suffix`, `directory`, `rootdn`, `rootpw`, and `index`), proceed to entering records. OpenLDAP offers the `ldapadd` command for this task.

If possible, add the objects to the database in bundles for practical reasons. LDAP is able to process the LDIF format (LDAP data interchange format) for this. An LDIF file is a simple text file that can contain an arbitrary number of 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 22.21 on page 467 would look like that in Example 22.23.

Example 22.23: 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 must be encoded correctly. Use an editor that supports UTF-8 (such as *Kate* or recent versions of *Emacs*). Otherwise, avoid umlauts and other special characters or use `recode` to recode the input to UTF-8.

Note

Save the file with the `.ldif` suffix then pass it to the server with the following command:

```
ldapadd -x -D <dn of the administrator> -W -f <file>.ldif
```

`-x` switches off the authentication with SASL in this case. `-D` declares the user that calls the operation. The valid DN of the administrator is entered here just like it has been configured in `slapd.conf`. In the current example, this is `cn=admin,dc=suse,dc=de`. `-W` circumvents entering the password on the command line (in clear text) and activates a separate password prompt. This password was previously determined in `slapd.conf` with `rootpw`. `-f` passes the file name. See the details of running `ldapadd` in Example 22.24.

Example 22.24: 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 individuals can be prepared in separate LDIF files. Example 22.25 adds Tux to the new LDAP directory.

Example 22.25: 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 22.26.

Example 22.26: 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:

1. Start `ldapmodify` and enter your password:

```
ldapmodify -x -D cn=admin,dc=suse,dc=de -W
Enter LDAP password:
```

2. Enter the changes while carefully complying with the syntax in the order presented below:

```
dn: cn=Tux Linux,ou=devel,dc=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`. `-x` requests activation of simple authentication. `(objectClass=*)` declares that all objects contained in the directory should be read. This command option can be used after the creation of a new directory tree to verify that all entries have been recorded correctly and the server responds as desired. 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
```

22.9.5 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 automatically enables any PAM and NSS related changes as required by LDAP (described below) and installs 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 22.27).

Example 22.27: 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`.

`glibc` name resolution through the `nsswitch` mechanism is adapted to the employment of LDAP with `nss_ldap`. A new, adapted file `nsswitch.conf` is created in `/etc/` with the installation of this package. More about the workings of `nsswitch.conf` can be found in *Configuration Files* on page 416. The following lines must be present in `nsswitch.conf` for user administration and authentication with LDAP (See Example 22.28):

Example 22.28: Adaptations in nsswitch.conf

```
passwd: compat
group:  compat

passwd_compat: ldap
group_compat:  ldap
```

These lines order the resolver library of `glibc` first to evaluate the corresponding files in `/etc` and additionally access the LDAP server as sources for authentication and user data. Test this mechanism, for example, by reading the content of the user database with the command `getent passwd`. The returned set should contain a survey of the local users of your system as well as all users stored on the LDAP server.

To prevent regular users managed through LDAP from logging in to the server with `ssh` or `login`, the files `/etc/passwd` and `/etc/group` each need to include an additional line. This is the line `+:::/:sbin/nologin` in `/etc/passwd` and `+::: in /etc/group`.

Configuration of the LDAP Client

After `nss_ldap`, `pam_ldap`, `/etc/passwd`, and `/etc/group` have been modified by YaST in the required way, the actual configuration work can begin on the first YaST dialog. See Figure 22.22.

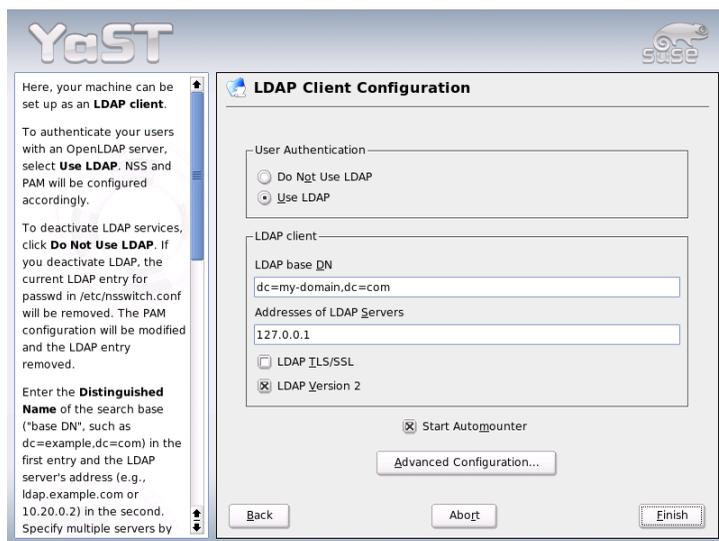


Figure 22.22: YaST: Configuration of the LDAP Client

Activate the use of LDAP for user authentication 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'. To mount directories on remote hosts automatically, select 'Start Automounter'. To modify data on the server as administrator, click 'Advanced Configuration'. See Figure 22.23.

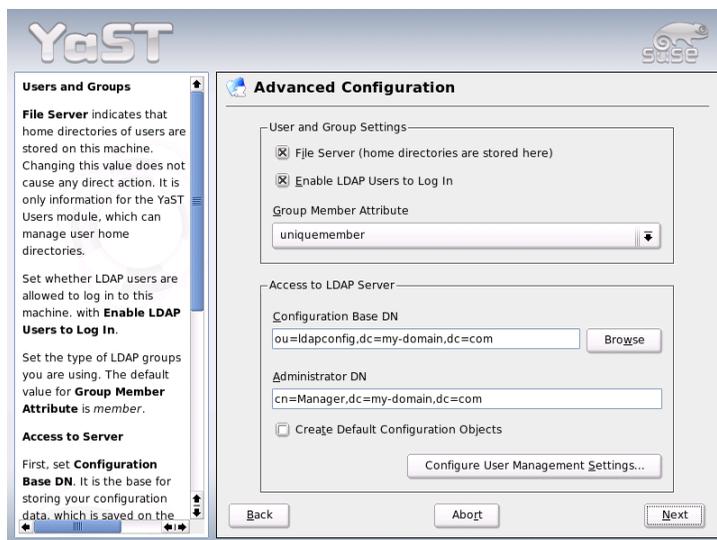


Figure 22.23: YaST: Advanced Configuration

The next dialog has two parts: In the upper area, set general options for users and groups, as reflected by the YaST user module. In the lower area, 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 this ensures 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 With this, specify the type of LDAP group to use, 'member' (default setting) or 'uniquemember'.

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 'Administrator DN'.

Click 'Configure User Management Settings' to edit entries on the LDAP server. In the dialog 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

Using 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

The dialog for module configuration (Figure 22.24 on the next page) allows selection and modification of 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 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 opens in which to change all settings belonging to the attribute. Accept the changes with 'OK'.

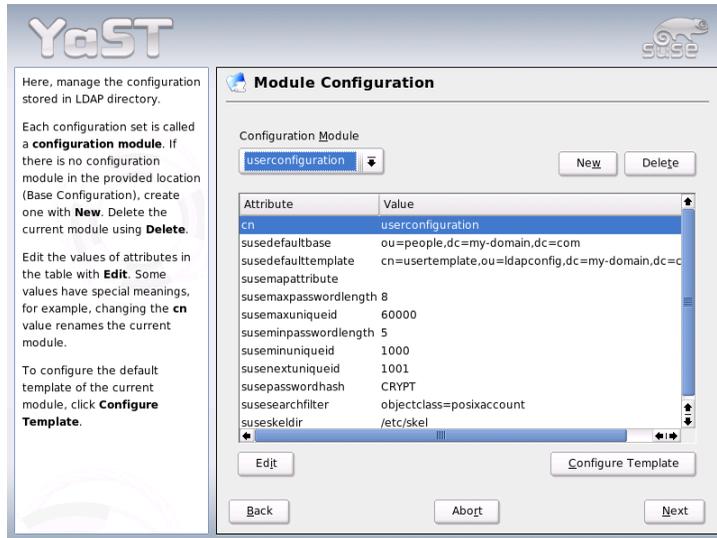


Figure 22.24: YaST: Module Configuration

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. Clicking 'Delete' deletes the currently selected module.

The YaST modules for group and user administration embed templates 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 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 22.25 on the facing page). 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 needs or leave some of them empty. Empty attributes are deleted on the LDAP server.

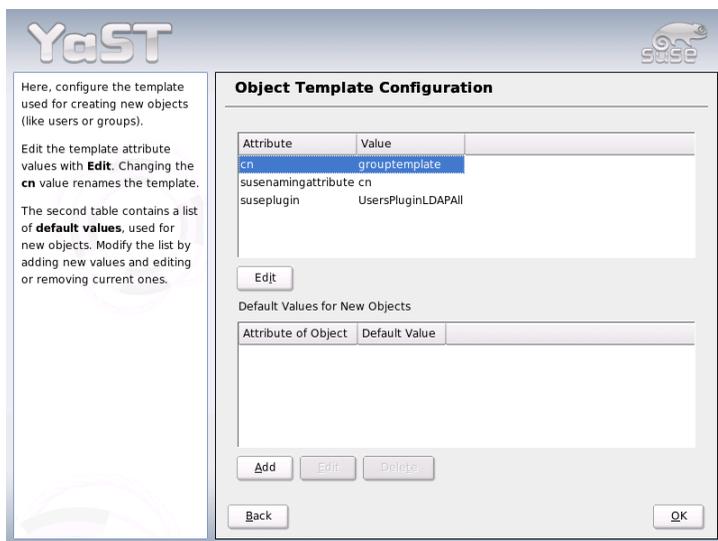


Figure 22.25: YaST: Configuration of an Object Template

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 standard values can be added, existing attribute and 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 were defined with the procedure described in *Configuration of the LDAP Client* on page 479. When LDAP is used, this form leads to another form for the registration of LDAP-specific attributes. It is shown in Figure 22.26. Select all attributes for which to change the value then click 'Edit'. Closing the form that opens with 'Continue' returns to the initial input form for user administration.

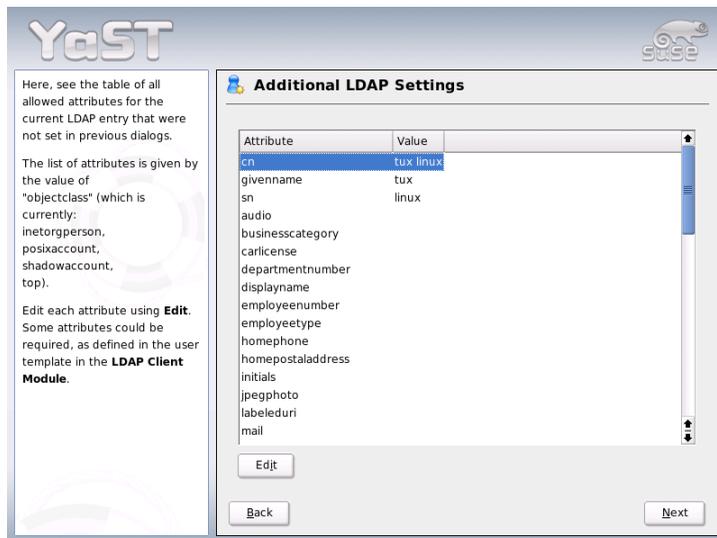


Figure 22.26: YaST: Additional LDAP Settings

The initial input form of user administration offers 'LDAP Options'. This gives the possibility to apply LDAP search filters to the set of available users or to go to the module for the configuration of LDAP users and groups by selecting 'LDAP User and Group Configuration'.

22.9.6 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>.

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/admin22/> or on an installed system in `/usr/share/doc/packages/openldap2/admin-guide/index.html`

The following redbooks from IBM regard the subject of LDAP:

Understanding LDAP A detailed general introduction to the basic principles of LDAP: <http://www.redbooks.ibm.com/redbooks/pdfs/sg2444986.pdf>.

LDAP Implementation Cookbook The target audience consists of administrators of *IBM SecureWay Directory*. However, important general information about LDAP is also 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.

22.10 NFS — Shared File Systems

As mentioned in *NIS — Network Information Service* on page 459, 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.

22.10.1 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 22.27 on the facing page).

22.10.2 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 `rpcportmap 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, use the following command:

```
mount sun:/home /home
```

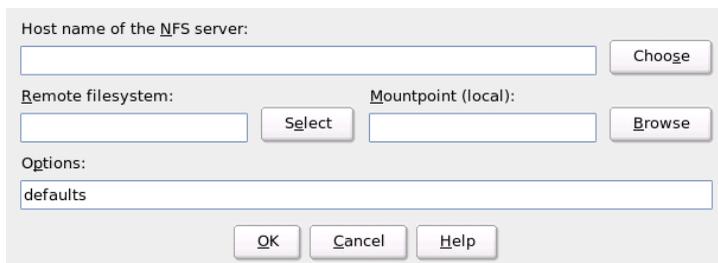


Figure 22.27: NFS Client Configuration with YaST

22.10.3 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 22.28 on the next 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 22.29 on page 489. 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.

Note

Automatic Firewall Configuration

If a firewall is active on your system (SuSEfirewall2), YaST adapts its configuration for the NFS server by enabling the `nfs` service when ‘Open Ports in Firewall’ is selected.

Note

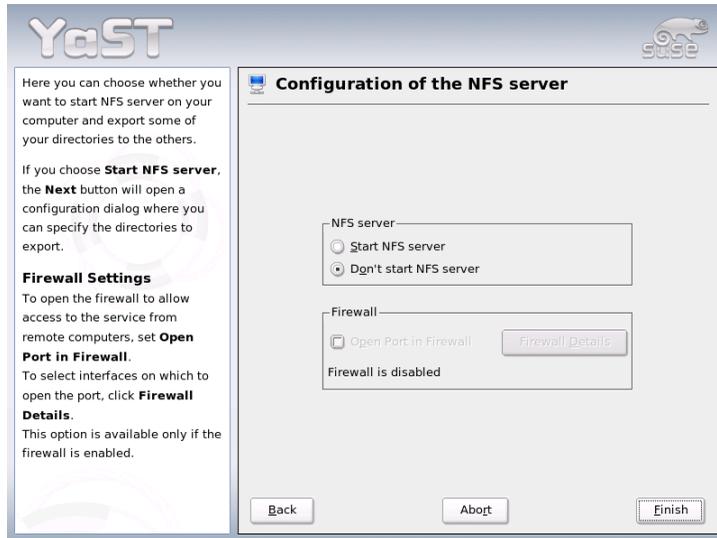


Figure 22.28: NFS Server Configuration Tool

22.10.4 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)

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`.



Figure 22.29: Configuring an NFS Server with YaST

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 22.13: 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.

<code>root_squash</code>	This makes sure the user <code>root</code> of the given machine does not have <code>root</code> permissions on this file system. This is achieved by assigning user ID 65534 to users with user ID 0 (<code>root</code>). This user ID should be set to <code>nobody</code> (which is the default).
<code>no_root_squash</code>	Does not assign user ID 0 to user ID 65534, keeping the <code>root</code> permissions valid.
<code>link_relative</code>	Converts absolute links (those beginning with <code>/</code>) to a sequence of <code>././</code> . This is only useful if the entire file system of a machine is mounted (default).
<code>link_absolute</code>	Symbolic links remain untouched.
<code>map_identity</code>	User IDs are exactly the same on both client and server (default).
<code>map_daemon</code>	Client and server do not have matching user IDs. This tells <code>nfsd</code> to create a conversion table for user IDs. The ugidd daemon is required for this to work.

Your `exports` file might look like Example 22.29.

Example 22.29: /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`.

22.11 DHCP

22.11.1 The DHCP Protocol

The purpose of the *dynamic host configuration protocol* (DHCP) is to assign network settings centrally from a server rather than configuring them locally on each and every workstation. A 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 tries to assign the same address to the client each time it receives a request from it, even over longer periods. This, of course, only works as long as the network does not have more hosts than addresses.

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 supplies not only the IP address and the netmask, but also the host name, domain name, gateway, and name server addresses for the client to use. 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 shows how to use the DHCP server `dhcpd` in a network to manage its entire configuration from one central point.

22.11.2 DHCP Software Packages

Both a DHCP server and DHCP clients are available for SUSE LINUX. The DHCP server available is `dhcpcd` (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 is launched automatically on each system boot to watch for a DHCP server. It does not need a configuration file to do its job and works out of the box in most standard setups. For more complex situations, use the ISC `dhclient`, which is controlled by means of the configuration file `/etc/dhclient.conf`.

22.11.3 The DHCP Server `dhcpcd`

The core of any DHCP system is the dynamic host configuration protocol daemon. This server *leases* addresses and watches how they are used, according to the settings defined in the configuration file `/etc/dhcpcd.conf`. By changing the parameters and values in this file, a system administrator can influence the program's behavior in numerous ways. Look at the basic sample `/etc/dhcpcd.conf` file in Example 22.30.

Example 22.30: The Configuration File `/etc/dhcpcd.conf`

```
default-lease-time 600;           # 10 minutes
max-lease-time 7200;             # 2 hours

option domain-name "cosmos.all";
option domain-name-servers 192.168.1.1, 192.168.1.2;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option subnet-mask 255.255.255.0;

subnet 192.168.1.0 netmask 255.255.255.0
{
    range 192.168.1.10 192.168.1.20;
    range 192.168.1.100 192.168.1.200;
}
```

This simple configuration file should be sufficient to get the DHCP server to assign IP addresses in the network. Make sure a semicolon is inserted at the end of each line, because otherwise `dhcpcd` 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 *DNS — Domain Name System* on page 438.
- 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 this 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. 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.

22.11.4 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. Addresses assigned explicitly always take priority over dynamic addresses from the pool. 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 Example 22.31, are added to the configuration file of Example 22.30 on page 492, the DHCP daemon always assigns the same set of data to the corresponding host under all circumstances.

Example 22.31: Additions to the Configuration File

```
host earth {
hardware ethernet 00:00:45:12:EE:F4;
fixed-address 192.168.1.21;
}
```

The name of the respective host (`host <hostname>`), here `earth`) 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 with `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.

22.11.5 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 user ID `nobody` and run in a chroot environment (`/var/lib/dhcp`). To make this possible, the configuration file `dhcpd.conf` must be located in `/var/lib/dhcp/etc`. The init script automatically copies the file to this directory when starting.

The server's behavior with regard to this feature can be controlled by means of entries in the file `/etc/sysconfig/dhcpd`. To run `dhcpd` without the chroot environment, set the variable `DHCPD_RUN_CHROOTED` in `/etc/sysconfig/dhcpd` to "no".

To enable `dhcpd` to resolve host names even from within the chroot environment, some other configuration files must be copied as well:

- `/etc/localtime`
- `/etc/host.conf`
- `/etc/hosts`
- `/etc/resolv.conf`

These files are copied to `/var/lib/dhcp/etc/` when starting the init script. 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`.

22.11.6 DHCP Configuration with YaST

The YaST DHCP module allows you to set up your own DHCP server for the local network. The module can work in two different modes:

Initial Configuration (Wizard) 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 in the Expert Module and Displaying Help Texts

All dialogs of the DHCP module have a similar layout. The left part of the dialog window displays a tree view for accessing the individual configuration steps. The selected configuration dialog is displayed to the right. To get help for the current dialog, click the life preserver icon at the bottom left of the window. To close the help window and go back to the tree, click the icon depicting a tree structure.

Note

Initial Configuration (Wizard)

After launching the module for the first time, YaST starts a four-part configuration wizard. You can set up a basic DHCP server by completing this wizard.

Selecting the Network Interface In the first step, YaST looks for the network interfaces available on your system then displays them in a list. From the list, select the interface on which the DHCP server should listen and select 'Open Firewall for Selected Interface' to open the firewall for this interface. See Figure 22.30.

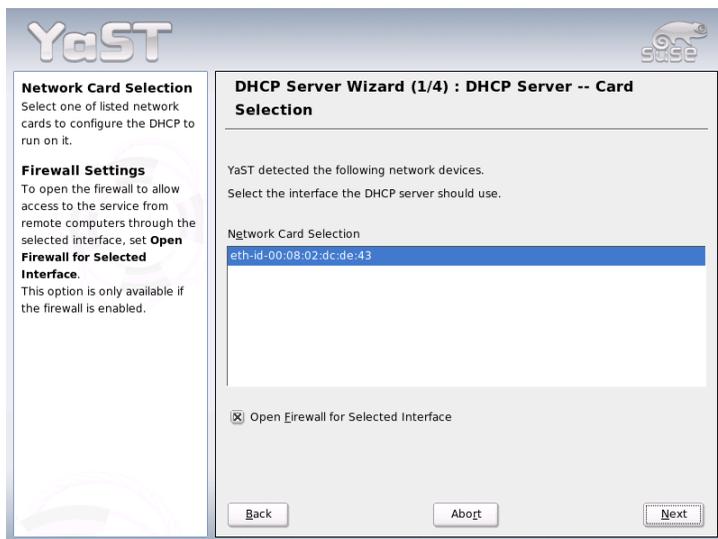


Figure 22.30: DHCP Server: Selecting the Network Interface

Global Settings In the entry fields, provide the network specifics for all clients the DHCP server should manage. These specifics 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, and the lease time. See Figure 22.31 on the next page.

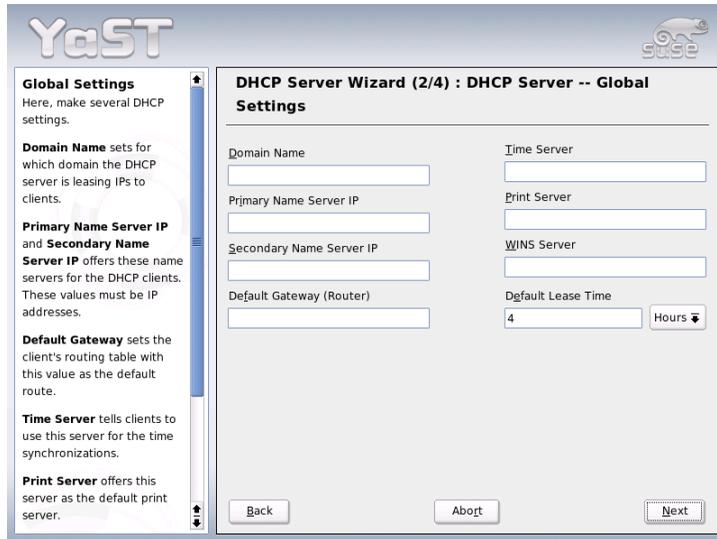


Figure 22.31: DHCP Server: Global Settings

Dynamic DHCP In this step, configure how dynamic IP addresses should be assigned to clients. To do so, specify an IP range from which the server can assign addresses to DHCP clients. All these addresses must be covered by the same netmask. Also specify the lease time during which a client may keep its IP address without needing to request an extension of the lease. Optionally, specify the maximum lease time — the period during which the server reserves an IP address for a particular client (see Figure 22.32 on the facing page).

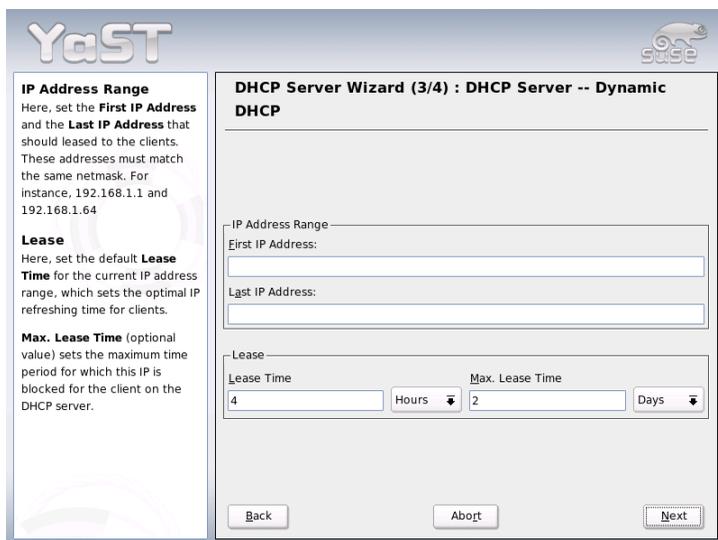


Figure 22.32: DHCP Server: Dynamic DHCP

Finishing the Configuration and Setting the Start Mode

After the third part of the configuration wizard, a last dialog is shown in which to define how the DHCP server should be started. Here, determine whether to start the DHCP server automatically when the system is booted or to start it manually (e.g., for test purposes) when needed. Click 'Finish' to complete the configuration of the server. See Figure 22.33 on the next page.

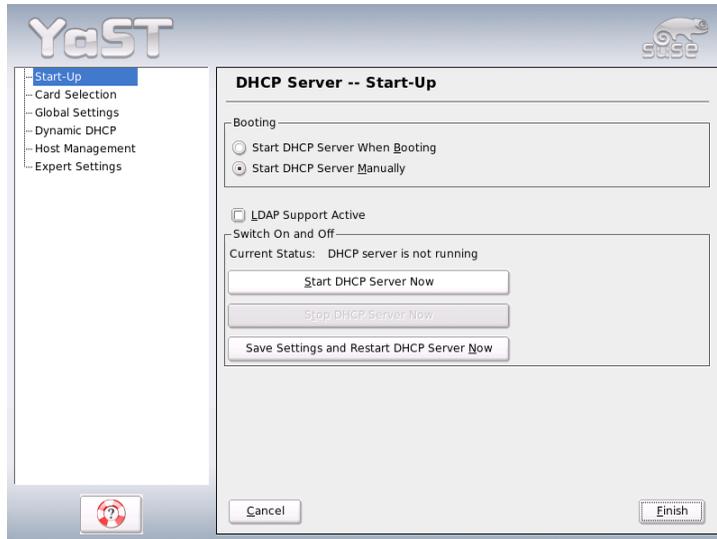


Figure 22.33: DHCP Server: Start-Up

22.11.7 For More Information

More information about DHCP is available at the web site of the *Internet Software Consortium* (<http://www.isc.org/products/DHCP/>).

Information is also available in the manual pages of `dhcpcd`, `dhcpcd.conf`, `dhcpcd.leases`, and `dhcpcd-options`.

22.12 Time Synchronization with xntp

The exact time plays an important role in many processes in a computer system. For this purpose, 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 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. First, `xntp` regularly corrects the local computer clock on the basis of collected correction data. Second, it continuously corrects the local time with the help of time servers in the network. Third, it enables the management of local reference clocks, such as radio-controlled clocks.

22.12.1 Configuration in the Network

`xntp` is preset to use the local computer clock as a time reference. The easiest way to use a time server in the network is to set server parameters. For example, if a time server called `ntp.example.com` is available in the network, add this server to the file `/etc/ntp.conf` in the form `server ntp.example.com`.

To add more 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 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 a time server. 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.

22.12.2 Setting up a Local Reference Clock

The software package `xntp` contains drivers for connecting local reference clocks. A list of supported clocks is available in the `xntp-doc` package in the file `/usr/share/doc/packages/xntp-doc/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 `/usr/share/doc/packages/xntp-doc/html/refclock.htm` provides links to the driver pages describing these parameters. For example, the “Type 8” clock requires an additional mode that specifies the clock more precisely. The Conrad DCF77 receiver module, for example, has mode 5. To make `xntp` accept this clock as a reference, specify the keyword `prefer`. The complete `server` line for a Conrad DCF77 receiver module would be:

```
server 127.127.8.0 mode 5 prefer
```

Other clocks follow the same pattern. Following the installation of the `xntp-doc` package, the documentation for `xntp` is available in the directory `/usr/share/doc/packages/xntp-doc/html`.

22.12.3 NTP Client Configuration with YaST

As well as this manual configuration of `xntp`, SUSE LINUX facilitates the configuration of an NTP client with YaST. Use the easy quick configuration or complex configuration. Both are described in the following.

Quick NTP Client Configuration

The easy NTP client configuration comprises two dialogs. Set the start mode of `xntpd` and the server to query in the first dialog. To start `xntpd` automatically when the system is booted, click 'When Booting System'. Then click 'Select' to access a second dialog in which to select a suitable time server for your network.

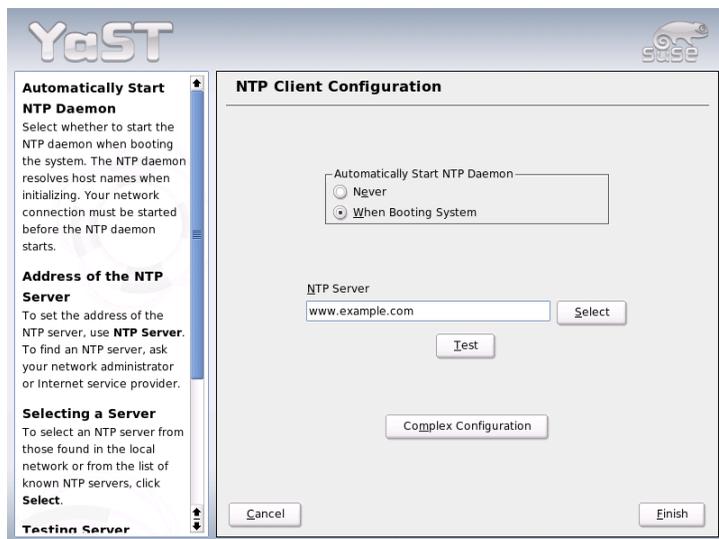


Figure 22.34: YaST: Configuring an NTP Client

In the detailed server selection dialog, determine whether to implement time synchronization using a time server from your local network or an Internet-based time server that takes care of your time zone ('Public NTP Server'). For a local time server, click 'Lookup' to start an SLP query for available time servers in your network. Select the most suitable time server from the list of search results and exit the dialog with 'OK'. For a public time server, select your country (time zone) and a suitable server from the list under 'Public NTP Server' then exit the dialog with 'OK'. In the main dialog, test the availability of the selected server with 'Test' and quit the dialog with 'Finish'.

Complex NTP Client Configuration

The complex configuration of an NTP client can be accessed under ‘Complex Configuration’ from the main dialog of the ‘NTP Client’ module, shown in Figure 22.34 on the preceding page, after selecting the start-up mode as described in the quick configuration.

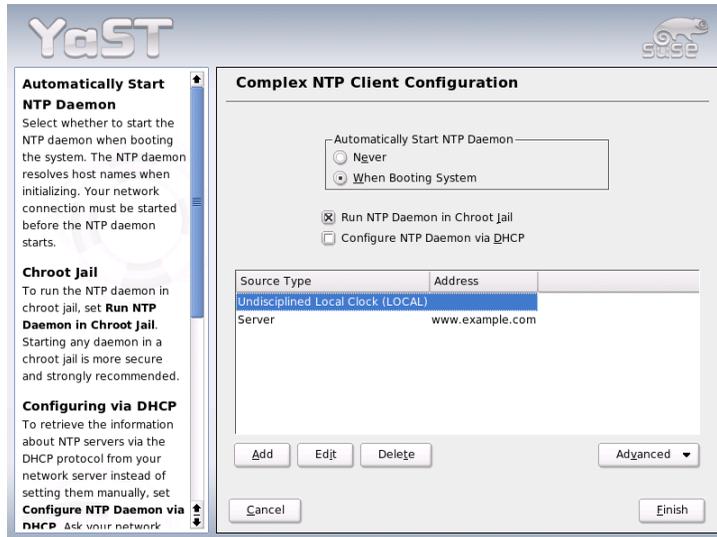


Figure 22.35: YaST: Complex NTP Client Configuration

In ‘Complex NTP Client Configuration’, determine whether `xntpd` should be started in a chroot jail. This increases the security in the event of an attack over `xntpd`, because it prevents the attacker from compromising the entire system. ‘Configure NTP Daemon via DHCP’ sets up the NTP client to get a list of the NTP servers available in your network via DHCP.

The servers and other time sources for the client to query are listed in the lower part. Modify this list as needed with ‘Add’, ‘Edit’, and ‘Delete’. ‘Advanced’ provides the possibility to view the log files of your client or tune the firewall to the NTP client configuration.

Click ‘Add’ to add a new source of time information. In the following dialog, select the type of source with which the time synchronization should be made. The following options are available:

- Server** Another dialog enables you to select an NTP server (as described in *Quick NTP Client Configuration* on page 503). Activate ‘Use for Initial Synchronization’ to trigger the synchronization of the time information between the server and the client when the system is booted. An input field allows you to specify additional options for `xntpd`. Refer to `/usr/share/doc/packages/xntp-doc` for more information.
- Peer** To use a peer in the same network instead of a server, enter the address of the respective system. The rest of the dialog is the same as the dialog for ‘Server’.
- Radio Clock** To use a radio clock in your system for the time synchronization, enter the clock type, unit number, device name, and other options in this dialog. Click ‘Driver Calibration’ to fine-tune the driver. Detailed information about the operation of a local radio clock is available in file: `///usr/share/doc/packages/xntp-doc/html/refclock.htm`.
- Broadcasting** Time information and queries can also be transmitted via broadcast in the network. In this dialog, enter the address to which such broadcasts should be sent. Additional options can be configured as described in `/usr/share/doc/packages/xntp-doc`.
- Accepting Broadcasting Packets** If you want your client to receive its information via broadcast, enter the address from which the respective packets should be accepted in this fields. Further options are described in `/usr/share/doc/packages/xntp-doc`.

The Apache Web Server

This chapter presents the web server Apache. As well as explaining how to perform installation and configuration, it also describes a few modules. Some variations for virtual hosts are also mentioned.

23.1	Basics	508
23.2	Setting up the HTTP Server with YaST	509
23.3	Apache Modules	510
23.4	Threads	511
23.5	Installation	511
23.6	Configuration	513
23.7	Using Apache	518
23.8	Active Contents	518
23.9	Virtual Hosts	524
23.10	Security	527
23.11	Troubleshooting	528
23.12	For More Information	529

23.1 Basics

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*.

23.1.1 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).

23.1.2 HTTP

The clients are usually web browsers, like Konqueror or Mozilla. Communication between the browser and the web server takes place using the hypertext transfer protocol (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>.

23.1.3 URLs

Clients use URLs, such as http://www.suse.com/index_us.html, to request pages from the server. A URL consists of:

Protocol Frequently-used protocols:

- <http://> The http protocol
- <https://> Secure, encrypted version of HTTP
- <ftp://> file transfer protocol for downloading and uploading files

Domain In this example, www.suse.com. The domain can be subdivided into 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).

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 JavaServer Page, or some other resource.

The responsible Internet mechanism, such as the domain name system, DNS, forwards the query to the domain `www.suse.com`, 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://support.novell.com/linux/`.

The file path is relative to the `DocumentRoot`, which can be changed in the configuration file. *DocumentRoot* on page 514 describes how this is done.

23.1.4 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 the server should use, can be configured as described in *DirectoryIndex* on page 515. In this example, `http://www.suse.com` is sufficient to prompt the server to deliver the page `http://www.suse.com/index_us.html`.

23.2 Setting up the HTTP Server with YaST

Apache 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 (‘HTTP Server Configuration’).

In this dialog, first enable the ‘HTTP service’ itself. This also opens the corresponding ports (port 80) in the firewall (‘Open Firewall on Selected Ports’). The lower part of the window (‘Settings/Summary’) allows setting up the local HTTP server: ‘Listen to’ (default is `port 80`), ‘modules’, ‘Default Host’, and ‘Hosts’. The ‘Edit’ allows changing the currently selected setting.

First check the ‘Default Host’ and, if necessary, adjust the configuration to your requirements. Then, activate the desired modules with ‘Modules’. There are additional dialogs for more detailed configuration, especially for the creation of virtual hosts.

23.3 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 accessing such modules. Apart from Perl and PHP, additional scripting languages, such as Python or Ruby, are also available. There are also modules for secure data transmission (secure sockets layer, SSL), user authentication, expanded logging, and many more functions.

With a necessary amount of know-how, Apache can be adapted with custom-written modules to all kinds of requirements and preferences. For information, refer to *Additional Sources* on page 530.

Several “handlers” can be specified for processing queries (by means of directives in the configuration file). These handlers can be part of Apache or a module invoked for processing the query, so this procedure can be arranged in a very flexible way. It is also possible to use custom modules with Apache to influence the way in which requests are processed.

The modularization in Apache has reached an advanced level, where everything except some minor tasks is handled by means of modules. This has progressed so far that even HTTP is processed by way of modules. Accordingly, Apache 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) based on Apache.

Apache modules provide several additional useful features:

Virtual Hosts Support for virtual hosts means that a single instance of Apache and a single machine can be used for several web sites. To 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. This saves 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 JavaScript incompatibility of various browsers can be circumvented by delivering a suitable page version for every browser, provided you are prepared to adapt the JavaScript code for each individual browser.

Flexible Error Handling React flexibly and provide a suitable response in the event of an error, such as in the case of a nonexistent page. The response can even be generated dynamically, for example, with CGI.

23.4 Threads

A thread is a “lighter” form of a process. The advantage of a thread over a process is its lower resource consumption. For this reason, the use of threads instead of processes increases 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 several 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 *Installation* on the current page). The third mode — *perchild* — is not yet fully mature and is therefore not available for installation in SUSE LINUX.

23.5 Installation

23.5.1 Selecting Packages in YaST

For a basic installation, it is sufficient to select the Apache package `apache2`. Additionally, 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.

23.5.2 Activating Apache

After installation, Apache must be activated as a service in the runlevel editor. To start it during system boot, 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, see an example page, assuming `apache2-example-pages` is installed.

23.5.3 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 described in *Generating Active Contents with Modules* on page 521.

23.5.4 Other Recommended Packages

It is advisable to install the documentation provided in the package `apache2-doc`. After the package has been installed and the server started as described in *Activating Apache* on the current page), the documentation can be accessed directly with the URL `http://localhost/manual`.

To be able to develop modules for Apache or compile third-party modules, the package `apache2-devel` is additionally required along with the corresponding development tools. These also contain the `apxs` tools, which are described in *Installing Modules with apxs* on this page.

23.5.5 Installing 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.

Install a module from source code with a command like `apxs2 -c -i -a mod_foo.c`. Other options of `apxs2` are described in its man page. The modules should then be activated in `/etc/sysconfig/apache2` with the entry `APACHE_MODULES` as described in *Configuration with SuSEconfig* on the next 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`.

23.6 Configuration

Following the installation of Apache, additional changes are only necessary if you have special needs or preferences. Apache can be configured either with YaST and SuSEconfig or by directly editing the file `/etc/apache2/httpd.conf`.

23.6.1 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 explaining 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 is interpreted by the main configuration file. In this way, changes to the configuration are 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 is only activated if the respective flag is set in the variable `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.

23.6.2 Manual Configuration

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. The loading sequence is determined by the modules themselves. 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 only want to make pages available on the intranet, because 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 from external access.

Directory

Use this directive to set 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.example.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 *AccessFileName* on the following page.

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 *AllowOverride* on the page before). 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 *Virtual Hosts* on page 524) 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 makes 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 *Directory* on page 515)

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 set `.html`, because this causes 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>
```

23.7 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`.

23.8 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 these 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.). Instead, 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, because 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.

23.8.1 Server Side Includes

Server-side includes (SSIs) 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 (`<!--`) 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 *Server-Side Includes* on page 517.

23.8.2 Common Gateway Interface

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.

23.8.3 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 (`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.

23.8.4 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 *Configuration* on page 513).

If necessary, additional directories can be specified. In this case, Apache searches these directories for executable programs. However, this represents a security risk, because any user can 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.

23.8.5 Generating Active Contents with Modules

Many modules are available for use with Apache. The term “module” is used in two different senses. First, there are modules that can be integrated in Apache to handle specific functions, such as the described modules for embedding programming languages.

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.

23.8.6 `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. 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 *Installation* on page 511). Following the installation, the Apache configuration file includes 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:

```
ScriptAlias /cgi-bin/ "/srv/www/cgi-bin/"
```

The entries for mod_perl are:

```
<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:

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, because 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, because 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 (despite some similarities to CGI), because no separate process is started for the interpreter.

23.8.7 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.

23.8.8 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/>.

23.8.9 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/>.

23.9 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. There are several options regarding virtual hosts:

- Name-based virtual hosts
- IP-based virtual hosts
- Operation of multiple instances of Apache on one machine

23.9.1 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, specify a suitable directive. `NameVirtualHost *. *` is sufficient to prompt Apache to accept all incoming requests. Subsequently, configure the individual hosts:

```
<VirtualHost *>
    ServerName www.example.com
    DocumentRoot /srv/www/htdocs/example.com
    ServerAdmin webmaster@example.com
    ErrorLog /var/log/apache2/www.example.com-error_log
    CustomLog /var/log/apache2/www.example.com-access_log common
</VirtualHost>

<VirtualHost *>
    ServerName www.myothercompany.com
    DocumentRoot /srv/www/htdocs/myothercompany.com
    ServerAdmin webmaster@myothercompany.com
    ErrorLog /var/log/apache2/www.myothercompany.com-error_log
    CustomLog /var/log/apache2/www.myothercompany.com-access_log common
</VirtualHost>
```

A `VirtualHost` entry must also be configured for the domain originally hosted on the server (`www.example.com`). 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 to 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 clients.

23.9.2 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, because 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 like the following:

```
/sbin/ifconfig eth0:0 192.168.1.20
/sbin/ifconfig eth0:1 192.168.1.21
```

All these IPs are 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/apache2/www.myothercompany.com-error_log
    CustomLog /var/log/apache2/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/apache2/www.anothercompany.com-error_log
  CustomLog /var/log/apache2/www.anothercompany.com-access_log common
</VirtualHost>
```

`VirtualHost` directives are only specified for the additional domains. The original domain (`www.example.com`) is configured through its own settings (under `DocumentRoot`, etc.) outside the `VirtualHost` blocks.

23.9.3 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:

```
Listen 192.168.1.10:80
```

For the other two instances:

```
Listen 192.168.1.20:80
Listen 192.168.1.21:80
```

23.10 Security

23.10.1 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.

23.10.2 Access Permissions

DocumentRoot Should Belong to root

By default, the `DocumentRoot` directory (`/srv/www/htdocs`) and the CGI directories 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

If users should be allowed to publish files, it is possible to declare a subdirectory of their home directory as suitable for web publishing. This subdirectory is traditionally named `~/public_html`. This is activated by default in SUSE LINUX. See *UserDir* on page 518 for details.

These web pages can be accessed by specifying the user in the URL. The URL contains the element `~username` as a shortcut to the respective directory in the home directory of the user. 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`.

23.10.3 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 *Security* on page 530.

23.11 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 `/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 the following command in a root console.

```
tail -f /var/log/apache2/*_log
```

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/>.

23.12 For More Information

23.12.1 Apache

Apache is shipped with detailed documentation. The installation of this documentation is described in *Installation* on page 511. 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>.

23.12.2 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/cgiic/>

23.12.3 Security

The latest patches for the SUSE LINUX 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.

23.12.4 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 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 to work with all computers and modify the files and subsequently have the latest version of the data available on all computers.

24.1	Available Data Synchronization Software	532
24.2	Determining Factors for Selecting a Program	534
24.3	Introduction to Unison	538
24.4	Introduction to CVS	539
24.5	Introduction to Subversion	542
24.6	Introduction to rsync	545
24.7	Introduction to mailsync	547

24.1 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.

24.1.1 Unison

Unison is not a network file system. Instead, 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 checksums, 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.

24.1.2 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.

24.1.3 subversion

In contrast to CVS, which “evolved,” subversion is a consistently designed project. subversion was developed as a technically improved successor to CVS. 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 (Web-based Distributed Authoring and Versioning). WebDAV extends the functionality of the HTTP protocol to allow collaborative write access to files on remote web servers.

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.

24.1.4 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.

24.1.5 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.

24.2 Determining Factors for Selecting a Program

24.2.1 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 between 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.

24.2.2 Portability

`subversion`, `CVS`, and `unison` are also available for many other operating systems, including various Unix and Windows systems.

24.2.3 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.

24.2.4 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 subversion 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.

As opposed to subversion or CVS, where it is possible to partially accept changes in cases of conflict, WebDAV only performs a check-in when the complete modification is considered successful.

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.

24.2.5 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` and `svn status` or `cvs update` are ignored due to the large number of files.

24.2.6 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.

24.2.7 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` and `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.

24.2.8 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.

24.2.9 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.

24.2.10 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.

24.2.11 Protection against Data Loss

CVS has been used by developers for a long time to manage program projects and is extremely stable. Because the development history is saved, CVS even provides protection against certain user errors, such as unintentional deletion of a file. 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 24.1: 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 Sel.	Dir.	Sel./file, dir.	Dir.	Mailbox
History	-	x/x	-	-
Hard Disk Space	o	--	o	+
GUI	+	o/o	-	-
Difficulty	+	o/o	+	o
Attacks	+(ssh)	+/(ssh)	+(ssh)	+(SSL)
Data Loss	+	++/++	+	+

24.3 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.

24.3.1 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 CVS on page 533).

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.

24.3.2 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 because 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 (for example, whether to perform the synchronization automatically in clear cases) can be controlled by means of command-line parameters specified when the program is started. View the complete list of all parameters 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`.

24.3.3 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 package `unison`.

24.4 Introduction to CVS

CVS is suitable for synchronization purposes if individual files are edited frequently and are stored in a file format, such as ASCII text or program source text. The use of CVS for synchronizing data in other formats, such as JPEG files, is possible, but leads to large amounts of data, because all variants of a file are stored permanently on the CVS server. In such cases, most of the capabilities of CVS cannot be used. The use of CVS for synchronizing files is only possible if all workstations can access the same server.

24.4.1 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 `cvs init` can be used to initialize the CVS server from the client side. This needs to be done only once.

Finally, the synchronization must be assigned a name. Select or create a directory on the client exclusively to contain files to manage with CVS (the directory can also be empty). The name of the directory is also the name of the synchronization. In this example, the directory is called `synchome`. Change to this directory and enter the following command to set the synchronization name to `synchome`:

```
cvs import synchome tux wilber
```

Many CVS commands require a comment. For this purpose, CVS starts an editor (the editor defined in the environment variable `$EDITOR` or `vi` if no editor was defined). The editor call can be circumvented by entering the comment in advance on the command line, such as in the following example:

```
cvs import -m 'this is a test' synchome tux wilber
```

24.4.2 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, commit the newly added files and directories with `cvs commit file1 directory1`.

If you change to another workstation, check out the synchronization repository, if this has not been done during an earlier session at the same workstation (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 in the same line and committed, you might get a conflict, indicated with **C**.

In this case, look at the conflict marks (`>>` and `<<`) in the file and decide between the two versions. As this can be a rather unpleasant job, you might decide to abandon your changes, delete the local file, and enter `cvs up` to retrieve the current version from the server.

24.4.3 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/>

24.5 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 *subversion* on page 533.

24.5.1 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` provides information about an existing repository.

```
svnlook info /path/to/repository
```

A server must be configured to allow different users to access the repository. Either use the Apache web server with WebDAV to do this or use `svnserve`, the server packaged with subversion. Once `svnserve` is up and running, the repository can be accessed with a URL with `svn://` or `svn+ssh://`. Users that should authenticate themselves when calling `svn` can be set in `/etc/svnserve.conf`.

A decision for Apache or for `svnserve` depends on many factors. It is recommended to browse the subversion book. More information about it can be found in *For More Information* on page 545.

24.5.2 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 one of the following commands:

```
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. Operations (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.

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.
...
```

24.5.3 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>.

24.6 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.

24.6.1 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 also creates a log file listing all connections. This file is stored in `/var/log/rsyncd.log`.

It is then possible to test the transfer from a client system. Do this with the following command:

```
rsync -avz sun::FTP
```

This command lists all files present in the directory `/srv/ftp` of the server. This request is also logged in the log file `/var/log/rsyncd.log`. To start an actual transfer, provide a target directory. Use `.` for the current directory. For example:

```
rsync -avz sun::FTP .
```

By default, no files are deleted while synchronizing with `rsync`. If this should be forced, the additional option `--delete` must be stated. To ensure that no newer files are deleted, the option `--update` can be used instead. Any conflicts that arise must be resolved manually.

24.6.2 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 sit at `http://rsync.samba.org/`.

24.7 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

24.7.1 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 `MOILLSYNC` 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.

24.7.2 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`.

24.7.3 For More Information

The README in `/usr/share/doc/packages/maillsync/`, which is included in mailsync, provides additional information. In this connection, RFC 2076 “Common Internet Message Headers” is of special interest.

Samba

Using Samba, a Unix machine can be configured as a file and print server for DOS, Windows, and OS/2 machines. This chapter introduces the basics of the Samba configuration and describes the YaST modules you can use for configuring Samba in your network.

25.1	Configuring the Server	553
25.2	Samba as Login Server	557
25.3	Configuring a Samba Server with YaST	559
25.4	Configuring Clients	560
25.5	Optimization	562

Samba has developed into a fully-fledged and rather complex product. This chapter presents an overview of its basic functionality. Details are discussed in the enclosed digital documentation. Enter `apropos samba` at the command line to display some manual pages or just browse the `/usr/share/doc/packages/samba` directory if Samba is installed for more online documentation and examples. A commented example configuration (`smb.conf.SuSE`) can be found in the `examples` subdirectory.

Some important new features of the enclosed version 3 of the `samba` package:

- Support for Active Directory
- Improved Unicode support
- The internal authentication mechanisms have been completely revised
- Improved support for the Windows 2000 and 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

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

Samba uses the SMB protocol (server message block) that is based on the Net-BIOS 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.

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.

All common operating systems, such as Mac OS X, Windows, and OS/2, support the SMB protocol. The TCP/IP protocol must be installed on all computers. Samba provides a client for the different UNIX flavors. For Linux, there is a kernel module for SMB that allows the integration of SMB resources on the Linux system level.

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.

25.1 Configuring the Server

If you intend to use Samba as a server, install `samba`. Start the services required for Samba with `rcnmb start` && `rcsmb start` and stop them 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.

25.1.1 The global Section

The following parameters of the `[global]` section need some adjustment to match the requirements of your network setup so other machines can access your Samba server via SMB in a Windows environment.

workgroup = TUX-NET This line assigns the Samba server to a workgroup. Replace `TUX-NET` with an appropriate workgroup of your networking environment. Your Samba server appears under its DNS name unless this name has been assigned to any other machine in the network. If the DNS name is not available, set the server name using `netbiosname=MYNAME`. See `mansmb.conf` for more details about this parameter.

os level = 2 This parameter triggers whether your Samba server tries to become LMB (local master browser) for its 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. First test the changes in an isolated network or at a noncritical time of day.

wins support and wins server To integrate your Samba server into an existing Windows network with an active WINS server, enable the `wins server` option and set its value to the IP address of that WINS server.

If your Windows machines are connected to separate subnets and should still be aware of each other, you need to set up a WINS server. To turn a Samba server into such a WINS server, set the option `wins support = Yes`. Make sure that only one Samba server of the network has this setting enabled. The options `wins server` and `wins support` must never be enabled at the same time in your `smb.conf` file.

25.1.2 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 25.1: 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 25.2: 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 with 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 makes the share invisible in the network environment.

read only = No By default, Samba prohibits write access to any exported share by means of the `read only = Yes` parameter. To make a share writable, set the value `read only = No`, which is synonymous with `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. For the group to have read or write access, deactivate the line `valid users = %S`.

25.1.3 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 registration, 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

25.2 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 done 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 Example 25.3 on the next page.

Example 25.3: Global Section in smb.conf

```
[global]
workgroup = TUX-NET
domain logons = Yes
domain master = Yes
```

If encrypted passwords are used for verification purposes — this is the default setting with well-maintained MS Windows 9x installations, MS Windows NT 4.0 from service pack 3, and all later products — the Samba server must be able to handle these. The entry `encrypt passwords = yes` in the `[global]` section enables this (with Samba version 3, this is now the default). In addition, it is necessary to prepare user accounts and passwords in an encryption format that conforms with Windows. Do this with the command `smbpasswd -a name`. Create the domain account for the computers, required by the Windows NT domain concept, with the following commands:

Example 25.4: 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 25.5: Automated Setup of a Machine Account

```
add machine script = /usr/sbin/useradd -g nogroup -c "NT Machine Account" \
-s /bin/false %m\$$
```

To make sure 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`.

25.3 Configuring a Samba Server with YaST

Start the server configuration with choosing the work group or domain that will be controlled by your new Samba server. Select an existing work group/domain from the drop down menu 'Workgroup or Domain Name' or enter a new one. In the next step, specify whether your server should act as PDC (primary domain controller) or as BDC (backup domain controller).

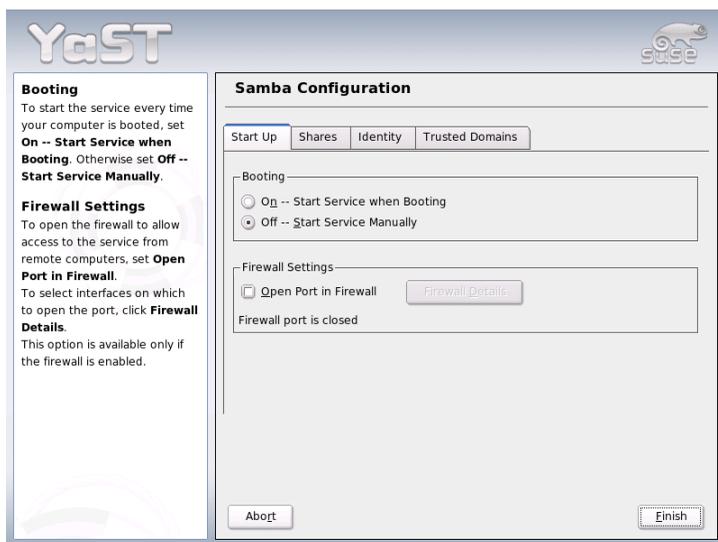


Figure 25.1: Samba Configuration — Start Up

Activate Samba in 'Start Up' (Figure 25.1). Use 'Open Ports in Firewall' and 'Firewall Details' to adapt the firewall on the server in such a way that the ports for the `netbios-ns`, `netbios-dgm`, `netbios-ssn`, and `microsoft-ds` services are open on all external and internal interfaces, thus ensuring a smooth operation of the Samba server.

In 'Shares' (Figure 25.2 on the next page), determine the Samba shares to activate. Use 'Toggle Status' to switch between 'Active' and 'Inactive'. Click 'Add' to add new shares.

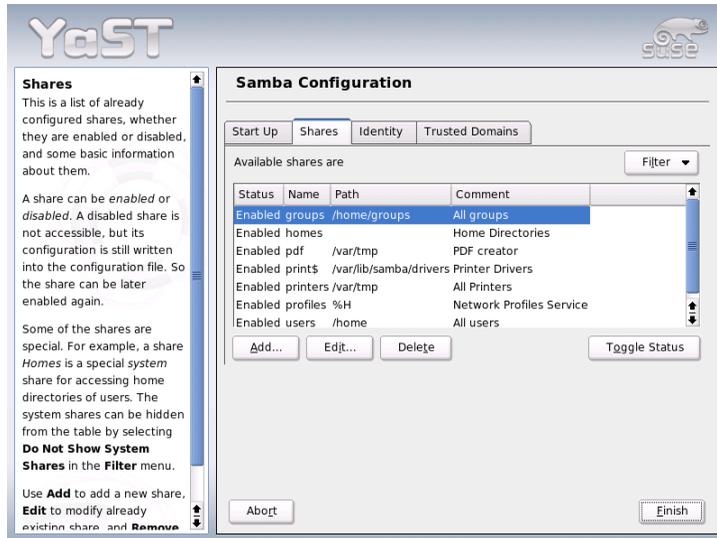


Figure 25.2: Samba Configuration — Shares

In ‘Identity’ (Figure 25.3 on the facing page), determine the domain with which the host is associated (‘Base Settings’) and whether to use an alternative host name in the network (‘NetBIOS Host Name’).

25.4 Configuring Clients

Clients can only access the Samba server via TCP/IP. NetBEUI and NetBIOS via IPX cannot be used with Samba.

25.4.1 Configuring a Samba Client with YaST

Configure a Samba client to access resources (files or printers) on the Samba server. Enter the domain or workgroup in the dialog ‘Samba Workgroup’. Click ‘Browse’ to display all available groups and domains, which can be selected with the mouse. If you activate ‘Also Use SMB Information for Linux Authentication’,

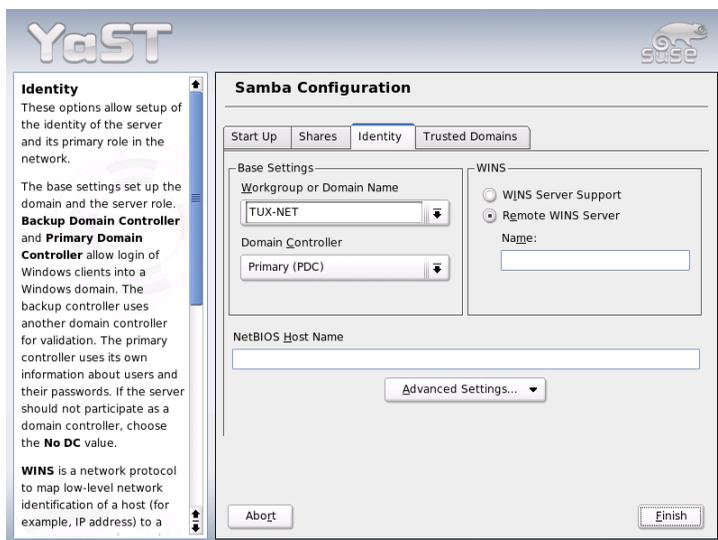


Figure 25.3: Samba Configuration — Identity

the user authentication runs over the Samba server. After completing all settings, click 'Finish' to finish the configuration.

25.4.2 Windows 9x and ME

Windows 9x and ME already have built-in support for TCP/IP. However, this is not installed as the default. To add TCP/IP, go to 'Control Panel' → 'System' and choose 'Add' → 'Protocols' → 'TCP/IP from Microsoft'. After rebooting your Windows machine, find the Samba server by double-clicking the desktop icon for the network environment.

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

25.5 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 relevant section of 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.

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 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

Internet

The Internet has become the number one platform for network communications worldwide. As a 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.

26.1	<code>smpppd</code> as Dial-up Assistant	564
26.2	Configuring an ADSL or T-DSL Connection	566
26.3	Proxy Server: Squid	567

26.1 smpppd as Dial-up Assistant

26.1.1 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 `ippod` or `pppd`. Basically, all that needs to be done to go online is to start these programs correctly.

If you have a flat-rate connection that does not generate any additional costs for the dial-up connection, simply start the respective daemon. Control the dial-up connection with a KDE applet or a command-line interface. If the Internet gateway is not the host you are using, you might want to control the dial-up connection by way of a network host.

This is where `smpppd` is involved. It provides a uniform interface for auxiliary programs and acts in two directions. First, it programs the required `pppd` or `ippod` 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.

26.1.2 Configuring smpppd

The connections provided by `smpppd` are automatically configured by YaST. The actual dial-up programs `kineternet` and `cineternet` are also preconfigured. Manual settings are only required to configure additional features of `smpppd`, such as remote control.

The configuration file of `smpppd` is `/etc/smpppd.conf`. By default, it does not enable remote control. The most important options of this configuration file are:

open-inet-socket = `<yes|no>` To control `smpppd` via the network, this option must be set to `yes`. The port on which `smpppd` listens is `3185`. If this parameter is set to `yes`, the parameters `bind-address`, `host-range`, and `password` should also be set accordingly.

bind-address = `<ip>` If a host has several IP addresses, use this parameter to determine at which IP address `smpppd` should accept connections.

host-range = <min ip> <max ip> The parameter `host-range` defines a network range. Hosts whose IP addresses are within this range are granted access to `smpppd`. All hosts not within this range are denied access.

password = <password> By assigning a password, limit the clients to authorized hosts. As this is a plain-text password, you should not overrate the security it provides. If no password is assigned, all clients are permitted to access `smpppd`.

slp-register = <yes|no> With this parameter, the `smpppd` service can be announced in the network via SLP.

More information about `smpppd` is available in `man 8 smpppd` and `man 5 smpppd.conf`.

26.1.3 Configuring `kinetnet`, `cinetnet`, and `qinetnet` for Remote Use

`kinetnet`, `cinetnet`, and `qinetnet` can be used locally as well as to control a remote `smpppd`. `cinetnet` is the command-line counterpart of the graphical `kinetnet`. To prepare these utilities for use with a remote `smpppd`, edit the configuration file `/etc/smpppd-c.conf` manually or using `kinetnet`. This file only uses three options:

sites = <list of sites> Here, tell the front-ends where to search for `smpppd`. The front-ends test the options in the order specified here. The option `local` orders the establishment of a connection to the local `smpppd`. `gateway` points to an `smpppd` on the gateway. The connection should be established as specified under `server` in `config-file`. `slp` orders the front-ends to connect to an `smpppd` found via SLP.

server = <server> Here, specify the host on which `smpppd` runs.

password = <password> Insert the password selected for `smpppd`.

If `smpppd` is active, you can now try to access it, for example, with `cinetnet --verbose --interface-list`. If you experience difficulties at this point, refer to `man 5 smpppd-c.conf` and `man 8 cinetnet`.

26.2 Configuring an ADSL or T-DSL Connection

26.2.1 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.

The `ppp` and `smpppd` packages must be installed. It is best to use YaST for this purpose. Configure your network card with YaST. Do not activate `dhcp`. Set a fixed IP address instead, such as `192.168.2.22`.

The parameters set with the DSL module of YaST are saved in the file `/etc/sysconfig/network/providers/<provider>`. 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`. If necessary, start the network with the command `rcnetwork start` and `smpppd` with the command `rcsmpppd start`.

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. Start the connection by clicking the icon and terminate the connection later with another click.

26.2.2 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 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 cron job, 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 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 with YaST or manually. Set the parameter `DEMAND=yes` in the file `/etc/sysconfig/network/providers/<provider>` and 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 article *DSL Gateway for Private Networks in SuSE Linux 8.0 or Higher* in our Support Database at <http://portal.suse.com>. To find the article, enter the keyword *gateway* in the search dialog.

26.3 Proxy Server: Squid

Squid is a widely-used proxy cache for Linux and UNIX platforms. This section 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.

26.3.1 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. This enables clients to 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 normally proxies only HTTP connections. It does also support the protocols FTP, Gopher, SSL, and WAIS, but it does not support other Internet protocols, such as Real Audio, news, or video conferencing. Because Squid only supports the UDP protocol to provide communication between different caches, many other multimedia programs are not supported.

26.3.2 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 *Configuring a Transparent Proxy* on page 578.

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 it is not desirable to increase the overall traffic on the network. For a very large network, it would make sense to configure a proxy server for every subnet-network and connect them to a parent proxy, which in turn is connected to the proxy cache of the ISP.

All this communication is handled by ICP (Internet cache protocol) running on top of the UDP protocol. Data transfers between caches are handled using HTTP (hypertext transmission protocol) based on TCP.

To find the most appropriate server from which to get the objects, one cache sends an ICP request to all sibling proxies. These answer the requests via ICP responses with a HIT code if the object was detected or a MISS if it was not. If multiple HIT responses were found, the proxy server decides from which server to download, depending on factors such as which cache sent the fastest answer or which one is closer. If no satisfactory responses are received, the request is sent to the parent cache.

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.

26.3.3 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 tend to be rather small, the seek time of the hard disk is more important than its data throughput. For the purposes of a proxy, hard disks with high rotation speeds are probably the better choice, because they allow the read-write head to be positioned in the required spot more quickly. Fast SCSI hard disks nowadays have a seek time of under four 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. If, for example, one GB is available for the cache and the users only surf ten MB per day, it would take more than one hundred days to fill the cache.

The easiest way to determine the needed cache size is to consider the maximum transfer rate of the connection. With a 1 Mbit/s connection, the maximum transfer rate is 125 KB/s. If all this traffic ends up in the cache, in one hour it would add up to 450 MB and, assuming that all this traffic is generated in only eight working hours, it would reach 3.6 GB in one day. Because the connection is normally not used to its upper volume limit, it can be assumed that the total data volume handled by the cache is approximately two GB. This is why two 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 *cachemgr.cgi* on page 581.

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.

26.3.4 Starting Squid

Squid is already preconfigured in SUSE LINUX, so you can start it right after the installation. To ensure a smooth start-up, the network should be configured in such a way that at least one name server and the Internet can be reached. Problems can arise if a dial-up connection is used with a dynamic DNS configuration. In 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`.

Commands for Starting and Stopping Squid

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 *Options for Access Controls* on page 576.

After modifying the configuration file `/etc/squid/squid.conf`, Squid must reload the configuration file. Do this with `rcsquid reload`. Alternatively, completely restart Squid with `rcsquid restart`.

The command `rcsquid status` can be used to check if the proxy is running. The command `rcsquid stop` causes Squid to shut down. This can take a while, because Squid waits up to half a minute (`shutdown_lifetime` option in `/etc/squid/squid.conf`) before dropping the connections to the clients and writing its data to the disk.

Caution

Terminating \{}protect\{}dbapplication{Squid}

Terminating Squid with `kill` or `killall` can damage the cache. To be able to restart Squid, the damaged 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 hierarchy or the log files. To remove these, delete the `/var/cache/squid` directory manually.

Local DNS Server

Setting up a local DNS server makes sense even if it does not manage its own domain. It then simply acts as a caching-only name server and is also able to resolve DNS requests via the root name servers without requiring any special configuration (see *Starting the Name Server BIND* on page 438). 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. Enter the name server of the provider in the configuration file `/etc/named.conf` under `forwarders` along with its IP address. However, if you have a firewall running, make sure DNS requests can pass it.

26.3.5 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 `/etc/squid/squid.conf` provides detailed information about the options and many examples. Nearly all entries begin with `#` (the lines are commented) and the relevant specifications can be found at the end of the line. The given values almost always correlate with the default values, so removing the comment signs without changing any of the parameters actually has little effect in most cases. If possible, leave the sample as it is and insert the options along with the modified parameters in the line below. In this way, easily interpret the default values and the changes.

Note

Adapting the Configuration File after an Update

If you have updated from an earlier Squid version, it is recommended to edit the new `/etc/squid/squid.conf` and only apply the changes made in the previous file. If you try to implement the old `squid.conf`, risk that the configuration no longer functions, because options are sometimes modified and new changes added.

Note

General Configuration Options (Selection)

http_port 3128 This is the port on which Squid listens for client requests. The default port is 3128, but 8080 is also common. If desired, specify several port numbers separated by blank spaces.

cache_peer *<hostname>* *<type>* *<proxy-port>* *<icp-port>*

Here, enter a parent proxy, for example, if you want to use the proxy of your ISP. As *<hostname>*, enter the name and IP address of the proxy to use and, as *<type>*, enter `parent`. For *<proxy-port>*, enter the port number that is also 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 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 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 cron job set in the configuration file `/etc/logrotate/squid`.

append_domain <domain> With *append_domain*, specify which domain to append automatically when none is given. Usually, your own domain is entered here, so entering *www* in the browser accesses your own web server.

forwarded_for on If you set the entry to *off*, Squid removes the IP address and the system name of the client from HTTP requests.

negative_ttl 5 minutes; negative_dns_ttl 5 minutes

Normally, you do not need to change these values. If you have a dial-up connection, however, the Internet may, at times, not be accessible. Squid makes a note of the failed requests then refuses to issue new ones, although the Internet connection has been reestablished. In a case such as this, change the *minutes* to *seconds* then, after clicking *Reload* in the browser, the dial-up process should be reengaged after a few seconds.

never_direct allow <acl_name> To prevent Squid from taking requests directly from the Internet, use the above command to force connection to another proxy. This must have previously been entered in *cache_peer*. If `all` is specified as the *<acl_name>*, force all requests to be forwarded directly to the *parent*. This might be necessary, for example, if you are using a provider that strictly stipulates the use of its proxies or denies its firewall direct Internet access.

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 separate package that can be installed and configured.

auth_param basic program /usr/sbin/pam_auth

If users must be authenticated on the proxy, set a corresponding program, such as `pam_auth`. When accessing `pam_auth` for the first time, the user sees a login window in which to enter the 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 for download from the Internet. To ensure that only clients with a successful ident lookup are permitted, define a corresponding ACL here:

```
acl identhosts ident REQUIRED

http_access allow identhosts
http_access deny all
```

Here, too, replace *REQUIRED* with a list of permitted user names. Using *ident* can slow down the access time quite a bit, because *ident* lookups are repeated for each request.

26.3.6 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.
- 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 the kernel of the proxy server supports a transparent proxy. If not, add these options to the kernel and recompile it. For more details, refer to *The Linux Kernel* on page 205.

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/services`) 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/services`) 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 26.1: 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 *Starting Squid* on page 571. 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`.

26.3.7 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 Example 26.2 on the next page.

Example 26.2: Access Rules

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl webservers src 192.168.1.7/255.255.255.255 # webservers IP
```

Then add the rules in Example 26.3.

Example 26.3: Access Rules

```
http_access allow manager localhost
http_access allow manager webservers
http_access deny manager
```

Configure a password for the manager for access to more options, like closing the cache remotely or viewing more information about the cache. For this, configure the entry `cachemgr_passwd` with a password for the manager and the list of options to view. This list appears as a part of the entry comments in `/etc/squid/squid.conf`.

Restart Squid every time the configuration file is changed. Do this easily with `rcsquid reload`.

Viewing the Statistics

Go to the corresponding web site — `http://webservers.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`.

26.3.8 squidGuard

This section is not intended to explain an extensive configuration of squidGuard, only to introduce it and give some advice for using it. For more in-depth configuration issues, refer to the squidGuard web site at `http://www.squidguard.org`.

squidGuard is a free (GPL), flexible, and fast filter, redirector, and access controller plug-in for Squid. It lets you define multiple access rules with different restrictions for different user groups on a Squid cache. squidGuard uses Squid’s standard redirector interface.

squidGuard can do the following:

- Limit the web access for some users to a list of accepted or well-known web servers or URLs.
- Block access to some listed or blacklisted web servers or URLs for some users.
- Block access to URLs matching a list of regular expressions or words for some users.
- Redirect blocked URLs to an “intelligent” CGI-based information page.
- Redirect unregistered users to a registration form.
- Redirect banners to an empty GIF.
- Use different access rules based on time of day, day of the week, date, etc.
- Use different rules for different user groups.

squidGuard and Squid cannot be used to:

- Edit, filter, or censor text inside documents.
- Edit, filter, or censor HTML-embedded script languages, such as JavaScript or VBscript.

Before it can be used, install `squidGuard`. Provide a minimal configuration file as `/etc/squidguard.conf`. Find configuration examples in <http://www.squidguard.org/config/>. Experiment later with more complicated configuration settings.

Next, create a dummy “access denied” page or a more or less complex CGI page to redirect Squid if the client requests a blacklisted web site. Using Apache is strongly recommended.

Now, configure Squid to use squidGuard. Use the following entry in the `/etc/squid.conf` file:

```
redirect_program /usr/bin/squidGuard
```

Another option called `redirect_children` configures the number of “redirect” (in this case `squidGuard`) processes running on the machine. `squidGuard` is fast enough to handle many requests: on a 500 MHz Pentium with 5,900 domains and 7,880 URLs (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.

26.3.9 Cache Report Generation with Calamaris

Calamaris is a Perl script used to generate reports of cache activity in ASCII or HTML format. It works with native Squid access log files. The Calamaris home page is located at <http://Calamaris.Cord.de/>. The program is quite easy to use.

Log in as `root` then enter `cat access.log.files | calamaris <options> > reportfile`. It is important when piping more than one log file that the log files are chronologically ordered with older files first. These are some options of the program:

- a** output all available reports
- w** output as HTML report
- l** include a message or logo in report header

More information about the various options can be found in the program’s manual page with `man calamaris`.

A typical example is:

```
cat access.log.2 access.log.1 access.log | calamaris -a -w \  
> /usr/local/httpd/htdocs/Squid/squidreport.html
```

This puts the report in the directory of the web server. Apache is required to view the reports.

Another powerful cache report generator tool is SARG (Squid Analysis Report Generator). More information about this is available at: <http://web.onda.com.br/orso/>.

26.3.10 For More Information

Visit the home page of Squid at <http://www.squid-cache.org/>. Here, find the “Squid User Guide” and a very extensive collection of FAQs on Squid.

Following the installation, a small howto about transparent proxies is available in howtoen as `/usr/share/doc/howto/en/txt/TransparentProxy.gz`. In addition, mailing lists are available for Squid at squid-users@squid-cache.org. The archive for this is located at <http://www.squid-cache.org/mail-archive/squid-users/>.

Security in Linux

Masquerading and the firewall ensure a controlled data flow and data exchange. SSH (Secure Shell) enables you to log in to remote hosts over an encrypted connection. The encryption of files or entire partitions protects your data in the event that third parties gain access to your system. Apart from these purely technical instructions, the last section focuses on security aspects in Linux networks.

27.1	Masquerading and Firewalls	588
27.2	SSH: Secure Network Operations	598
27.3	Encrypting Partitions and Files	603
27.4	Security and Confidentiality	605

27.1 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.

27.1.1 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).

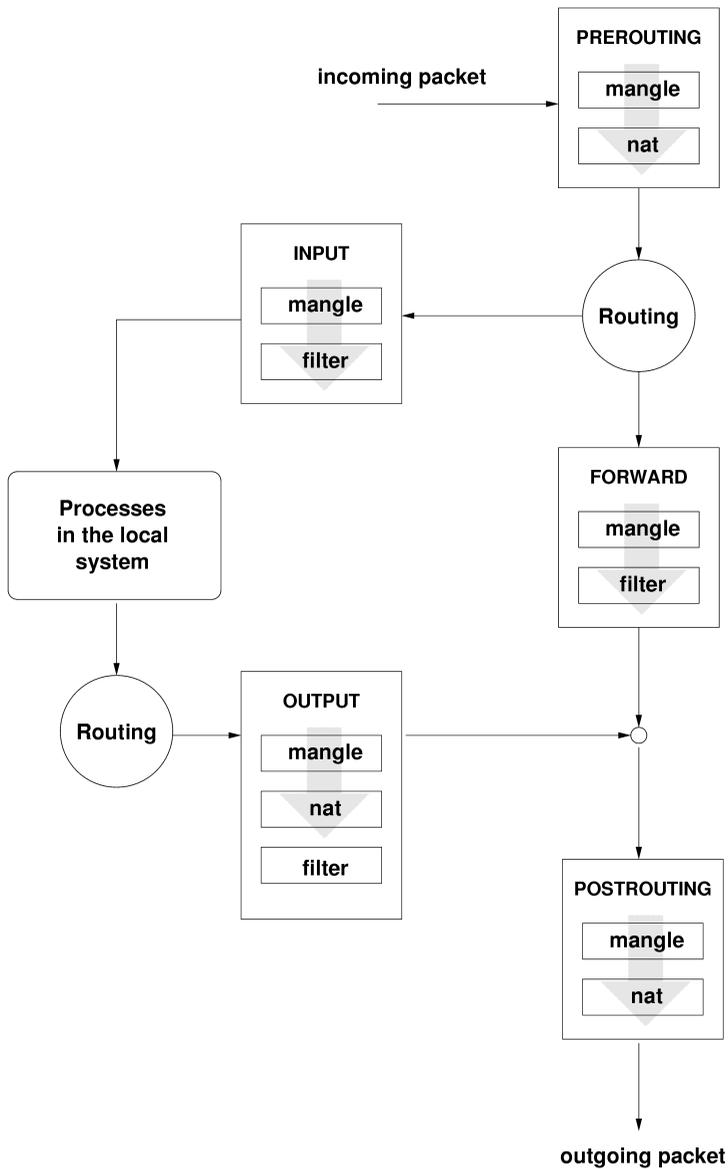


Figure 27.1: iptables: A Packet's Possible Paths

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 27.1 on the page before 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.

27.1.2 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 *Netmasks and Routing* on page 401) 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 netmask 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 must 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 reply packets. The router must identify these incoming packets and translate their target addresses, so packets can be forwarded to the correct host in the local network.

With the routing of inbound traffic depending on the masquerading table, there is no way to open a connection to an internal host from the outside. For such a connection, there would be no entry in the table. In addition, any connection already established has a status entry assigned to it in the table, so the entry cannot be used by another connection.

As a consequence of all this, you might experience some problems with a number of application protocols, such as ICQ, `cucme`, IRC (DCC, CTCP), and FTP (in PORT mode). Netscape, 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.

27.1.3 Firewalling Basics

Firewall is probably the term most widely used to describe a mechanism that provides and manages a link between networks while also controlling the data flow between them. Strictly speaking, the mechanism described in this section is called a *packet filter*. A packet filter regulates the data flow according to certain criteria, such as protocols, ports, and IP addresses. This allows you to block packets that, according to their addresses, are not supposed to reach your network.

To allow public access to your web server, for example, explicitly open the corresponding port. However, a packet filter does not scan the contents of packets with legitimate addresses, such as those directed to your web server. For example, if incoming packets were intended to compromise a CGI program on your web server, the packet filter would still let them through.

A more effective but more complex mechanism is the combination of several types of systems, such as a packet filter interacting with an application gateway or proxy. In this case, the packet filter rejects any packets destined for disabled ports. Only packets directed to the application gateway are accepted. This gateway or proxy pretends to be the actual client of the server. In a sense, such a proxy could be considered a masquerading host on the protocol level used by the application. One example for such a proxy is Squid, an HTTP proxy server. To use Squid, the browser must be configured to communicate via the proxy. Any HTTP pages requested are served from the proxy cache and pages not found in the cache are fetched from the Internet by the proxy. As another example, the SUSE proxy-suite (`proxy-suite`) provides a proxy for the FTP protocol.

The following section focuses on the packet filter that comes with SUSE LINUX. For further information about packet filtering and firewalling, read the Firewall HOWTO included in the `howto` package. If this package is installed, read the HOWTO with `less /usr/share/doc/howto/en/Firewall-HOWTO.gz`.

27.1.4 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 *Net-masks and Routing* on page 401), enable network address translation (NAT), so hosts on the internal network can access the external one.

Demilitarized Zone (DMZ) While hosts located in this zone can be reached both from the external and the internal network, they cannot access the internal network themselves. This setup can be used to put an additional line of defense in front of the internal network, because the DMZ systems are isolated from the internal network.

Any kind of network traffic not explicitly allowed by the filtering rule set is suppressed by `iptables`. Therefore, each of the interfaces with incoming traffic must be placed into one of the three zones. For each of the zones, define the services or protocols allowed. The rule set is only applied to packets originating from external hosts. Locally generated packets are not captured by the firewall.

The configuration can be performed with YaST (see *Configuration with YaST* on this page). It can also be made manually in the file `/etc/sysconfig/SuSEfirewall2`, which is well commented. Additionally, a number of example scenarios are available in `/usr/share/doc/SuSEfirewall2/EXAMPLES`.

Configuration with YaST

Note

Automatic Firewall Configuration

YaST automatically starts a firewall on all configured interfaces. If a server is configured and activated on the system, YaST modifies the automatically generated firewall configuration with the options 'Open Ports on Selected Interface in Firewall' or 'Open Ports on Firewall' in the server configuration modules. Some server module dialogs feature a 'Firewall Details' button for activating additional services and ports. The YaST firewall configuration module can only be used to activate, deactivate, or independently reconfigure the firewall.

Note

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 five sections:

Reconfigure or Stop If `SuSEfirewall2` is already active on the system because the automatic firewall configuration and initialization was not deactivated during the installation, this dialog appears. You can manually edit the firewall settings that were automatically generated by YaST with 'Reconfigure

Firewall Settings’ or select ‘Stop Firewall and Remove from Boot Process’ to stop and skip the firewall entirely when the system is booted. If there is no active firewall on the system, this dialog does not appear and the configuration begins with the ‘Basic Settings’.

Basic Settings Specify the interfaces to protect. To protect an individual host that is not connected to an internal network, enter the interface facing the Internet. Several interfaces can be specified in a comma-separated list. If the system is connected to an internal network, the interface facing the internal network must also be specified to protect the system against the internal network. This puts your system in a DMZ. This type of configuration is usually only needed in company networks. Click ‘Next’ to exit this dialog.

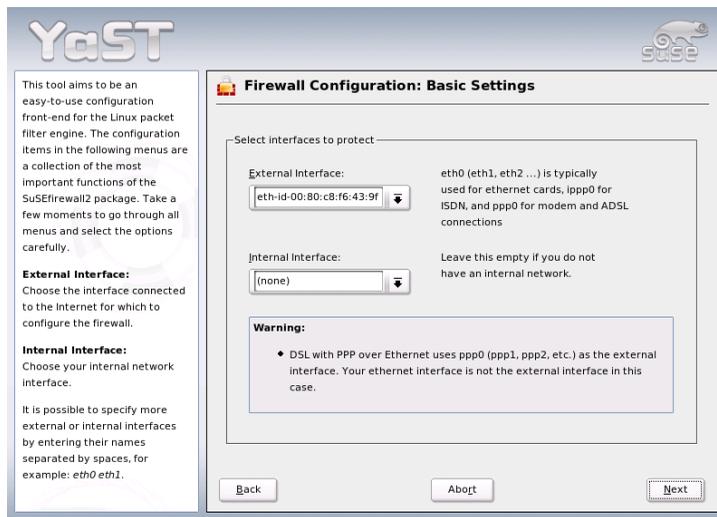


Figure 27.2: Selecting the Interfaces to Protect

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 your host is not used 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, disable this option if you want internal hosts to access the firewall.

‘Protect All Running Services’ Enable this to deny access to the TCP and UDP services of the firewall from the outside completely. 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.

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 found in larger organizations (corporate networks), which require extensive configuration and 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 `ispnp0`. 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 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 `diad` 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-DEFAULT 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 (020405B40402080A061AFEB0000000001030300)
```

27.1.5 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.

27.2 SSH: Secure Network Operations

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`.

27.2.1 The OpenSSH Package

SUSE LINUX installs the package `OpenSSH` by default. The programs `ssh`, `scp`, and `sftp` are then available as alternatives to `telnet`, `rsh`, `rcp`, and `ftp`.

27.2.2 The ssh Program

Using the `ssh` program, it is possible to log in to remote systems and work interactively. It replaces both `telnet` and `rlogin`. The `slogin` program is just a symbolic link pointing to `ssh`. For example, log in to the host `sun` with the command `ssh sun`. The host then prompts for the password on `sun`.

After successful authentication, you can work on the remote command line or use interactive applications, such as `YaST`. If the local 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 executed on `sun`.

27.2.3 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 than 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. Suppress all output by giving the option `-q`.

`scp` also provides a recursive copying feature for entire directories. The command `scp -r src/ sun:backup/` copies the entire contents of the directory `src` including all subdirectories to the `backup` directory on the host `sun`. If this subdirectory does not exist yet, it is created automatically.

The option `-p` tells `scp` to leave the time stamp of files unchanged. `-C` compresses the data transfer. This minimizes the data volume to transfer, but creates a heavier burden on the processor.

27.2.4 sftp — Secure File Transfer

The `sftp` program can be used instead of `scp` for secure file transfer. During an `sftp` session, you can use many of the commands known from `ftp`. The `sftp` program may be a better choice than `scp`, especially when transferring data for which the file names are unknown.

27.2.5 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. To continue 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 and a server key, which is regenerated by the SSH daemon every hour. Both allow the SSH client to encrypt a freely chosen session key, which is sent to the SSH server. The SSH client also tells the server which encryption method (cipher) to use.

Version 2 of the SSH protocol does not require a server key. Both sides use an algorithm according to Diffie-Helman to exchange their keys.

The private host and server keys are absolutely required to decrypt the session key and cannot be derived from the public parts. Only the SSH daemon contacted can decrypt the session key using its private keys (see `man /usr/share/doc/packages/openssh/RFC.nroff`). This initial connection phase can be watched closely by turning on the verbose debugging option `-v` of the SSH client.

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`.

27.2.6 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.

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`.

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`.

27.2.7 X, Authentication, and Forwarding Mechanisms

Beyond the previously described security-related improvements, SSH also simplifies the use of remote X applications. If you run `ssh` with the option `-X`, the `DISPLAY` variable is automatically set on the remote machine and all X output is exported to the remote machine over the existing SSH connection. At the same time, X applications started remotely and locally viewed with this method cannot be intercepted by unauthorized individuals.

By adding the option `-A`, the `ssh-agent` authentication mechanism is carried over to the next machine. This way, you can work from different machines without having to enter a password, but only if you have distributed your public key to the destination hosts and properly saved it there.

Both mechanisms are deactivated in the default settings, but can be permanently activated at any time in the 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`.

27.3 Encrypting Partitions and Files

27.3.1 Application Scenarios

Every user has some confidential data not intended to be accessible by third parties. The more connected and mobile you are, the more paranoid you should be handling your data. The encryption of files or of whole partitions is always sensible when third parties have access over a network connection or direct physical access. The following list features a few imaginable usage scenarios.

Laptops If you commute or travel with your laptop, it can be a very good idea to encrypt the partitions on the hard disk that contain confidential data. If you lose your laptop or it is stolen, your data is safe from access in an encrypted file system residing in a single file.

Removable Media USB flash drives or external hard disks are as prone to being stolen as laptops. A encrypted file system offers protection against third parties in those cases.

27.3.2 Setting up a Crypto File System with YaST

YaST offers the encryption of files or partitions during installation as well as in an already installed system. An encrypted file can always be created, because it fits nicely in an existing partition layout. To encrypt an entire partition, you need to dedicate a partition for encryption in the partition layout. The standard partitioning proposal as suggested by YaST does not, by default, include an encrypted partition. Add it manually in the partitioning dialog.

Creating an Encrypted Partition during Installation

Caution

Password Input

Observe the warnings about password security when creating the password for encrypted partitions and memorize it well. Without the password, it is impossible to access the encrypted data.

Caution

The YaST expert dialog for partitioning, described in *Expert Partitioning with YaST* on page 19, offers the options necessary for creating an encrypted partition. Click 'Create' like when creating a regular partition. In the dialog that opens, enter partitioning parameters for the new partition, such as the desired formatting and the mount point. Complete the creating by clicking 'Encrypt File System'. In the following dialog, set the password and repeat it for security reasons. The new encrypted partition is created after the partitioning dialog is completed by clicking 'OK'. The operating system queries the user for that password while booting before the partition can be mounted.

If you do not want to mount the encrypted partition during start-up, hit `(Enter)` when prompted for the password. Then decline the offer to enter the password again. The encrypted file system is not mounted in this case and the operating system continues booting, which is the safer way of protecting your data. The partition is available to all users once it has been mounted.

If the encrypted file system should only be mounted when necessary, check 'Do Not Mount During Booting' in the 'fstab Options' dialog. The corresponding partition will not be considered for mounting during system start-up. To make it available afterwards, mount it manually with `mount <name_of_partition> <mount_point>`. Enter the password when prompted to mount the partition. When finished with the partition, unmount it with `umount name_of_partition` to keep it safe from access by other users.

Creating an Encrypted Partition on a Running System

Caution

Activating Encryption in a Running System

It is also possible to create encrypted partitions on a running system like during installation. However, encrypting an existing partition destroys all data on it.

Caution

On a running system, select ‘System’ → ‘Partitioning’ in the YaST control center. Click ‘Yes’ to proceed. Instead of selecting ‘Create’, as mentioned above, click ‘Edit’. The rest of the procedure is the same.

Installing Encrypted Files

As well as using a partition, it is possible to create encrypted file systems within single files for holding confidential data. These are created from the same YaST dialog. Select ‘Crypt File’ and enter the path to the file to create along with its intended size. Accept the proposed settings for formatting and the file system type. Then specify the mount point and decide whether the crypto file system should be mounted during boot.

27.3.3 Encrypting the Content of Removable Media

Removable media, like external hard disks or USB flash drives, are recognized by YaST like any other hard disk. It is possible to encrypt files or partitions on such media by proceeding as described above. Do not select to mount these media during boot, because they are usually only connected while the system is running.

27.4 Security and Confidentiality

One of the main characteristics of a Linux or UNIX system is its ability to handle several users at the same time (multiuser) and to allow these users to perform several tasks (multitasking) on the same computer simultaneously. Moreover, the operating system is network transparent. The users often do not know whether the data and applications they are using are provided locally from their machine or made available over the network.

With the multiuser capability, the data of different users must be stored separately. Security and privacy need to be guaranteed. Data security was already an important issue, even before computers could be linked through networks. Just like today, the most important concern was the ability to keep data available in spite of a lost or otherwise damaged data medium, a hard disk in most cases.

This section is primarily focused on confidentiality issues and on ways to protect the privacy of users, but it cannot be stressed enough that a comprehensive security concept should always include procedures to have a regularly updated, workable, and tested backup in place. Without this, you could have a very hard time getting your data back — not only in the case of some hardware defect, but also if the suspicion arises that someone has gained unauthorized access and tampered with files.

27.4.1 Local Security and Network Security

There are several ways of accessing data:

- personal communication with people who have the desired information or access to the data on a computer
- directly from the console of a computer (physical access)
- over a serial line
- using a network link

In all these cases, a user should be authenticated before accessing the resources or data in question. A web server might be less restrictive in this respect, but you still would not want it to disclose all your personal data to any surfer.

In the list above, the first case is the one where the highest amount of human interaction is involved, such as when you are contacting a bank employee and are required to prove that you are the person owning that bank account. Then you are asked to provide a signature, a PIN, or a password to prove that you are the person you claim to be. In some cases, it might be possible to elicit some intelligence from an informed person just by mentioning known bits and pieces to win the confidence of that person by using clever rhetoric. The victim could be led to reveal gradually more information, maybe without even becoming aware of it. Among hackers, this is called *social engineering*. You can only guard against this by educating people and by dealing with language and information in a conscious way. Before breaking into computer systems, attackers often try to target receptionists, service people working with the company, or even family members. In many cases, such an attack based on social engineering is only discovered at a much later time.

A person wanting to obtain unauthorized access to your data could also use the traditional way and try to get at your hardware directly. Therefore, the machine should be protected against any tampering so that no one can remove, replace, or cripple its components. This also applies to backups and even any network cable or the power cord. Also secure the boot procedure, because there are some well-known key combinations that might provoke unusual behavior. Protect yourself against this by setting passwords for the BIOS and the boot loader.

Serial terminals connected to serial ports are still used in many places. Unlike network interfaces, they do not rely on a network protocol to communicate with the host. A simple cable or an infrared port is used to send plain characters back and forth between the devices. The cable itself is the weakest point of such a system: with an older printer connected to it, it is easy to record anything that runs over the wires. What can be achieved with a printer can also be accomplished in other ways, depending on the effort that goes into the attack.

Reading a file locally on a host requires other access rules than opening a network connection with a server on a different host. There is a distinction between local security and network security. The line is drawn where data must be put into packets to be sent somewhere else.

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. Prevent others from using such parameters during boot by setting an additional password in `/boot/grub/menu.lst` (see *Booting and Boot Managers* on page 185). 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.

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, especially if the program is being executed with special privileges (see *File Permissions* on the page before).

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 *File Permissions* on the preceding 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 X Authentication

As mentioned at the beginning, network transparency is one of the central characteristics of a UNIX system. X, the windowing system of UNIX operating systems, can make use of this feature in an impressive way. With X, it is basically no problem to log in at a remote host and start a graphical program that is then 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 *SSH: Secure Network Operations* on page 598.

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 in *Buffer Overflows and Format String Bugs* on page 609, 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 notice this, because they get an error message saying the connection was terminated due to a failure.

The fact that there are protocols not secured against hijacking through encryption, which only perform a simple authentication procedure upon establishing the connection, makes it easier for attackers.

Spoofing is an attack where packets are modified to contain counterfeit source data, usually the IP address. Most active forms of attack rely on sending out such fake packets — something that, on a Linux machine, can only be done by the superuser (`root`).

Many of the attacks mentioned are carried out in combination with a DoS. If an attacker sees an opportunity to bring down a certain host abruptly, even if only for a short time, it makes it easier for him to push the active attack, because the host will not be able to interfere with the attack for some time.

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 among hosts to disguise itself as one of the trusted hosts. Usually, the attacker analyzes some packets received from the server to get the necessary information. The attacker often needs to target a well-timed DoS attack at the name server as well. Protect yourself by using encrypted connections that are able to verify the identity of the hosts to which to connect.

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.

27.4.2 Some General Security Tips and Tricks

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 fifteen and twenty 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 makes your system safer. Open ports, with the socket state LISTEN, can be found with the program `netstat`. As for the options, it is recommended to use `netstat -ap` or `netstat -anp`. The `-p` option allows you to see which process is occupying a port under which name.

Compare the `netstat` results with those of a thorough port scan done from outside your host. An excellent program for this job is `nmap`, which not only checks out the ports of your machine, but also draws some conclusions as to which services are waiting behind them. However, port scanning may be interpreted as an aggressive act, so do not do this on a host without the explicit approval of the administrator. Finally, remember that it is important not only to scan TCP ports, but also UDP ports (options `-sS` and `-sU`).

- To monitor the integrity of the files of your system in a reliable way, use the program `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 works, it might actually be worthless.

- Check your log files. Whenever possible, write a small script to search for suspicious entries. Admittedly, this is not exactly a trivial task. In the end, only you can know which entries are unusual and which are not.
- Use `tcp_wrapper` to restrict access to the individual services running on your machine, so you have explicit control over which IP addresses can connect to a service. For further information regarding `tcp_wrapper`, consult the manual pages of `tcpd` and `hosts_access` (`man 8 tcpd`, `man hosts_access`).
- Use `SUSEfirewall` to enhance the security provided by `tcpd` (`tcp_wrapper`).
- Design your security measures to be redundant: a message seen twice is much better than no message at all.

27.4.3 Using the Central Security Reporting Address

If you discover a security-related problem (please check the available update packages first), write an e-mail to `security@suse.de`. Please include a detailed description of the problem and the version number of the package concerned. SUSE will try to send a reply as soon as possible. You are encouraged to pgp encrypt your e-mail messages. SUSE's pgp key is:

```
ID:3D25D3D9 1999-03-06 SUSE Security Team <security@suse.de>  
Key fingerprint = 73 5F 2E 99 DF DB 94 C4 8F 5A A3 AE AF 22 F2 D5
```

This key is also available for download from <http://www.suse.de/security>.

Part IV

Administration

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.

28.1	Advantages of ACLs	622
28.2	Definitions	623
28.3	Handling ACLs	623
28.4	ACL Support in Applications	632

28.1 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.

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 that 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`.

28.2 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*).

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 28.1 on the next 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.

28.3 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.

28.3.1 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 28.1 provides a summary of the various types of ACL entries that are possible.

Table 28.1: 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 28.2 demonstrates this mechanism.

Table 28.2: Masking Access Permissions

Entry Type	Text Form	Permissions
named user	user:jane:r-x	r-x
mask	mask::rw-	rw-
	effective permissions:	r--

28.3.2 ACL Entries and File Mode Permission Bits

Figure 28.1 and Figure 28.2 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.

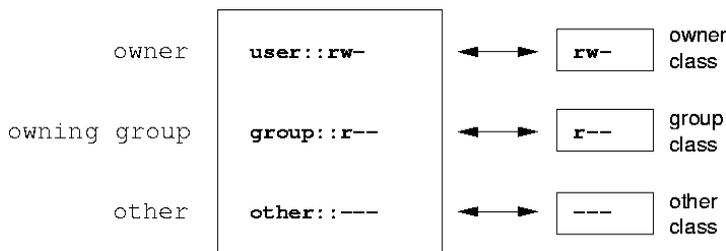


Figure 28.1: Minimum ACL: ACL Entries Compared to Permission Bits

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 28.1. In the case of an extended ACL — with *mask* — the *group class* permissions are mapped to the *mask* entry. This is shown in Figure 28.2.

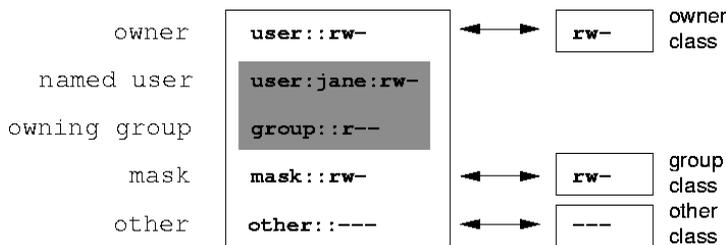


Figure 28.2: Extended ACL: ACL Entries Compared to Permission Bits

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.

28.3.3 A Directory with Access ACL

The handling of access ACLs is demonstrated in three steps by means of the following example:

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 *ACL Entries and File Mode Permission Bits* on page 625. 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 28.2 on page 624). 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:---
```

28.3.4 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:

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 inherits 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`, as is 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:----
```

`touch` passes `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 *Effects of a Default ACL* on page 629).

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.

28.3.5 The ACL Check Algorithm

A check algorithm is applied before any process or application is granted access to an ACL-protected file system object. As a basic rule, the ACL entries are examined in the following sequence: *owner*, *named user*, *owning group* or *named group*, and *other*. The access is handled in accordance with the entry that best suits the process. Permissions do not accumulate.

Things are more complicated if a process belongs to more than one group and would potentially suit several *group* entries. An entry is randomly selected from the suitable entries with the required permissions. It is irrelevant which of the entries triggers the final result “access granted”. Likewise, if none of the suitable *group* entries contains the correct permissions, a randomly selected entry triggers the final result “access denied”.

28.4 ACL Support in 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 http://sdb.suse.de/en/sdb/html/81_acl.html and <http://acl.bestbits.at/>. Also see the man pages for `getfacl`, `acl(5)`, and `setfacl(1)`.

Note

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.

29.1	List of Open Files: <code>lsdf</code>	636
29.2	User Accessing Files: <code>fuser</code>	637
29.3	File Properties: <code>stat</code>	638
29.4	Processes: <code>top</code>	638
29.5	Process List: <code>ps</code>	639
29.6	Process Tree: <code>pstree</code>	641
29.7	Who Is Doing What: <code>w</code>	642
29.8	Memory Usage: <code>free</code>	642
29.9	Kernel Ring Buffer: <code>dmesg</code>	643
29.10	File Systems and Their Usage: <code>mount</code> , <code>df</code> , and <code>du</code>	643
29.11	The <code>/proc</code> File System	644
29.12	<code>procinfo</code>	646
29.13	PCI Resources: <code>lspci</code>	647
29.14	System Calls of a Program Run: <code>strace</code>	648
29.15	Library Calls of a Program Run: <code>ltrace</code>	650
29.16	Specifying the Required Library: <code>ldd</code>	650
29.17	Additional Information about ELF Binaries	651
29.18	Interprocess Communication: <code>ipcs</code>	651
29.19	Time Measurement with <code>time</code>	652

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). Comments 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.

29.1 List of Open Files: `lsdf`

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:

```
$ lsdf -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
```

29.2 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.

29.3 File Properties: stat

The command `stat` displays file properties:

```
$ 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:

```
$ stat . --filesystem
  File: "."
    ID: 0          Namelen: 255      Type: ext2/ext3
Blocks: Total: 19347388  Free: 17831731  Available: 16848938  Size: 4096
Inodes: Total: 9830400  Free: 9663967
```

If you use the z shell (`zsh`), you must enter `/usr/bin/stat`, because the z shell has a shell built-in `stat` with different options and a different output format:

```
% type stat
stat is a shell builtin
% stat .
device 769
inode 4554808
mode 16877
nlink 12
uid 11994
gid 50
rdev 0
size 4096
atime 1091536882
mtime 1091535740
ctime 1091535740
blksize 4096
blocks 8
link
```

29.4 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 nsd
 1363 root        15   0   488  452  404  S  0.0   0.1    0:00.04 nsd
 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 **F** while `top` is running, a menu opens with which to make extensive changes to the format of the output.

The parameter `-U <UID>` monitors only the processes associated with a particular user. Here, `<UID>` is the user ID of the user. The following command returns the UID of the user on the basis of the user name and displays his processes:

```
$ top -U $(id -u <username>)
```

29.5 Process List: ps

The command `ps` produces a list of processes. If the parameter `r` is added, only processes currently using computing time 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
```

The process list can be formatted according to your needs. The option `-L` returns a list of all keywords. Enter the following command to issue a list of all processes sorted by memory usage:

```
$ ps ax --format pid,rss,cmd --sort rss
  PID  RSS  CMD
    2     0 [ksoftirqd/0]
    3     0 [events/0]
   17     0 [kblockd/0]
[... ]
 10164 5260 xterm
 31110 5300 xterm
 17010 5356 xterm
 3896 29292 /usr/X11R6/bin/X -nolisten tcp -br vt7 -auth /var/lib/xdm/authdir/au
```

29.6 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
```

29.7 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 -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.

29.8 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 the following line:

```
-/+ buffers/cache:          191          311
```

This 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.

29.9 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 flash drive.

Older events are logged in the files `/var/log/messages` and `/var/log/warn`.

29.10 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/hdal 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,wsiz=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/hdal       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 without delay. 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.

29.11 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
fddiv_bug     : no
[...]
```

The allocation and use of interrupts can be queried with the following command:

```
$ 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`. 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
bffffe000-c0000000 rw-p bffffe000 00:00 0
fffffe000-ffffff00 ---p 00000000 00:00 0

```

29.12 procinfo

Important information from the `/proc` file system is summarized by the command `procinfo`:

```

$ procinfo
Linux 2.6.4-54.5-default (geeko@buildhost) (gcc 3.3.3 ) #1 1CPU [roth.suse.de]

```

Memory:	Total	Used	Free	Shared	Buffers
Mem:	516696	513200	3496	0	43284
Swap:	530136	1352	528784		

```

Bootup: Wed Jul 7 14:29:08 2004      Load average: 0.07 0.04 0.01 1/126 5302

```

```

user  :      2:42:28.08   1.3% page in :      0
nice  :      0:31:57.13   0.2% page out:      0
system:      0:38:32.23   0.3% swap in :      0
idle  :   3d 19:26:05.93  97.7% swap out:      0
uptime:  4d  0:22:25.84          context :207939498

```

```

irq 0: 776561217 timer          irq 8:      2 rtc
irq 1:  276048 i8042           irq 9:     24300 VIA8233
irq 2:      0 cascade [4]      irq 11:  38610118 acpi, eth0, uhci_hcd
irq 3:      3                irq 12:  3435071 i8042
irq 4:      3                irq 14:  2236471 ide0
irq 6:      2                irq 15:    251 ide1

```

To see all the information, use the parameter `-a`. The parameter `-n⟨N⟩` produces updates of the information every $\langle N \rangle$ seconds. In this case, terminate the program by pressing `Ⓞ`.

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

```

29.13 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`.

29.14 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 -e open 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
fcntl164(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 -e open ls myfile.txt

open("/etc/ld.so.preload", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
open("/lib/tls/librt.so.1", O_RDONLY) = 3
open("/lib/libacl.so.1", O_RDONLY) = 3
open("/lib/libselinux.so.1", O_RDONLY) = 3
open("/lib/tls/libc.so.6", O_RDONLY) = 3
open("/lib/tls/libpthread.so.0", O_RDONLY) = 3
open("/lib/libattr.so.1", O_RDONLY) = 3
open("/proc/mounts", O_RDONLY) = 3
[...]
open("/proc/filesystems", O_RDONLY) = 3
open("/proc/self/attr/current", O_RDONLY) = 4

```

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`.

29.15 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:

```
$ ltrace -c find /usr/share/doc
% time      seconds  usecs/call   calls   errors  syscall
-----
 86.27      1.071814      30      35327
10.15      0.126092      38       3297
 2.33      0.028931       3      10208
 0.55      0.006861       2       3122
 0.39      0.004890       3       1567
[... ]
 0.00      0.000003       3         1
 0.00      0.000001       1         1
-----
100.00      1.242403                    58269          3 total
```

29.16 Specifying the Required Library: ldd

The command `ldd` can be used to find out which libraries would load the dynamic executable specified as argument:

```
$ 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)
libseline.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/sash
      not a dynamic executable
$ file /bin/sash
/bin/sash: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), \
for GNU/Linux 2.2.5, statically linked, stripped
```

29.17 Additional Information about ELF Binaries

The content of binaries can be read with the `readelf` utility. This even works with ELF files that were built for other hardware architectures:

```
$ readelf --file-header /bin/ls
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00
  Class:                   ELF32
  Data:                     2's complement, little endian
  Version:                  1 (current)
  OS/ABI:                   UNIX - System V
  ABI Version:              0
  Type:                     EXEC (Executable file)
  Machine:                  Intel 80386
  Version:                  0x1
  Entry point address:      0x8049b40
  Start of program headers: 52 (bytes into file)
  Start of section headers: 76192 (bytes into file)
  Flags:                    0x0
  Size of this header:      52 (bytes)
  Size of program headers:  32 (bytes)
  Number of program headers: 9
  Size of section headers:  40 (bytes)
  Number of section headers: 29
  Section header string table index: 26
```

29.18 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
```

29.19 Time Measurement with time

The time spent by commands can be determined with the `time` utility. This utility is available in two versions, as a shell built-in and as a program (`/usr/bin/time`).

```
$ time find . > /dev/null  
  
real    0m4.051s  
user    0m0.042s  
sys     0m0.205s
```

Part V

Appendix



Information Sources and Documentation

A wide range of information sources exist that are applicable to your SUSE LINUX system. Some of these sources are SUSE-specific, but many are more general sources. Some are already available on your system or installation media and others can be accessed over the Internet.

SUSE Documentation

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 SUSE Help Center gives 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 itself is installed. As a rule, documentation of this kind is kept quite general so 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?” or “What is a RAID?” Texts in this category are generally quite short.

Guides are documents that can deal with a topic in much greater detail than HOWTOs and FAQs. Examples include kernel programming and network administration. 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 or 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, or similar item. A man page is normally divided into various sections, such as name, syntax, description, options, and files.

To display a man page, enter `man` followed by the name of the command, as in `man ls`, which shows a help text for the `ls` command. Use the cursor keys to move the visible area. `Q` exits `man`. To print a man page (for example for the command `ls`), enter a command like `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, for `grep`. Access it with `info grep`.

Info pages are more detailed than man pages. They are divided into different *nodes* — pages that can be read with an info reader, which works much like a web browser. Use `P` (previous page) and `N` (next page) to navigate in an info page. `Q` exits `info`. Other keys are listed in the `info` documentation (`info info`).

Both man pages and info pages can be read in Konqueror. Enter `man: <command>` or `info: <command>` in the URL line to open the desired documentation.

Standards and Specifications

There are various sources that provide information about standards or specifications.

www.linuxbase.org The Free Standards Group is an independent nonprofit organization that promotes the distribution of free software and Open Source software. The organization endeavors to achieve this by defining distribution-independent standards. The maintenance of several standards, such as the important LSB (Linux Standard Base), is supervised by this organization.

http://www.w3.org The World Wide Web Consortium (W3C) is certainly one of the best-known standards organizations. It was founded in October 1994 by TIM BERNERS-LEE and concentrates on standardizing web technologies. W3C promotes the dissemination of open, license-free, and manufacturer-independent specifications, such as HTML, XHTML, and XML. These web standards are developed in a four-stage process in *working groups* and are presented to the public as *W3C recommendations* (REC).

http://www.oasis-open.org OASIS (Organization for the Advancement of Structured Information Standards) is an international consortium specializing in the development of standards for web security, e-business, business transactions, logistics, and interoperability between various markets.

http://www.ietf.org The Internet Engineering Task Force (IETF) is an internationally active cooperative of researchers, network designers, suppliers, and users. It concentrates on the development of Internet architecture and the smooth operation of the Internet by means of protocols.

Every IETF standard is published as an RFC (Request for Comments) and is available free-of-charge. There are six types of RFC: proposed standards, draft standards, Internet standards, experimental protocols, information documents, and historic standards. Only the first three (proposed, draft, and full) are IETF standards in the narrower sense (see <http://www.ietf.org/rfc/rfc1796.txt>).

http://www.ieee.org The Institute of Electrical and Electronics Engineers (IEEE) is an organization that draws up standards in the areas of information technology, telecommunication, medicine and health care, transport, and others. IEEE standards are subject to a 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 subject to a charge.

<http://www.din.de>, <http://www.din.com>

The Deutsches Institut für Normung (DIN) is a registered technical and scientific association. It was founded in 1917. According to DIN, the organization is “the institution responsible for standards in Germany and represents German interests in worldwide and European standards organizations.”

The association brings together manufacturers, consumers, trade professionals, service companies, scientists and others who have an interest in the establishment of standards. The standards are subject to a charge and can be ordered using the DIN home page.

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 reiserfs 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 system tree (`--rebuild-tree`). Normally you only need this option if the `--check` option reports "corruption that can be fixed with `--fix-fixable`". This includes: zeroing invalid data-block pointers, correcting `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 discovered 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 corruption 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 fixable mode. If the flag indicating a fatal corruption is found set in the superblock, then reiserfsck 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 partition 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>.

TODD

Faster recovering, signal handling, i/o error handling, etc.

SEE ALSO

`mkreiserfs(8)`, `reiserfstune(8)` `resize_reiserfs(8)`, `debugreiserfs(8)`,

Manual Page of e2fsck

E2FSCK(8)

E2FSCK(8)

NAME

e2fsck - check a Linux second extended file system

SYNOPSIS

```
e2fsck [ -pacyrdfvstDFSV ] [ -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 system (ext2fs). E2fsck also supports ext2 filesystems containing a journal, which are also sometimes known as ext3 filesystems, by first applying the journal to the filesystem before continuing with normal e2fsck processing. After the journal has been applied, a filesystem will normally 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 compressing directories for smaller directories, or for filesystems using traditional linear directories.
- 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 supported:
 - ea_ver=extended_attribute_version
Assume the format of the extended attribute blocks in the filesystem is the specified version number. The version 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 numbers 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 filesystem is already in the standard byte-order, e2fsck will take no action.
- S This option will byte-swap the filesystem, regardless 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 following 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 completion 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 trouble, try running the `debugfs(8)` command and send the output of the `stat(lu)` 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 displays when it is run, so I know which version you are running.

AUTHOR

This version of e2fsck was written by Theodore Ts'o <tytso@mit.edu>.

SEE ALSO

`mke2fs(8)`, `tune2fs(8)`, `dumpe2fs(8)`, `debugfs(8)`

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.

Glossary

access permissions

The account is defined by the user name or login name and the password. The access permissions are generally set by the *system administrator*. The access permissions define to which user group the new user is assigned and the resulting permissions.

account

See *access permissions*.

ACL (Access Control List)

Extension of the conventional permission concept for files and directories.

ADSL (Asymmetric Digital Subscriber Line)

Transmission procedure that transmits data about one hundred times faster than ISDN in the telephone network.

AGP (Accelerated Graphics Port)

A high-speed slot for graphics cards based on PCI, but offering a larger *bandwidth*. Furthermore, AGP graphics cards can revert directly (without routing around the processor) to the *random access memory* and *main memory*, in contrast to PCI models, to swap graphics data there.

ATAPI (Advanced Technology Attachment Packet Interface)

ATAPI is a type of CD-ROM drive that is connected to an (E)IDE controller. Apart from ATAPI drives, there are SCSI CD-ROM drives, handled by a SCSI controller, and proprietary CD-ROM drives that use their own controller or are connected to a sound card.

backup

A backup is a duplicate of data used to restore data that has been damaged or lost. Backups should be made regularly, especially of important files.

bandwidth

Maximum load capacity of a data channel.

BIOS

Small component responsible for the initialization of important hardware processes. This essential procedure is complete when the boot menu appears on the screen.

bookmark

A mostly personal collection of interesting web page or file references directly accessible in the browser.

booting

The sequence of computer operations from power-up until the system is ready for use.

browser

Program that searches and displays contents. Today it is mostly used for programs that display contents of *world wide web* pages.

client

A program or computer in a networking environment that connects to and requests information from a *server*.

command line

Text-based mode of operating in which commands are entered at a prompt. A command line can be accessed from within a graphical environment as well as from virtual consoles.

console

Formerly synonymous with *terminal*. In Linux, there are several *virtual consoles* that allow the screen to be used for several independent, parallel work sessions.

CPU (Central Processing Unit)

See *processor*.

cursor

The cursor is normally a block character that marks the place for input on a computer screen. This term also often refers to the symbol representing the location of the mouse in graphical interfaces.

daemon

A daemon (disk and execution monitor) is a program that monitors in the background and comes into action when required. Daemons answer FTP or HTTP requests, for example, or control activity in the PCMCIA slots.

DDC (Direct Display Channel)

Communication standard between the monitor and the graphics card, which transmits various parameters, such as monitor name or resolution, to the graphics card.

directory

A structure for organizing files on a computer or network. A directory can contain a collection of files, other directories, or both. Organized in a hierarchical manner, many directories form a file system.

DNS (Domain Name System)

A system that converts name-based addresses to \Leftrightarrow TCP/IP addresses and vice versa.

driver

A program between the operating system and the hardware that *translates* the communication between these two layers.

e-mail (electronic mail)

The means of transporting mail electronically between registered users via a network. As with *normal* mail (often referred to as *snail mail*), the address must be entered. In e-mail, it is in the form *sender@sender's-domain* to *recipient@recipient's-domain*. E-mail not only lets you send text, but also sound files or pictures. It has many advantages: it is inexpensive and mail usually reaches its destination within minutes.

EIDE (Enhanced Integrated Drive Electronics)

Improved \Leftrightarrow IDE standard that allows hard disks with a size over 512 MB.

environment

A *shell* usually provides an environment in which the user can perform temporary settings. These settings include path specifications for programs, the user name, the current path, and the appearance of prompts. The data is saved in an *environment variable*. The assignment of the environment variables is possible, for example, by means of the configuration files of the shell.

environment variable

An element of the *environment* of the *shell*. Every environment variable has a name that is usually capitalized. The variables are assigned values, such as path names.

ethernet

Popular standard for less expansive computer networks.

EXT2 (Second Extended File System)

One of the oldest file systems supported by Linux.

FAQ (Frequently Asked Questions)

Widely-used acronym for documents providing answers to frequently-asked questions.

firewall

Protects a local network or host from unauthorized access over a network using various security measures.

free software

See *GNU*.

FTP (File Transfer Protocol)

A *protocol* based on *TCP/IP* for transferring files.

GNOME (GNU Network Object Model Environment)

A user-friendly graphical desktop environment for Linux.

GNU (GNU is Not Unix)

GNU is a project of the Free Software Foundation (FSF)TM. Closely linked to the *GNU Project* is the name of RICHARD STALLMAN (RMS). The aim

of the GNU Project is to create a Unix-compatible operating system free not so much in the sense of *free of cost*, but in the sense of *freedom*: having the right to obtain, modify, and change the software. To guarantee the freedom of the source code (the actual program code), every change of the original code must be free as well, so modifications of or additions to the original software cannot compromise this freedom in any way. The now classic GNU Manifesto (<http://www.gnu.org/gnu/manifesto.html>) explains many aspects of this thinking. In legal terms, GNU software is protected by the GNU General Public License, or *GPL* (<http://www.gnu.org/copyleft/gpl.html>), and by the GNU Lesser General Public License, or *LGPL* (<http://www.gnu.org/copyleft/lgpl.html>).

In connection with the GNU Project, all Unix tools and utilities are being redeveloped and, in part, provided with more or enhanced functionalities. Even complex software systems, such as EMACS or glibc, are integral components of the project. The *Linux* kernel, subject to the GPL, profits from this project (especially from the tools), but should not be seen as the same thing.

GPL (GNU General Public License)

See *GNU*.

home directory

A private directory in the Linux system that belongs to a specific user (usually in `/home/<username>`). Except the superuser *root*, only the user has full access rights in his home directory.

host name

Name of a machine. In Linux, this is usually the name by which it can be reached on the network.

HTML (Hypertext Markup Language)

The most important language used in the *World Wide Web* for designing the contents. The layout commands made available by HTML define how a document looks and how it is displayed in a *browser*.

HTTP (Hypertext Transfer Protocol)

A protocol used between the *browsers* and Internet servers to transmit *HTML* pages over the *World Wide Web*.

IDE (Integrated Drive Electronics)

A widely-used hard disk standard in low-grade and middle-grade PCs.

Internet

Worldwide computer network based on \Leftrightarrow *TCP/IP*, which is used by a very large population.

IP address

A numerical 32-bit Internet address, appearing in four decimal series separated by periods (for example, 192.168.10.1), which is uniquely assigned to a machine connected to \Leftrightarrow *TCP/IP* networks.

IRQ (Interrupt Request)

A request to the \Leftrightarrow *operating system* carried out by a hardware component or a program to assign it processor capacity.

ISDN (Integrated Services Digital Network)

A popular digital standard for high-speed data transferral over the telephone network.

KDE (K Desktop Environment)

User-friendly graphical desktop environment for Linux.

kernel

The kernel is the central core of the Linux operating system. It manages memory, contains the drivers that enable communication with the hardware, and handles processes and tasks. Applications run on top of the kernel.

LAN (Local Area Network)

A LAN is a local \Leftrightarrow *network* and is usually rather small.

LILO (Linux Loader)

Small program installed in the boot sector of the hard disk that not only can start Linux, but other operating systems as well.

link

A link is a pointer to a file, just as widely used in the Internet as in the Linux file system. In Linux, there is a distinction made between *hard* and *symbolic* links. While *hard* links refer to the exact position in the file system, the symbolic link only points to the respective name.

Linux

High performance UNIX-like operating system core distributed freely under the GPL (⇒*GNU*). The name is an acronym (*Linus' uniX*) and refers to its creator, LINUS TORVALDS. Although the name, in a strict sense, only refers to the kernel itself, the popular understanding of the term *Linux* usually entails the entire system.

login

Authentication of a user by user name and password to gain access to a computer system or network.

logout

The procedure of closing down an interactive Linux session and getting back to the ⇒*login* prompt.

main memory

Physical memory of limited capacity that can be accessed rather quickly. This is often referred to as RAM, random access memory.

man pages

Traditional documentation for Unix systems, which can be read using the command *man*.

MBR (Master Boot Record)

The first physical sector of the hard disk from which the content is loaded to the main memory and executed by the ⇒*BIOS*. This code then loads either the operating system from a hard disk partition or a more sophisticated boot loader, such as ⇒*LILO*.

MD5

Algorithm for generating checksums.

mounting

This describes the *insertion* of file systems into the directory tree of the system.

MP3

Very efficient compression procedure for audio files that reduces the size by a factor of ten in contrast to an uncompressed audio file.

multitasking

Operating systems that can invoke more than one program simultaneously are called multitasking systems.

multiuser

Enables more than one user to work simultaneously on the same system.

network

The interconnection of several computers, accomplished normally using *servers* and *clients*.

NFS (Network File System)

A *protocol* for accessing a file system shared over a network.

NIS (Network Information Service)

A centralized data administration system in networks. User names and passwords can be simultaneously managed network-wide by NIS.

operating system

Program that permanently runs in the background on a computer and enables basic system operations.

partition

Logically-independent section of a hard disk, each possibly containing different file systems. In Windows, also known as *drives*.

path

Unique description of a file's position in a file system.

plug and play

Automatic hardware component configuration technology. Resources, such as IRQ and DMA, are configured and managed separately from the system.

process

In Linux, started programs or executable files run as processes, often referred to as tasks. Processes can be controlled by commands like *top* entered in the *shell*.

processor

The processor is the *brain* of every computer, working through and performing commands given by a user or a program in machine language. The processor has control over the entire system and is responsible for the actual performance of the computer.

prompt

See [☞command line](#).

protocol

Standard specifically defined for regulating communication for hardware, software, or networks. There is a multitude of these standards. The most common examples are [☞HTTP](#) and [☞FTP](#).

proxy

Most commonly used cache implemented by Internet providers that stores frequently requested contents in a database to allow other machines requesting those pages to load them directly from it. This process not only reduces the time it takes to download this information, but also conserves the available bandwidth.

RAM (Random Access Memory)

See [☞main memory](#).

ReiserFS

A file system that logs its changes to a journal. Compared to Ext2, this features allows a file system to be restored very speedily. ReiserFS is optimized for small files.

root

The user undertaking the configuration and maintenance of a complex computer system, such as a network. This system administrator is usually the only person who has access to all parts of the system (root permissions).

root directory

The base directory of the file system that does not have any parent directory (all other directories have a parent directory). In UNIX, the root directory is represented as a `/`.

SCSI (Small Computer Systems Interface)

Hard disk standard implemented in servers and other high-level machines because of its high-speed performance. See *server*.

server

A server is usually a rather powerful computer that offers services, such as HTTP, DNS, and FTP, or data to other machines connected via a network. There are also programs called servers, like the *X server*.

shell

An especially flexible command line often equipped with its own specific programming language. Examples of shells are Bash, sh, and tcsh.

SMTP (Simple Mail Transfer Protocol)

Protocol for transferring *e-mails*.

SSL (Secure Socket Layer)

Encryption procedure for transferring *HTTP* data.

superuser

See *root*.

system administrator

See *root*

task

See *process*.

TCP/IP

Internet communication protocol finding increased use in local networks, known as *intranets*.

telnet

Telnet is the *protocol* and command for communicating with other hosts. Normally, the user only sees telnet as a means for logging into a remote system.

terminal

Formerly, the designation of a keyboard and monitor combination connected to a central computer. On workstations, this term is also used for programs that emulate a real terminal.

Tux

Name of the Linux penguin (see <http://www.sjbaker.org/tux/>).

UNIX

UNIX is an operating system that is widely distributed, above all on workstations in networks. Since the beginning of the 1990s, there has been a freely available version for PCs: Linux.

URL (Uniform Resource Locator)

Unique Internet address that contains the type (e.g., <http://>) and the name of the host (e.g., www.suse.de).

user directory

See [☞home directory](#).

VESA (Video Electronics Standard Association)

Industrial consortium that defines, among other things, important video standards.

wild card

Placeholder for one (symbol: `?`) or more (symbol: `*`) unknown characters, most often used in commands (especially search commands).

window manager

A window manager is the layer that interacts between the [☞X Window System](#) and the user. It is responsible, among other things, for your desktop display. There is a wide variety of window managers available, one of the more popular ones being `kwm` for [☞KDE](#).

WWW (World Wide Web)

Based on the [☞HTTP](#) protocol, this is a hyperlinked collection of documents, files, and images that can be viewed with a web browser.

X Window System

The X Window System is the standard for graphical interfaces in Linux. It is simply the middle layer between the hardware and the [☞window manager](#), such as KDE or GNOME.

X11

See [☞X Window System](#).

YaST (Yet another Setup Tool)

The SUSE LINUX system assistant.

YP

See ↗ *NIS*.

Bibliography

- [1] *SUSE LINUX (User Guide)*. SUSE, 10. 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 Network 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 148
- 64-bit Linux 179
 - kernel specifications 182
 - runtime support 180
 - software development 181

A

- ACLs 621–633
 - access 623, 626
 - check algorithm 632
 - default 623, 629
 - definitions 623
 - effects 629
 - handling 623
 - masks 627
 - permission bits 625
 - structure 624
 - support 632
- ACPI
 - disabling 10
- addresses
 - IP 400
 - MAC 400
- ADSL
 - configuring 566–567
 - dial-on-demand 566
- Apache 143, 507–530
 - apxs 512
 - CGI 520
 - configuring 513–518
 - content negotiation 510
 - default page 509

- DocumentRoot 514
 - flags 514
 - installing 511–513
 - logging 517, 518
 - modules 510
 - activating 513
 - loading 515
 - mod_perl 521
 - mod_php4 523
 - mod_python 524
 - mod_ruby 524
 - permissions 515, 528
 - security 527–528
 - Squid 581
 - SSI 517, 520
 - starting 512
 - threads 511
 - troubleshooting 528
 - virtual hosts 510, 524–527
- authentication
 - PAM 385–392

B

- backups 57
 - creating with YaST 88
 - restoring 88
- Bash
 - .bashrc 216
 - .profile 216
 - profile 216
- BIND 438–447
- BIOS
 - boot sequence 8

- virus protection 114
- Bluetooth 291, 346
 - hciconfig 351
 - hcitool 350
 - network 348
 - opd 353
 - pand 352
 - sdptool 351
- boot disks 188
 - CDs 188
 - creating
 - DOS 116
 - creating with dd 118
 - creating with rawrite 117
- booting 229, 659, 663
 - boot loaders 232
 - boot managers 188
 - boot sectors 186, 187
 - CD, from 8
 - CD 2, from 119
 - configuring 29
 - YaST 198–200
 - DOS 187
 - floppy disks, from 116, 119
 - graphic 114
 - disabling 114
 - GRUB 114, 185, 189–204
 - initrd 230–234
 - creating 233
 - LILO 114
 - linuxrc 231
 - loader 199
 - locations for 200
 - log 94
 - management 187
 - methods 114
 - process 186
 - system freezes 114
 - USB sticks 188
 - Windows 187

C

- cards
 - graphics 67
 - drivers 252
 - network 423
 - testing 423
 - radio 78
 - sound 77
 - TV 78

- CD-ROM drives
 - ATAPI 120
 - supported 120
- CDs
 - booting from 8, 188
- cellular phones 293
- chown 148
- CJK 225
- coldplug 365
- commands
 - chown 148
 - e2fsck 663
 - fonts-config 253
 - free 220
 - getfacl 627
 - grub 189
 - head 148
 - hotplug 361
 - hwinfo 364
 - ldapadd 475
 - ldapdelete 477
 - ldapmodify 476
 - ldapsearch 477
 - lp 63
 - nice 148
 - rpm 156
 - rpmbuild 156
 - scp 599
 - setfacl 627
 - sftp 599
 - slptool 437
 - smbpasswd 558
 - sort 148
 - ssh 598
 - ssh-agent 602
 - ssh-keygen 601
 - tail 148
 - udev 369
- configuration files 416
 - .bashrc 216, 219
 - .emacs 221
 - .mailsync 547
 - .profile 216
 - .xsession 602
 - acpi 317
 - apache2 513
 - asound.conf 78
 - config 207
 - crontab 216
 - csh.cshrc 227

- dhclient.conf	492
- dhcp	416
- dhcpd.conf	492
- exports	488, 490, 579
- foomatic/filter.conf	145
- fstab	25, 126, 174
- group	141
- grub.conf	195
- gshadow	149
- host.conf	418
· alert	419
· multi	419
· nospoof	419
· order	419
· trim	419
- HOSTNAME	421
- hosts	403, 418
- hotplug	360
- httpd.conf	513, 514
- hwinfo	364
- hwup	362
- ifcfg-*	416
- inittab	224, 235, 237
- inputrc	225
- irda	356
- kernel	233
- language	225, 227
- lilo.conf	232
- logrotate.conf	217
- menu.lst	190, 232
- modprobe.conf	78, 146, 210
- modules.conf	146
- modules.dep	210
- named.conf	438, 440–447, 573
- network	416
· providers	566
- networks	418
- nscd.conf	421
- nsswitch.conf	419, 478
- pam_unix2.conf	478
- passwd	141
- permissions	615
- powersave	317
- powersave.conf	154
- profile	216, 219, 227
- rc.config	243
- resolv.conf	220, 417, 438, 571
- routes	434
- samba	557
- services	557

- slapd.conf	469
- smb.conf	552, 553
- smppd.conf	564
- smpppd-c.conf	565
- squid.conf	572, 573, 576, 578, 581, 583
- squidguard.conf	583
- sshd_config	602
- suseconfig	244
- sysconfig	93, 243–244
- syslinux.cfg	232
- termcap	225
- wireless	416
- XF86Config	248
· Device	251
· Monitor	252
· Screen	250
- xml/catalog	145
- xml/suse-catalog.xml	145
configuring	243
- Apache	513–518
- cable modem	425
- CD-ROM	59
- DNS	80, 438
- DSL	428
- e-mail	81
- firewalls	87
- graphics cards	67
- groups	84
- GRUB	189, 195
- hard disk controllers	64
- hard disks	
· DMA	74
- hardware	58–79
- IPv6	434
- IrDA	356
- ISDN	430
- joysticks	75
- keyboard	75
- language	93
- laptops	298–303
- modems	426
- mouse	75
- networks	79–83, 423
· manually	412–435
- NFS	81
- NTP	
· client	82
- printing	59–64
- radio	78
- routing	83, 434

- Samba 553
 - clients 83, 560
 - servers 83
- scanner 75
- security 83–88
- software 46–57
- sound cards 77
- Squid 573
- SSH 598
- system 43–95
- system services 82
- T-DSL 430
- time zone 93
- TV 78
- users 83
- X 64
- consoles
 - assigning 224
 - graphical
 - disabling 114
 - switching 224
- core files 219
- cpuspeed 324
- crashes 659, 663
- cron 216
- crypto file system 603
- CVS 533, 539–542

D

- data security 291
- depmod 210
- device nodes
 - udev 369
- DHCP 491–500
 - configuring with YaST 496
 - dhcpd 492–494
 - packages 492
 - server 492–494
 - static address assignment 494
- digital cameras 292
- disks
 - boot 89
 - creating 201
 - floppy
 - formatting 118
 - module 89
 - rescue 89
- DNS 403
 - BIND 438–447
 - configuring 80, 438

- domains 417
- forwarding 439
- logging 442
- mail exchanger 404
- name servers 417
- NIC 403
- options 441
- reverse lookup 446
- security and 614
- Squid and 573
- starting 439
- top level domain 403
- troubleshooting 439
- zones
 - files 444
- domain name system *see* DNS
- DOS
 - sharing files 551

E

- e-mail
 - configuring 81
 - synchronization 289
 - synchronizing 533
 - mailsync 547–550
- e2fsck 663
- editors
 - Emacs 221–222
 - vi 222
- Emacs 221–222
 - .emacs 221
 - default.el 221
- encoding
 - ISO-8859-1 227
 - UTF-8 148
- encryption
 - files 603
 - partitions 603
- error messages
 - bad interpreter 25
 - permission denied 25
- Evolution 293

F

- FAT file system 23
- file servers 81
- file systems 376–384
 - ACLs 622–633
 - e2fsck 663
 - encrypting 603

- Ext2 377
- Ext3 378–379
- FAT 23
- JFS 380
- LFS 383
- limitations 383
- NTFS 23, 25
- ReiserFS 379–380
- reiserfsck 659
- repairing 175
- selecting 376
- supported 382
- sysfs 360
- terms 376
- XFS 381
- files
 - encrypting 603
 - finding 218
 - synchronizing 531–550
 - CVS 533, 539–542
 - mailsync 533, 547–550
 - rsync 534
 - subversion 533
 - Unison 532, 538–539
- firewalls 87, 588
 - packet filters 588, 591
 - Squid and 579
 - SuSEfirewall2 588, 592
- Firewire (IEEE1394)
 - hard disks 292
- flash drives 292
 - booting from 188
- floppy disks
 - booting from 188
- fonts 254
 - CID-keyed 258
 - TrueType 253
 - X11 core 257
 - Xft 254
- FTP
 - servers 143

G

- GPL 669
- graphical user interface 64–73
- graphics
 - 3D 258–261
 - 3Ddiag 260
 - diagnosis 260
 - drivers 258

- installation support for 260
- SaX 259
- support for 258
- testing 260
- troubleshooting 260
- cards
 - 3D 258–261
 - drivers 252
- GLIDE 258–261
- OpenGL 258–261
 - drivers 258
 - testing 260
- groups
 - administering 84
 - name changes 142
- GRUB 185–204
 - boot management 187
 - boot menu 190
 - boot password 196
 - boot process 186
 - boot sectors 187
 - booting 189
 - commands 189–197
 - device names 191
 - device.map 189, 194
 - GRUB Geom Error 203
 - GRUB shell 196
 - grub.conf 189, 195
 - JFS and GRUB 203
 - limitations 188
 - Master Boot Record (MBR) 186
 - menu editor 193
 - menu.lst 189, 190
 - partition names 191
 - troubleshooting 202
 - uninstalling 201

H

- hard disks
 - DMA 74
 - parallel use of 124
- harden_suse 143
- hardware
 - CD-ROM 59
 - ATAPI 120
 - hard disk controllers 64
 - information 74
 - ISDN 430
 - SCSI devices 121
- hciconfig 351

hcitool	350
head	148
help	
- info pages	218
- man pages	218
- X	253
host names	80
hotplug	359–366, 433
- agent	361, 362
· devices	362
· interfaces	362
· PCI	364
· USB	364
- blacklist	364
- device names	360
- error analysis	366
- event recorder	366
- events	361
- log files	366
- map files	364
- modules	
· automatic loading of	364
- network devices	363
- PCI	365
- storage devices	363
- whitelist	364
hwinfo	364

I

I18N	225
inetd	82, 144
info pages	218
init	235
- adding scripts	240
- inittab	235
- scripts	238–241
input method	
- CJK	225
insmod	209
installation support	
- 3D graphics cards and	260
installing	
- boot loader	114
- GRUB	189
- network, from	116
- packages	157
- text mode	112–113
- VNC	111
- YaST	7–40
internationalization	225

Internet

- cinternet	565
- dial-up	564–565
- DSL	428
- ISDN	430
- kinternet	565
- qinternet	565
- smpppd	564–565
- TDSL	430
- web servers	<i>see</i> Apache

IP addresses

- classes	400
- dynamic assignment	491
- IPv6	404
· configuring	434
- masquerading	590
- private	402
IrDA	291, 355–357
- configuring	356
- starting	356
- stopping	356
- troubleshooting	357

J

jade	<i>see</i> SGML, openjade
jade_dsl	144
joysticks	
- configuring	75

K

kernels	206–213
- caches	220
- compiling	206, 211
- configuring	207–211
- daemon	210
- error messages	211
- installing	212
- limits	384
- modprobe.conf	210
- module loader	210
- modules	208–211
· compiling	211
· modprobe.conf	146
· network cards	423
- parameters	206
- problems	234
- sources	207
- version 2.6	145
keyboard	
- Asian characters	225

- configuring	75
- layout	224
- mapping	224
· compose	225
· multikey	225
- X Keyboard Extension	225
- XKB	225
Kmod	<i>see</i> kernels, module loader
Kontakt	293
KPilot	293
KPowersave	289
KSysguard	289

L

L10N	225
languages	93
laptops	286–291
- hardware	286
- IrDA	355–357
- PCMCIA	286, 433
- power management	286, 313–324
- SCPM	287, 305
- SLP	288
LDAP	464–486
- access control	472
- ACLs	470
- adding data	473
- administering groups	484
- administering users	484
- deleting data	477
- directory tree	466
- ldapadd	473
- ldapdelete	477
- ldapmodify	476
- ldapsearch	477
- modifying data	476
- searching data	477
- server configuration	469
- YaST	
· modules	479
· templates	479
- YaST LDAP client	477
LFS	383
license	<i>see</i> GPL
Lightweight Directory Access Protocol	<i>see</i> LDAP
LILO	
- configuring	114
Linux	
- networks and	395

- sharing files with another OS	551
- uninstalling	201
linuxrc	102
linuxthreads	146
Local Area Networks	<i>see</i> networks, LANs
locale	
- UTF-8	148
localization	225
locate	218
log files	217
- apache2	518, 529
- boot.msg	94, 317
- httpd	516, 518, 529
- log	86
- messages	94, 439, 597
- Squid	572, 574, 580
- Unison	539
- XFree86	260

logging

- login attempts	86
- logrotate	
· configuring	217
Logical Volume Manager	<i>see</i> LVM
logrotate	217
LSB	
- installing packages	156
lsmmod	210
LVM	
- YaST	127

M

man pages	218
masquerading	590
- configuring with SuSEfirewall2	592
Master Boot Record	<i>see</i> MBR
MBR	186, 187
memory	
- RAM	220
mobility	285–294
- cellular phones	293
- data security	291
- digital cameras	292
- external hard disks	292
- Firewire (IEEE1394)	292
- laptops	286
- PDAs	293
- USB	292
modems	
- cable	425
- YaST	426

modinfo 210
 modprobe 210
 modules
 - loading 105
 - parameters 105
 monitor settings 64
 mountd 490
 mouse
 - configuring 75
 multicast DNS 148

N

name servers *see* DNS
 NAT *see* masquerading
 NetBIOS 553
 Network File System *see* NFS
 Network Information Service *see* NIS
 networks 395
 - base network address 402
 - Bluetooth 291, 348
 - broadcast address 402
 - configuration files 416–421
 - configuring 79–83, 412–435
 - IPv6 434
 - DHCP 491
 - DNS 403
 - integrating 422–435
 - IrDA 291
 - LANs 422–435
 - localhost 402
 - netmasks 401
 - routing 83, 400, 401
 - SLP 435
 - TCP/IP 396
 - wireless 290
 - WLAN 290
 - YaST 423
 NFS 486
 - clients 81, 486
 - exporting 488
 - importing 486
 - mounting 486
 - permissions 489
 - servers 81, 487
 nfsd 490
 NGPT 146
 nice 148
 NIS 459–463
 - clients 463
 - masters 460–462

 - slaves 460–462
 notebooks *see* laptops
 NPTEL 146, 147
 NSS 419
 - databases 420
 NTFS file system 23
 NTP
 - client 82
 nVidia 144

O

opd 353
 OpenSSH *see* SSH
 OS/2
 - sharing files 551

P

packages
 - building 145
 - compiling 163
 - compiling with build 164
 - installing 157
 - LSB 156
 - package manager 156
 - RPMs 156
 - uninstalling 157
 - verifying 156
 packet filters *see* firewalls
 PAM 385–392
 pand 352
 partitions
 - creating 15, 19, 20
 - encryption 603
 - expert 122
 - fstab 25
 - LVM 21
 - optimizing 123
 - parameters 20
 - partition table 186
 - RAID 21
 - resizing Windows 22
 - swap 21, 122
 - types 15
 - YaST 126
 PCMCIA 286, 296, 433
 - card manager 297
 - configuring 298
 - IrDA 355–357
 - ISDN 299
 - modem 299

- network cards	298	- Samba	553
- SCSI	299	- supported printers	60
- troubleshooting	300	- test page	62
- utilities	300	- troubleshooting	64
PDA's	293	· network	278
permissions		- xpp	63
- ACLs	622–633	protocols	
- file permissions	218	- FTP	508
phone exchange	432	- HTTP	508
Pluggable Authentication Modules	<i>see</i> PAM	- HTTPS	508
ports		- IPv6	404
- 53	441	- LDAP	464
- scanning	580	- SLP	435
PostgreSQL		- SMB	552
- updating	141	proxies	<i>see</i> Squid
power management	286, 313–331	- advantages	568
- ACPI	313, 316–322, 327	- caches	568
- APM	313, 315–327	- transparent	578
- battery monitor	315	R	
- charge level	328	RAID	
- cpufreqency	324	- soft	135
- cpuspeed	324	reiserfsck	659
- hibernation	315	removable media	
- powersave	324	- subfs	151
- standby	314	repairing systems	167
- suspend	314	rescue system	172
- YaST	332	- starting	172
powersave	324	- using	173
- configuring	325	resolver library	
printing	59–64, 263	- .local as top-level domain	148
- applications, from	63	RFCs	396
- command line	63	rmmod	209
- configuring with YaST	61	routing	83, 400, 434–435
- connection	61	- masquerading	590
- CUPS	63	- netmasks	401
- drivers	62	- routes	434
- foomatic-filters	145	- static	434
- GDI printers	60, 276	RPM	156–165
- Ghostscript driver	62	- database	
- IrDA	356	· rebuilding	158, 162
- kprinter	63	- dependencies	157
- LPRng	145	- patches	158
- network		- queries	160
· troubleshooting	278	- rpmnew	157
- port	61	- rpmorig	157
- PPD file	62	- rpmsave	157
- printer languages	59	- security	616
- problems	64	- SRPMS	163
- processing a print job	60	- tools	165
- queues	61		

- uninstalling 158
- updating 157
- verify 162
- verifying 156
- version 145
- rpmbuild 145, 156
- rsync 534, 545
- runlevels 92, 235–237
 - changing 237
 - editing in YaST 241
 - switching 92

S

- Samba 551–562
 - clients 83, 553, 560–562
 - configuring 553
 - help 562
 - installing 553
 - login 557
 - names 553
 - optimizing 562
 - permissions 556
 - printers 553
 - printing 561
 - security 556–557
 - servers 83, 553
 - shares 553, 555
 - SMB 552
 - starting 553
 - stopping 553
 - swat 557
 - TCP/IP and 552
- SaX 64
 - multihead 70
- scanning
 - configuring 75
 - troubleshooting 76
- SCPM 92, 305
 - advanced settings 309
 - configuring 307
 - laptops 287
 - managing profiles 308
 - resource groups 307
 - starting 307
 - switching profiles 308
- screen
 - resolution 251
- scripts
 - boot.udev 373
 - init.d 235, 238–241, 422

- boot 239
- boot.local 239
- boot.setup 239
- halt 239
- network 422
- nfsserver 422, 488
- portmap 422, 488
- rc 237–239
- sendmail 422
- squid 572
- xinetd 422
- ypbind 422
- ypserv 422
- irda 356
- mkinitrd 233
- modify_resolvconf 220, 417
- SuSEconfig 243–244
 - disabling 244
- SCSI devices
 - configuring 121
 - file names, assigning 121
- sdptool 351
- security 605–617
 - attacks 613–614
 - booting 606–608
 - bugs and 609, 612
 - configuring 83–88
 - DNS 614
 - encrypted file system 291
 - engineering 606
 - firewalls 87, 588
 - local 607–611
 - network 611–614
 - passwords 607–608
 - permissions 609
 - reporting problems 617
 - RPM signatures 616
 - Samba 556
 - serial terminals 606, 607
 - Squid 568
 - SSH 598–603
 - tcpd 617
 - tips and tricks 615
 - viruses 610
 - worms 614
 - X and 611
- Service Location Protocol *see* SLP
- SGML
 - directories 151
 - openjade 144

SLP	288, 435	- key pairs	600, 601
- browser	437	- scp	599
- Konqueror	437	- sftp	599
- registering services	436	- ssh	598
- slptool	437	- ssh-agent	602
SMB	<i>see</i> Samba	- ssh-keygen	601
software		- sshd	600
- compiling	163	- X and	602
- installing	48–54		
- removing	48–54	subfs	
sort	148	- removable media	151
sound		subversion	533, 542
- configuring in YaST	77	support	93
- fonts	78	SUSE LINUX	
- mixers	155	- installation	102
source		sx	144
- compiling	163	synchronizing data	290
spm	163	- e-mail	289
Squid	567	- Evolution	293
- access controls	581	- Kontakt	293
- ACLs	576	- KPilot	293
- Apache	581	system	
- cachemgr.cgi	581, 582	- configuring	43–95
- caches	568, 569	- freezes	114
size	570	- information	103
- Calamaris	584	- language	93
- configuring	573	- limiting resource use	219
- CPU and	571	- localizing	225
- damaged cache	572	- rescuing	172
- directories	572	- security	85
- DNS	573	- updating	56, 139–145, 165
- features	568	system monitoring	289
- firewalls and	579	- KPowersave	289
- log files	572, 574, 580	- KSysguard	289
- object status	569	system services	82
- permissions	572, 576		
- RAM and	571	T	
- reports	584	T-DSL	<i>see</i> ADSL
- security	568	tail	148
- squidGuard	582	TCP/IP	396
- starting	571	- ICMP	397
- statistics	581, 582	- IGMP	397
- stopping	572	- layer model	397
- system requirements	570	- packets	397, 399
- transparent proxies	578, 580	- services	396
- troubleshooting	572	- TCP	396
- uninstalling	572	- UDP	397
SSH	598–603	thread packages	
- authentication mechanisms	601	- NPTL	147
- daemon	600	time zones	93
		TV	

- card configuration 78
- U**
- udev 369
 - automization 371
 - keys 372
 - mass storage 373
 - regular expressions 371
 - rules 370
 - start script 373
 - sysfs 372
 - udevinfo 372
 - YaST 374
- ulimit 219
 - options 219
- uninstalling
 - GRUB 201
 - Linux 201
- updating 139–145, 165
 - online 46–48
 - passwd and group 141
 - problems 141
 - sound mixers 155
 - YaST 141
- USB
 - flash drives 292
 - hard disks 292
- users
 - /etc/passwd 388, 479
 - administering with YaST 83
 - name changes 142
- UTF-8
 - encoding 148
- V**
- variables
 - environment 225
- virtual consoles
 - switching 92
- virtual memory 21
- VNC
 - installation 111
- W**
- web servers 143
 - Apache *see* Apache
- whois 404
- Windows
 - sharing files 551
- wireless connections

- Bluetooth 346
- WLAN 290

- X**
- X 247
 - 3D 69
 - character sets 253
 - CID-keyed fonts 258
 - configuring 64
 - drivers 252
 - font systems 254
 - fonts 253
 - help 253
 - multihead 70
 - optimizing 248
 - SaX2 248
 - security 611
 - SSH and 602
 - TrueType fonts 253
 - virtual screen 251
 - X11 core fonts 257
 - xf86config 248
 - Xft 254
 - xft 253
- X Keyboard Extension *see* keyboard, X Keyboard Extension
- X Window System *see* X
- X.Org 248
- XF86Config
 - color depth 251
 - Depth 251
 - Device 251
 - Display 251
 - Files 249
 - InputDevice 249
 - Modeline 251
 - modelines 249
 - Modes 249, 251
 - Monitor 249, 251
 - ServerFlags 249
- Xft 254
- xinetd 144
- XKB *see* keyboard, X Keyboard Extension
- XML
 - catalog 145
 - directories 151
 - openjade 144

- Y**
- YaST

- 3D	259	- NFS server	81
- backups	57, 88	- NIS clients	35, 463
- boot configuration	198	- NTP	
- boot mode	29	· client	82
- cable modem	425	- online update	46–48, 98
- CD-ROM	59	- package dependencies	28
- configuring	43–95	- package manager	49
- Control Center	45	- partitioning	15, 19, 126
- DHCP	496	- power management	332
- disk creation	89	- printing	59–64
- disk space	16	- profile manager	92
- DMA	74	- radio cards	78
- DNS	80	- RAID	135
- driver CDs	95	- repairing systems	167
- DSL	428	- root password	31
- e-mail	81	- routing	83
- firewall	87	- runlevels	241
- graphical user interface	64–73	- safe settings	10
- graphics card	64	- Samba	
- graphics cards	67	· clients	83, 560
- group administration	84	· servers	83
- hard disk controllers	64	- scanner	75
- hardware	58–79	- SCPM	92
- hardware information	74	- security	83–88
- host name	80	- sendmail	81
- installation mode	12	- SLP browser	437
- installation scope	26	- software	46–57
- installation sources	46	- software updates	34
- installation suggestion	13	- sound cards	77
- installing with	7–40	- starting	8, 44
- ISDN	430	- support request	93
- joysticks	75	- sysconfig editor	93, 245
- keyboard layout	14, 75	- system security	85
- language	93	- system start-up	8
- language selection	12	- T-DSL	430
- LDAP client	477	- text mode	95–100, 112–113
- LVM	91, 127	· modules	98
- manual installation	11	· troubleshooting	113
- modem	426	- time zone	93
- monitor settings	64	- TV cards	78
- mouse	14, 75	- updating	56, 141
- ncurses	95	- user administration	83
- network card	423	- YOU	46–48
- network configuration	32, 79–83	YP	<i>see</i> NIS
- NFS client	81		