

openSUSE

12.3

www.suse.com

March 01, 2013

Reference



Reference

Copyright © 2006–2013 Novell, Inc. and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled “GNU Free Documentation License”.

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

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

Contents

About This Guide xiii

1 Available Documentation	xiv
2 Feedback	xv
3 Documentation Conventions	xv
4 About the Making of This Manual	xvi
5 Source Code	xvi
6 Acknowledgments	xvi

I Installation and Deployment 1

1 Installation with YaST 3

1.1 Choosing the Installation Media	3
1.2 Choosing the Installation Method	5
1.3 The Installation Workflow	9
1.4 System Start-Up for Installation	9
1.5 The Boot Screen	9
1.6 Welcome	12
1.7 Installation Mode	13
1.8 Clock and Time Zone	16
1.9 Desktop Selection	17
1.10 Suggested Partitioning	17
1.11 Create New User	20

1.12 Installation Settings	24
1.13 Performing the Installation	28
1.14 Configuration of the Installed System	29
1.15 Graphical Login	34

2 Remote Installation 35

2.1 Installation Scenarios for Remote Installation	35
2.2 Setting Up the Server Holding the Installation Sources	44
2.3 Preparing the Boot of the Target System	53
2.4 Booting the Target System for Installation	62
2.5 Monitoring the Installation Process	66

3 Advanced Disk Setup 71

3.1 Using the YaST Partitioner	71
3.2 LVM Configuration	82
3.3 Soft RAID Configuration	87

II Advanced Administration 93

4 Snapshots/Rollback with Snapper 95

4.1 Requirements	95
4.2 Using Snapper to Undo YaST and zypper Changes	96
4.3 Using Snapper to Restore Files from Hourly Backups	102
4.4 Managing Snapshots	102
4.5 Disabling Snapper	105
4.6 Creating Snapper Configurations	106
4.7 Limitations	108
4.8 Frequently Asked Questions	110

5 Remote Access with VNC 111

5.1 One-time VNC Sessions	111
5.2 Persistent VNC Sessions	114

III System 117

6 32-Bit and 64-Bit Applications in a 64-Bit System Environment 119

6.1 Runtime Support	119
6.2 Software Development	120
6.3 Software Compilation on Biarch Platforms	120
6.4 Kernel Specifications	121

7 Booting a Linux System 123

7.1 The Linux Boot Process	123
----------------------------------	-----

8 The `systemd` daemon 129

8.1 Basic Usage	130
8.2 System Start and Target Management	137
8.3 Advanced Usage	144
8.4 More information	147

9 The Boot Loader GRUB 149

9.1 Booting with GRUB	150
9.2 Configuring the Boot Loader with YaST	160
9.3 Uninstalling the Linux Boot Loader	165
9.4 Creating Boot CDs	166
9.5 The Graphical SUSE Screen	167
9.6 Troubleshooting	168
9.7 For More Information	169

10 The Boot Loader GRUB2 171

10.1 Main Differences from GRUB Legacy	171
10.2 Configuration File Structure	172
10.3 Configuring the Boot Loader with YaST	180
10.4 For More Information	185

11 Special System Features 187

11.1 Information about Special Software Packages	187
11.2 Virtual Consoles	194
11.3 Keyboard Mapping	194
11.4 Language and Country-Specific Settings	195

12 Dynamic Kernel Device Management with udev 199

12.1 The /dev Directory	199
12.2 Kernel uevents and udev	200
12.3 Drivers, Kernel Modules and Devices	200
12.4 Booting and Initial Device Setup	201
12.5 Monitoring the Running udev Daemon	201
12.6 Influencing Kernel Device Event Handling with udev Rules	203
12.7 Persistent Device Naming	209
12.8 Files used by udev	210
12.9 For More Information	210

IV Services 213

13 Basic Networking 215

13.1 IP Addresses and Routing	218
13.2 IPv6—The Next Generation Internet	221
13.3 Name Resolution	231

13.4 Configuring a Network Connection with YaST	232
13.5 NetworkManager	253
13.6 Configuring a Network Connection Manually	256
13.7 smpppd as Dial-up Assistant	271

14 SLP Services in the Network 275

14.1 Installation	275
14.2 Activating SLP	276
14.3 SLP Front-Ends in openSUSE	276
14.4 Installation over SLP	276
14.5 Providing Services via SLP	277
14.6 For More Information	278

15 The Domain Name System 279

15.1 DNS Terminology	279
15.2 Installation	280
15.3 Configuration with YaST	280
15.4 Starting the BIND Name Server	290
15.5 The /etc/named.conf Configuration File	292
15.6 Zone Files	296
15.7 Dynamic Update of Zone Data	300
15.8 Secure Transactions	300
15.9 DNS Security	302
15.10 For More Information	302

16 DHCP 303

16.1 Configuring a DHCP Server with YaST	304
16.2 DHCP Software Packages	307
16.3 The DHCP Server dhcpd	308

16.4 For More Information	311
17 Time Synchronization with NTP	313
17.1 Configuring an NTP Client with YaST	313
17.2 Manually Configuring ntp in the Network	318
17.3 Dynamic Time Synchronization at Runtime	318
17.4 Setting Up a Local Reference Clock	319
18 Sharing File Systems with NFS	321
18.1 Terminology	321
18.2 Installing NFS Server	322
18.3 Configuring NFS Server	322
18.4 Configuring Clients	331
18.5 For More Information	334
19 Samba	335
19.1 Terminology	335
19.2 Installing a Samba Server	337
19.3 Starting and Stopping Samba	337
19.4 Configuring a Samba Server	337
19.5 Configuring Clients	344
19.6 Samba as Login Server	344
19.7 For More Information	345
20 The Apache HTTP Server	347
20.1 Quick Start	347
20.2 Configuring Apache	349
20.3 Starting and Stopping Apache	364
20.4 Installing, Activating, and Configuring Modules	366
20.5 Getting CGI Scripts to Work	374

20.6 Setting Up a Secure Web Server with SSL	377
20.7 Avoiding Security Problems	384
20.8 Troubleshooting	385
20.9 For More Information	386

21 Setting up an FTP server with YaST **389**

21.1 Starting the FTP server	390
21.2 FTP General Settings	391
21.3 FTP Performance Settings	392
21.4 Authentication	392
21.5 Expert Settings	393
21.6 For more information	393

V Mobility **395**

22 Mobile Computing with Linux **397**

22.1 Laptops	397
22.2 Mobile Hardware	404
22.3 Cellular Phones and PDAs	405
22.4 For More Information	405

23 Power Management **407**

23.1 Power Saving Functions	407
23.2 Advanced Configuration and Power Interface (ACPI)	408
23.3 Rest for the Hard Disk	411
23.4 Troubleshooting	413
23.5 For More Information	414

24 Wireless LAN **417**

24.1 WLAN Standards	417
---------------------------	-----

24.2 Operating Modes	418
24.3 Authentication	419
24.4 Encryption	421
24.5 Configuration with YaST	421
24.6 Tips and Tricks for Setting Up a WLAN	430
24.7 Troubleshooting	431
24.8 For More Information	433

25 Using NetworkManager 435

25.1 Use Cases for NetworkManager	435
25.2 Enabling or Disabling NetworkManager	436
25.3 Configuring Network Connections	437
25.4 Using the KDE NetworkManager Front-End	441
25.5 Using GNOME NetworkManager	444
25.6 NetworkManager and VPN	447
25.7 NetworkManager and Security	448
25.8 Frequently Asked Questions	449
25.9 Troubleshooting	451
25.10 For More Information	452

26 Using Tablet PCs 455

26.1 Installing Tablet PC Packages	456
26.2 Configuring Your Tablet Device	456
26.3 Using the Virtual Keyboard	457
26.4 Rotating Your Display	457
26.5 Using Gesture Recognition	458
26.6 Taking Notes and Sketching with the Pen	460
26.7 Troubleshooting	462
26.8 For More Information	463

27 Copying and Sharing Files 465

27.1 Scenarios	466
27.2 Access Methods	467
27.3 Accessing Files Using a Direct Connection	468
27.4 Accessing Files on Different OS on the Same Computer	469
27.5 Copying Files between Linux Computers	471
27.6 Copying Files between Linux and Windows Computers with SSH	477
27.7 Sharing Files between Linux Computers	478
27.8 Sharing Files between Linux and Windows with Samba	482
27.9 For More Information	484

A An Example Network 485

B GNU Licenses 487

B.1 GNU General Public License	487
B.2 GNU Free Documentation License	490

About This Guide

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

Advanced Deployment Scenarios

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

Managing and Updating Software

Understand how to install or remove software with either YaST or using the command line, how to use the 1-Click Install feature, and how to keep your system up-to-date.

Administration

Learn how to configure and upgrade openSUSE, how to administrate your system in text mode, and get to know some important utilities for Linux administrators.

System

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

Services

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

Mobility

Get an introduction to mobile computing with openSUSE, get to know the various options for wireless computing and power management.

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

For an overview of the documentation available for your product and the latest documentation updates, refer to <http://doc.opensuse.org/> or to the following section.

1 Available Documentation

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

Start-Up (↑Start-Up)

Guides you step-by-step through the installation of openSUSE from DVD, or from an ISO image, gives short introductions to the GNOME and KDE desktops including some key applications running on it. Also gives an overview of LibreOffice and its modules for writing texts, working with spreadsheets, or creating graphics and presentations.

Reference (page i)

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

Security Guide (↑Security Guide)

Introduces basic concepts of system security, covering both local and network security aspects. Shows how to make use of the product inherent security software like AppArmor (which lets you specify per program which files the program may read, write, and execute) or the auditing system that reliably collects information about any security-relevant events.

System Analysis and Tuning Guide (↑System Analysis and Tuning Guide)

An administrator's guide for problem detection, resolution and optimization. Find how to inspect and optimize your system by means of monitoring tools and how to efficiently manage resources. Also contains an overview of common problems and solutions and of additional help and documentation resources.

Virtualization with KVM (↑Virtualization with KVM)

This manual offers an introduction to setting up and managing virtualization with KVM (Kernel-based Virtual Machine) on openSUSE. Also shows how to manage VM Guests with `libvirt` and `QEMU`.

Find HTML versions of most product manuals in your installed system under `/usr/share/doc/manual` or in the help centers of your desktop. Find the latest docu-

mentation updates at <http://www.suse.com/documentation> where you can download PDF or HTML versions of the manuals for your product.

2 Feedback

Several feedback channels are available:

Bugs and Enhancement Requests

To report bugs for a product component, or to submit enhancement requests, please use <https://bugzilla.novell.com/>. For documentation bugs, submit a bug report for the component *Documentation* of the respective product.

If you are new to Bugzilla, you might find the following articles helpful:

- http://en.opensuse.org/openSUSE:Submitting_bug_reports
- http://en.opensuse.org/openSUSE:Bug_reporting_FAQ

User Comments

We want to hear your comments about and suggestions for this manual and the other documentation included with this product. Use the User Comments feature at the bottom of each page in the online documentation or go to <http://www.suse.com/documentation/feedback.html> and enter your comments there.

Mail

For feedback on the documentation of this product, you can also send a mail to doc-team@suse.de. Make sure to include the document title, the product version and the publication date of the documentation. To report errors or suggest enhancements, provide a concise description of the problem and refer to the respective section number and page (or URL).

3 Documentation Conventions

The following typographical conventions are used in this manual:

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

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

4 About the Making of This Manual

This book is written in Novdoc, a subset of DocBook (see <http://www.docbook.org>). The XML source files were validated by `xmllint`, processed by `xsltproc`, and converted into XSL-FO using a customized version of Norman Walsh's stylesheets. The final PDF is formatted through XEP from RenderX. The open source tools and the environment used to build this manual are available in the package `daps` that is shipped with openSUSE. The project's home page can be found at <http://daps.sf.net/>.

5 Source Code

The source code of openSUSE is publicly available. Refer to http://en.opensuse.org/Source_code for download links and more information.

6 Acknowledgments

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

Have a lot of fun!

Your SUSE Team

Part I. Installation and Deployment

Installation with YaST

Install your openSUSE® system with YaST, the central tool for installation and configuration of your system. YaST guides you through the installation process and the basic configuration of your system. During the installation and configuration process, YaST analyzes both your current system settings and your hardware components and proposes installation settings based on this analysis. By default, YaST displays an overview of all installation steps on the left hand side of the window and provides online help texts for each step. Click *Help* to view the help text.

If you are a first-time user of openSUSE, you might want to follow the default YaST proposals in most parts, but you can also adjust the settings as described here to fine-tune your system according to your preferences. Many parts of the basic system configuration, such as user accounts or system language, can also be modified after the installation process.

1.1 Choosing the Installation Media

When installing openSUSE, choose from several media available either online or in the retail box:

DVD-retail

One DVD containing the openSUSE distribution for 32bit and 64bit systems. The second medium contains proprietary add-on software.

This installation option does not require any network access for installation, nor do you need to set up external repositories to install the full openSUSE. You can,

however, make the contents of the DVD available on an installation server and make them available all across your network.

DVD-download

One DVD5, available via download for 32bit or 64bit systems.

Choose this installation option if you want a fully-fledged openSUSE system. Beyond the downloading of the DVD ISO, there is no network connection required to make use of this installation option. Once the medium has been fully downloaded and the physical medium created, you can go ahead with the installation. You can also make the contents of the DVDs available on an installation server and make them available all across your network.

KDE4/GNOME LiveCD

The LiveCD versions, available via download, include the KDE4 or GNOME desktops together with the most popular applications for 32-bit or 64-bit systems.

Choose this medium option for a first look at openSUSE. The LiveCD version runs on your computer using RAM without touching your hard drive and no installation is needed. However, you can also install openSUSE from the running live system. There is no network connection required beyond the mere downloading of the medium.

TIP: Booting the LiveCD from an USB Stick

Live CD iso images can also be used as boot images for USB sticks. Create a bootable USB stick by using the command-line program `dd` with the following syntax:

```
dd if=ISO_IMAGE of=USB_STICK_DEVICE bs=4M
```

`dd` is available on Linux and MacOS by default. A Microsoft Windows* version can be downloaded from <http://www.chrysocome.net/dd>.

Warning: Using this `dd` command will erase all data on the USB device!

Mini CD

The Mini CD contains the minimal Linux system needed to run the installation. The installation system itself and the installation data are loaded from a network source. To install from a network providing SLP, please start the installation as described in Section 1.2.1, “Installing from a Network Server Using

SLP” (page 7). To install from a HTTP, FTP, NFS, or SMB server, follow the instructions in Section 1.2.2, “Installing from a Network Source without SLP” (page 8).

IMPORTANT: Add-On CDs—Installing Additional Software

Although add-on CDs (extensions or third-party products) cannot be used as stand-alone installation media, they can be embedded as additional software sources during the installation. Currently CDs with additional languages and non open source software are available as add-on CDs for openSUSE. Refer to Section 1.7.1, “Add-On Products” (page 14) for more information.

1.2 Choosing the Installation Method

After having selected the installation medium, determine the suitable installation method and boot option that best matches your needs:

Installing from the openSUSE Media

Choose this option if you want to perform a stand-alone installation and do not want to rely on a network to provide the installation data or the boot infrastructure. The installation proceeds exactly as outlined in Section 1.3, “The Installation Workflow” (page 9).

Installing from the LiveCD

In order to install from a LiveCD, boot the live system from CD. In the running system, launch the installation routine by clicking on the *Install* icon on the desktop. Phase one of the installation will be carried out in a window on the desktop. It is not possible to update an existing system with a LiveCD, you can only perform a new installation (with automatic configuration).

Installing from a Network Server

Choose this option if you have an installation server available in your network or want to use an external server as the source of your installation data. This setup can be configured to boot from physical media (Floppy, CD/DVD, or hard disk) or configured to boot via network using PXE/BOOTP. Refer to Section 1.2.1, “Installing from a Network Server Using SLP” (page 7), Section 1.2.2, “In-

stalling from a Network Source without SLP” (page 8), or Chapter 2, *Remote Installation* (page 35) for details.

Installing with openSUSE 12.3 Installer from Windows

Choose this installation option if you prefer a smooth transition from using Windows to using Linux. openSUSE 12.3 Installer allows you to boot into the openSUSE installation right from a running Windows by modifying the Windows boot loader. This installation option is only available from the DVD media. Refer to Section 1.2.3, “Installing with the openSUSE 12.3 Installer from Windows” (page 8) for details.

openSUSE supports several different boot options from which you can choose, depending on the hardware available and on the installation scenario you prefer. Booting from the openSUSE media or using openSUSE 12.3 Installer are the most straightforward options, but special requirements might call for special setups:

Table 1.1: *Boot Options*

Boot Option	Description
DVD	This is the easiest boot option. This option can be used if the system has a local DVD-ROM drive that is supported by Linux.
openSUSE 12.3 Installer	openSUSE 12.3 Installer is installed under Microsoft Windows and makes it possible to boot directly into the installation
PXE or BOOTP	Booting over the network 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 openSUSE system. See http://en.opensuse.org/SDB:PXE_boot_installation for more information.Refer to Chapter 2, <i>Remote</i>

Boot Option	Description
	<i>Installation</i> (page 35) for more information.
Hard Disk	openSUSE installation 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/architecture/</code> on the installation media to the hard disk and add an appropriate entry to the existing boot loader of a previous openSUSE installation.

TIP: Booting from DVD on UEFI machines

#amd64 em64t: DVD1 can be used as a boot medium for machines equipped with UEFI (Unified Extensible Firmware Interface). Refer to your vendor's documentation for specific information. If booting fails, try to enable CSM (Compatibility Support Module) in your firmware. ◀

1.2.1 Installing from a Network Server Using SLP

If your network setup supports OpenSLP and your network installation source has been configured to announce itself via SLP (described in Section 2.2, “Setting Up the Server Holding the Installation Sources” (page 44)), boot the system, press **F4** in the boot screen and select *SLP* from the menu.

The installation program configures the network connection with DHCP and retrieves the location of the network installation source from the OpenSLP server. If the automatic DHCP network configuration fails, you are prompted to enter the appropriate parameters manually. The installation then proceeds as described below with the exception of the network configuration step that is needed prior to adding additional repositories. This step is not needed as the network is already configured and active at this point.

1.2.2 Installing from a Network Source without SLP

If your network setup does not support OpenSLP for the retrieval of network installation sources, boot the system and press F4 in the boot screen to select the desired network protocol (NFS, HTTP, FTP, or SMB/CIFS). Provide the server's address and the path to the installation media.

The installation program automatically configures the network connection with DHCP. If this configuration fails, you are prompted to enter the appropriate parameters manually. The installation retrieves the installation data from the source specified. The installation then proceeds as described below with the exception of the network configuration step needed prior to adding additional repositories. This step is not needed as the network is already configured and active at this point.

1.2.3 Installing with the openSUSE 12.3 Installer from Windows

openSUSE 12.3 Installer is a Microsoft Windows application that prepares your computer to directly boot into the openSUSE installation without having to adjust BIOS settings. It is only available on DVD media. To use the installer, insert the openSUSE media under Windows. The openSUSE 12.3 Installer setup automatically starts (if not, run `openSUSE11_2_LOCAL.exe` from the DVD). Choose a language for the installation and follow the instructions on the screen. The language you choose here is also preconfigured to be used for the openSUSE installation.

On the next reboot, the Microsoft Windows boot loader launches. Choose *openSUSE 12.3 Installer* to start the openSUSE installation. In order to proceed with the installation, you will be prompted to insert the installation media. The installation proceeds as described below. When Microsoft Windows is booted again, openSUSE 12.3 Installer is automatically uninstalled.

TIP: Installing openSUSE alongside Microsoft Windows

openSUSE can easily be installed alongside Microsoft Windows. Carry out the installation as described below—an existing Windows installation will automatically be detected and a dual boot option will be installed. If Win-

dows covers the whole installation disk, the installation routine will make a proposal to shrink an existing Windows partition in order to make room for the openSUSE. Please read Section 1.10.1.1, “Resizing a Windows Partition” (page 19) prior to the installation for detailed information.

1.3 The Installation Workflow

The openSUSE installation is split into three main parts: preparation, installation, and configuration. During the preparation phase you configure some basic parameters such as language, time, desktop type, users, passwords, hard disk setup and installation scope. In the non-interactive installation phase the software is installed and the system is prepared for the first boot. Upon finishing the installation the machine reboots into the newly installed system and starts the final system configuration. You can choose whether to do a fully automatic or a manual configuration. In this stage, network and Internet access, as well as hardware components such as printers, are set up.

1.4 System Start-Up for Installation

You can install openSUSE from local installation sources, such as the openSUSE CDs or DVD, or from network source of an FTP, HTTP, NFS, or SMB server. Any of these approaches requires physical access to the system to install as well as user interaction during the installation. The installation procedure is basically the same regardless of the installation source. Any exceptions are sufficiently highlighted in the following workflow description.

1.5 The Boot Screen

The boot screen displays a number of options for the installation procedure. *Boot from Hard Disk* boots the installed system and is selected by default, because the CD is often left in the drive. Select one of the other options with the arrow keys and press Enter to boot it. The relevant options are:

Installation

The normal installation mode. All modern hardware functions are enabled. In case the installation fails, see “F5 Kernel” (page 11) for boot options that disable potentially problematic functions.

Rescue System

Starts a minimal Linux system without a graphical user interface. For more information, see Section “Using the Rescue System” (Appendix A, *Help and Troubleshooting*, ↑Start-Up). This option is not available on LiveCDs.

Check Installation Media

This option is only available when you install from media created from downloaded ISOs. In this case it is recommended to check the integrity of the installation medium. This option starts the installation system before automatically checking the media. In case the check was successful, the normal installation routine starts. If a corrupt media is detected, the installation routine aborts.

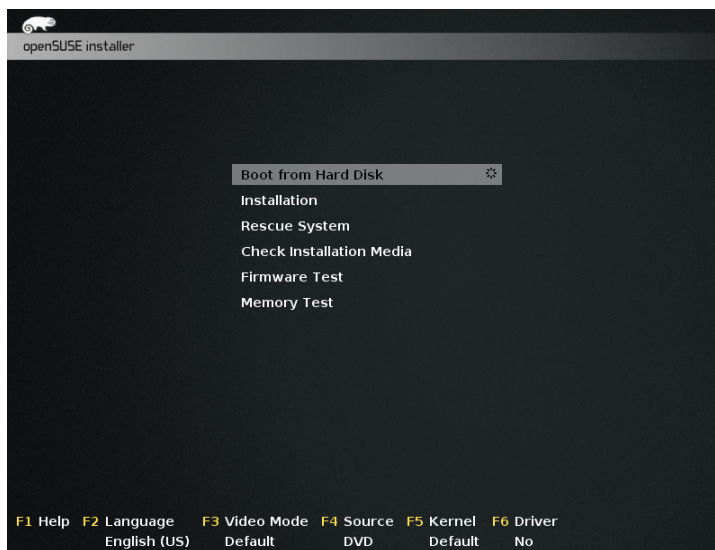
Firmware Test

Starts a BIOS checker that validates ACPI and other parts of your BIOS. This option is not available on the LiveCDs.

Memory Test

Tests your system RAM using repeated read and write cycles. Terminate the test by rebooting. For more information, see Section “Fails to Boot” (Appendix A, *Help and Troubleshooting*, ↑Start-Up). This option is not available on the LiveCDs.

Figure 1.1: *The Boot Screen*



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

F1 *Help*

Get context-sensitive help for the active element of the boot screen. Use the arrow keys to navigate, Enter to follow a link, and Esc to leave the help screen.

F2 *Language*

Select the display language and a corresponding keyboard layout for the installation. The default language is English (US).

F3 *Video Mode*

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

F4 *Source*

Normally, the installation is performed from the inserted installation medium. Here, select other sources, like FTP or NFS servers. If the installation is deployed on a network with an SLP server, select an installation source available on the server with this option. Find information about SLP in Chapter 14, *SLP Services in the Network* (page 275).

F5 *Kernel*

If you encounter problems with the regular installation, this menu offers to disable a few potentially problematic functions. If your hardware does not support ACPI (advanced configuration and power interface) select *No ACPI* to install without ACPI support. *No local APIC* disables support for APIC (Advanced Programmable Interrupt Controllers) which may cause problems with some hardware. *Safe Settings* boots the system with the DMA mode (for CD/DVD-ROM drives) and power management functions disabled.

If you are not sure, try the following options first: *Installation—ACPI Disabled* or *Installation—Safe Settings*. Experts can also use the command line (*Boot Options*) to enter or change kernel parameters.

F6 *Driver*

Press this key to notify the system that you have an optional driver update for openSUSE. With *File* or *URL*, load drivers directly before the installation starts. If you select *Yes*, you are prompted to insert the update disk at the appropriate point in the installation process.

F7 Arch

If you install from an installation medium supporting 32bit and 64bit architectures and have a processor with 64-bit support, select whether to install a 64-bit or 32-bit system. By default, a 64-bit system is installed on a computer with 64-bit support. To install a 32-bit system, press F7 then select *32bit*.

TIP: Using IPv6 during the Installation

By default you can only assign IPv4 network addresses to your machine. To enable IPv6 during installation, enter one of the following parameters at the bootprompt: `ipv6=1` (accept IPv4 and IPv6) or `ipv6only=1` (accept IPv6 only).

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

TIP: Installation without a Mouse

If the installer does not detect your mouse correctly, use Tab for navigation, arrow keys to scroll, and Enter to confirm a selection. Various buttons or selection fields contain a letter with an underscore. Use Alt + Letter to select a button or a selection directly instead of navigating there with the Tab button.

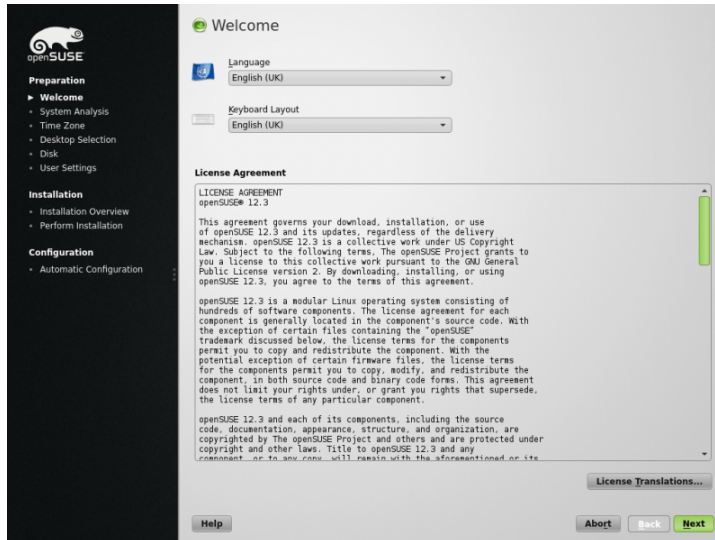
1.6 Welcome

Start the installation of openSUSE by choosing your language. Changing the language will automatically preselect a corresponding keyboard layout. Override this proposal by selecting a different keyboard layout from the drop-down menu. The language selected here is also used to assume a time zone for the system clock. This setting—along with the selection of secondary languages to install on your system—can be modified later in the *Installation Summary*, described in Section 1.12, “Installation Settings” (page 24). For information about language settings in the installed system, see Chapter 11, *Changing Language and Country Settings with YaST* (↑Start-Up).

Read the license agreement that is displayed beneath the language and keyboard selection thoroughly. Use *License Translations...* to access translations. If you agree to

the terms, click *Next* to proceed with the installation. If you do not agree to the license agreement click *Abort* to terminate the installation.

Figure 1.2: *Welcome*



1.7 Installation Mode

After a system analysis (where YaST probes for storage devices and tries to find other installed systems on your machine) the available installation modes are displayed. This step is skipped when installing from a LiveCD, since this medium only supports a new installation with automatic configuration.

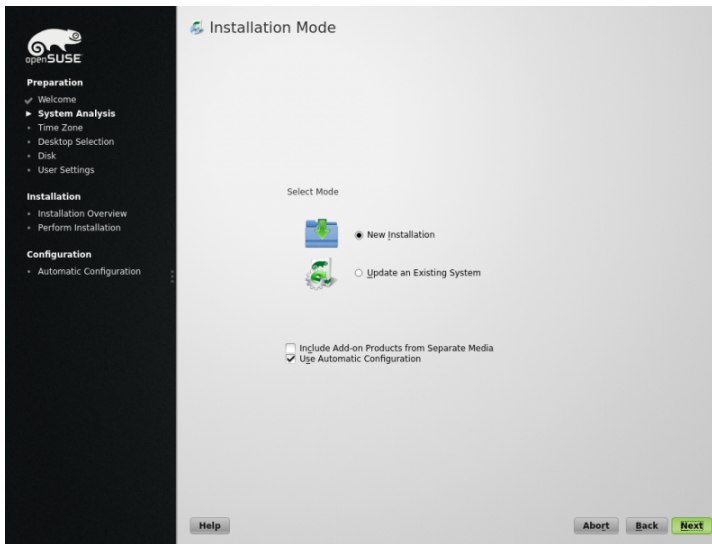
New installation

Select this option to start a new installation from scratch.

Update

Select this option to update an existing installation to a newer version. For more information about system updates, see Chapter 16, *Upgrading the System and System Changes* (↑Start-Up).

Figure 1.3: *Installation Mode*



By default, the automatic configuration is used when performing a new installation. In this mode the system automatically configures your hardware and the network, so the installation is performed with minimal user interaction. If necessary, you can change every configuration that is set up later in the installed system using YaST. Uncheck *Use Automatic Configuration* if you prefer a manual configuration during the installation.

Check *Include Add-On Products from Separate Media* to include add-on products during the installation. An add-on product can include extensions, third-party products and drivers or additional software for your system such as support for additional languages.

Click *Next* to proceed. If you selected to include an add-on product, proceed with Section 1.7.1, “Add-On Products” (page 14), otherwise skip the next section and advance to Section 1.8, “Clock and Time Zone” (page 16).

1.7.1 Add-On Products

Add-on products can be installed either from a local source (CD, DVD, or directory) or from a network source (HTTP, FTP, NFS, CIFS,...). When installing from a

network source, you need to configure the network first (unless you are performing a network installation— in this case the existing network configuration is used). Choose *Yes, Run the Network Setup* and proceed as described in Section 1.7.1.1, “Network Setup” (page 15). If the add-on product is available locally, select *No, Skip the Network Setup*.

Click *Next* and specify the product source. Source types available are *CD, DVD, Hard Disk, USB Mass Storage*, a *Local Directory* or a *Local ISO Image* (if no network was configured). If the add-on product is available on removable media, the system automatically mounts the media and reads its contents. If the add-on product is available on hard disk, choose *Hard Disk* to install from an unmounted hard drive, or *Local Directory/Local ISO Image* to install from the local file system. Add-on products may be delivered as a repository or as a set of rpm files. In the latter case, check *Plain RPM Directory*. Whenever a network is available, you can choose from additional remote sources such as HTTP, SLP, FTP, etc. It is also possible to specify a URL directly.

Check *Download Repository Description Files* to download the files describing the repository now. If unchecked, they will be downloaded once the installation starts. Proceed with *Next* and insert a CD or DVD if required. Depending on the product's content it may be necessary to accept additional license agreements.

It is also possible to configure add-on products later. Using add-on products on the installed system is described in Chapter 8, *Installing Add-On Products* (↑Start-Up).

1.7.1.1 Network Setup

When invoking the network setup, YaST scans for available network cards. If more than one network card is found, you must choose the card to configure from the list.

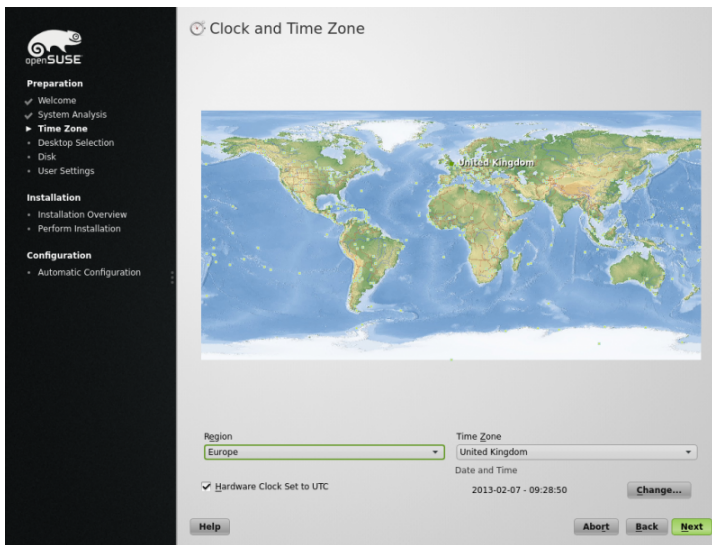
If an ethernet network adapter is not already connected, a warning will open. Make sure the network cable is plugged in and choose *Yes, Use It*. If your network is equipped with a DHCP server, choose *Automatic Address Setup (via DHCP)*. To manually set up the network choose *Static Address Setup* and specify *IP Address*, *Netmask*, *Default Gateway IP*, and the *DNS Server IP*.

Some networks require the use of a proxy server to access the Internet. Tick the checkbox *Use Proxy for Accessing the Internet* and enter the appropriate specifications. Click *Accept* to perform the network setup. The installation procedure will continue with the add-on products or repositories setup as described in Section 1.7.1, “Add-On Products” (page 14).

1.8 Clock and Time Zone

In this dialog, select your region and time zone. Both are preselected according to the selected installation language. To change the preselected values, either use the map or the drop down lists for *Region* and *Time Zone*. When using the map, point the cursor at the rough direction of your region and left-click to zoom. Now choose your country or region by left-clicking. Right-click to return to the world map.

Figure 1.4: *Clock and Time Zone*



To set up the clock, choose whether the *Hardware Clock is Set to UTC*. If you run another operating system on your machine, such as Microsoft Windows, it is likely your system uses local time instead. If you only run Linux on your machine, set the hardware clock to UTC and have the switch from standard time to daylight saving time performed automatically.

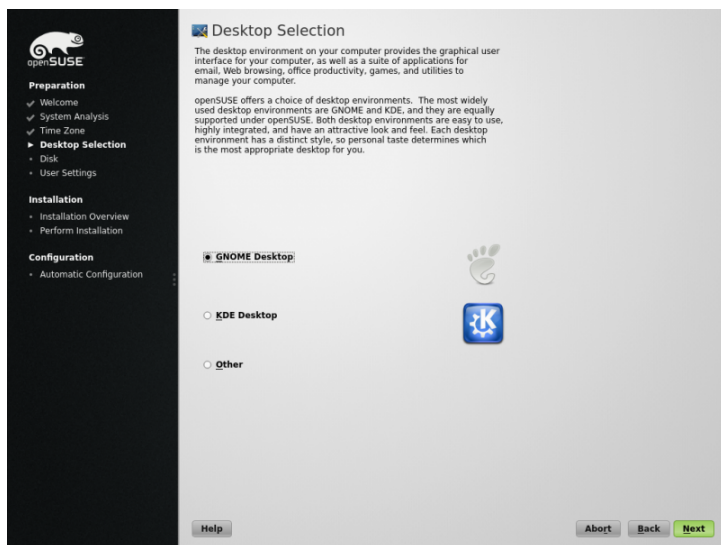
If a network is already configured, the time is automatically synced via Network Time Protocol (NTP) with a time server. Click *Change* to either alter the NTP settings or to *Manually* set the time. See Chapter 17, *Time Synchronization with NTP* (page 313) for more information on configuring the NTP service. When finished, click *Accept* to continue the installation.

1.9 Desktop Selection

In openSUSE, you can choose from various desktops. The major ones, *KDE* and *GNOME*, are powerful graphical desktop environments similar to Windows. This step is skipped when installing from a LiveCD, since this medium is already preconfigured to either use KDE or GNOME.

If you prefer a different desktop, choose *Other* for more options. The *XFCE Desktop* and the *LXDE Desktop* are fast and lightweight desktop environments suitable for modest hardware. With *Minimal X Window*, install a graphical window manager that allows for running stand-alone X applications and console windows but does not provide the usual integrated desktop functionality. In *Minimal Server Selection (Text Mode)*, only console terminals are available.

Figure 1.5: Desktop Selection



1.10 Suggested Partitioning

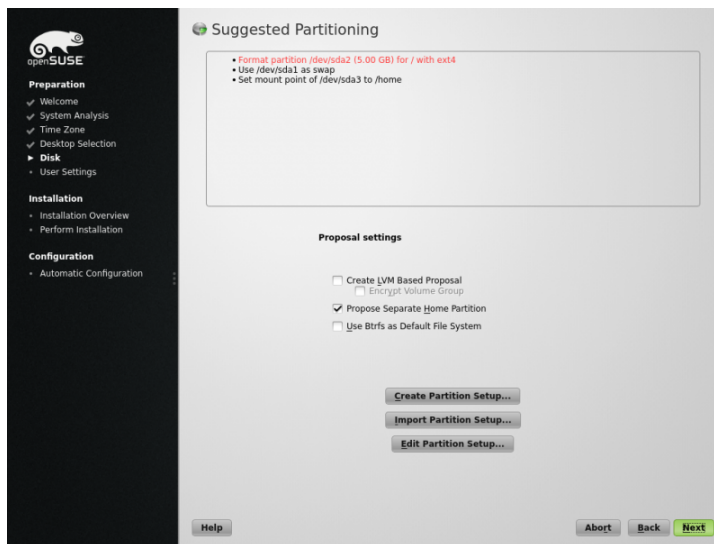
Define a partition setup for openSUSE in this step. In most cases a reasonable scheme that can be accepted without change is proposed. If a hard disk containing only Win-

dows FAT or NTFS partitions is selected as the installation target, YaST proposes to shrink one of these partitions. Accept the proposal with *Next* and proceed with the installation. Experienced users can also customize the proposal or apply their own partitioning scheme.

The proposed partitioning is *Partition Based* by default. If you prefer an *LVM Based* setup, check the respective option to automatically convert the proposal. Refer to Section 3.2, “LVM Configuration” (page 82) for more information about the Logical Volume Manager (LVM).

To make small changes in the proposal (like changing the file system type or encrypt partitions), select *Edit Partition Setup* and adjust the settings. See Section 3.1, “Using the YaST Partitioner” (page 71) for instructions.

Figure 1.6: *Suggested Partitioning*



1.10.1 Partitioning a Specific Disk

If your machine contains more than one hard disk and you want to restrict the partitioning proposal to just one disk, choose *Create Partition Setup* and then select a specific disk from the list. If the chosen hard disk does not contain any partitions yet, the whole hard disk will be used for the proposal. Otherwise, you can choose which exist-

ing partition(s) to utilize. To add a separate partition for you personal data check *Propose a Separate Home Partition*. Instead of the default partition-based proposal, it is possible to *Create an LVM Based Proposal*. Choose two times *Next* to proceed to the next step.

1.10.1.1 Resizing a Windows Partition

If the selected hard disk only contains a Windows FAT or NTFS partition, YaST offers to delete or shrink this partition. If you select *Delete Windows Completely*, the Windows partition is marked for deletion and the space is used for the installation of openSUSE.

WARNING: Deleting Windows

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

To *Shrink the Windows Partition*, you need to interrupt the installation and boot Windows to prepare the partition before shrinking it. For all Windows file systems, proceed as follows:

1. Deactivate a Virtual Memory file, if there is one.
2. Run `scandisk`.
3. Run `defrag`.

After these preparations, restart the openSUSE installation. When you turn to the partitioning setup, proceed as before and select *Shrink Windows Partition*. After a quick check of the partition, the dialog for resizing the Windows partition opens.

The bar graph shows how much disk space is currently occupied by Windows and how much space is still available. To change the proposed settings use the slider or the input fields to adjust the partition sizing.

Figure 1.7: *Resizing the Windows Partition*

Space Used	Space Free	Unused Disk
3356 MB	1000 MB	3839 MB

New Size (in MB)

4356

Free Space Remaining: 1000 MB

Minimum Partition Size: 3361 MB

Maximum Partition Size: 8195 MB

Cancel OK

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.

IMPORTANT: Writing on NTFS Partitions

By default, the Windows uses the NTFS file system. openSUSE includes read and write access to the NTFS file system, but this feature has a few limitations. This means that you cannot read or write encrypted or compressed files. Furthermore, the Windows file permissions are not honored at the moment. See <http://en.opensuse.org/SDB:NTFS> for more information.

1.10.2 Custom Partitioning

Set up your own partitioning scheme by selecting *Create Partition Setup* and then *Custom Partitioning*. The Expert Partitioner opens, displaying the current partition setup. Expand the *Hard Disks* in the System View pane by clicking on the +, then click on one of the listed hard disks. Now you can *Add*, *Edit*, *Resize*, or *Delete* partitions. For more information about custom partitioning and configuring advanced features, refer to Section 3.1, “Using the YaST Partitioner” (page 71)

1.11 Create New User

Create a local user in this step. Administrating local users is a suitable option for stand-alone workstations. If setting up a client on a network with centralized

user authentication, click *Change* and proceed with the Section 1.11.1, “Expert Settings” (page 22).

After entering the first name and last name, either accept the proposal or specify a new *Username* that will be used to log in. Finally, enter a password for the user. Reenter it for confirmation (to ensure that you did not type something else by mistake). To provide effective security, a password should be between five and eight characters long. The maximum length for a password is 72 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 (7-bit ASCII) and the digits 0 to 9 are allowed. Other special characters like umlauts or accented characters are not allowed.

Passwords you enter are checked for weakness. When entering a password that is easy to guess (such as a dictionary word or a name) you will see a warning. It is a good security practice to use strong passwords.

IMPORTANT: Username and Password

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

Figure 1.8: *Create New User*

The screenshot shows the 'Create New User' window in the openSUSE installer. The left sidebar contains a navigation menu with sections: Preparation (Welcome, System Analysis, Time Zone, Desktop Selection, Disk), User Settings (highlighted), Installation (Installation Overview, Perform Installation), and Configuration (Automatic Configuration). The main window title is 'Create New User'. It features four text input fields: 'User's Full Name', 'Username', 'Password', and 'Confirm Password'. Below these are three checkboxes: 'Use this password for system administrator' (checked), 'Receive System Mail' (unchecked), and 'Automatic Login' (checked). A 'Summary' section displays 'The authentication method is local /etc/passwd.' and 'The password encryption method is SHA-512.' with a 'Change...' button. At the bottom, there are 'Help', 'Abort', 'Back', and 'Next' buttons.

Three additional options are available:

Use this Password for the System Administrator

If checked, the same password you have entered for the user will be used for the system administrator `root`. This option is suitable for stand-alone workstations or machines in a home network that are administrated by a single user. When not checked, you are prompted for a system administrator password in the next step of the installation workflow (see Section 1.11.2, “Password for the System Administrator `root`” (page 23)).

Receive System Mail

Checking this box sends messages created by the system services to the user. 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.

The mails sent by system services are stored in the local mailbox `/var/spool/mail/username`, where *username* is the login name of the selected user. To read e-mails after installation, you can use any e-mail client, for example KMail or Evolution.

Automatic Login

This option automatically logs the current user in to the system when it starts. This is mainly useful if the computer is operated by only one user.

WARNING: Automatic Login

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

1.11.1 Expert Settings

Click *Change* in the Create User dialog to set up network authentication or, if present, import users from a previous installation. Also change the password encryption type in this dialog.

You can also add additional user accounts or change the user authentication method in the installed system. For detailed information about user management, see Chapter 10, *Managing Users with YaST* (↑Start-Up).

The default authentication method is *Local (/etc/passwd)*. If a former version of openSUSE or another system using */etc/passwd* is detected, you may import local users. To do so, check *Read User Data from a Previous Installation* and click *Choose*. In the next dialog, select the users to import and finish with *OK*.

Access to the following network authentication services can be configured:

LDAP

Users are administered centrally on an LDAP server for all systems in the network. More information is available in Section “Configuring an LDAP Client with YaST” (Chapter 4, *LDAP—A Directory Service*, ↑Security Guide).

NIS

Users are administered centrally on an NIS server for all systems in the network. See Section “Configuring NIS Clients” (Chapter 3, *Using NIS*, ↑Security Guide) for more information.

Windows Domain

SMB authentication is often used in mixed Linux and Windows networks, and Section “Configuring a Linux Client for Active Directory” (Chapter 5, *Active Directory Support*, ↑Security Guide).

Along with user administration via *LDAP* and *NIS*, you can use Kerberos authentication. To use it, select *Set Up Kerberos Authentication*. For more information on Kerberos, refer to Chapter 6, *Network Authentication with Kerberos* (↑Security Guide).

1.11.2 Password for the System Administrator `root`

If you have not chosen *Use this Password for the System Administrator* in the previous step, you will be prompted to enter a Password for the System Administrator `root`. Otherwise this configuration step is skipped.

`root` is the name of the superuser, or the administrator of the system. Unlike regular users (who may or may not have permission to access certain areas or execute certain commands on the system), `root` has unlimited access to change the system configuration, install programs, and set up new hardware. If users forget their passwords or have other problems with the system, `root` can help. The `root` account should only be used for system administration, maintenance, and repair. Logging in as `root` for

daily work is rather risky: a single mistake could lead to irretrievable loss of system files.

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

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

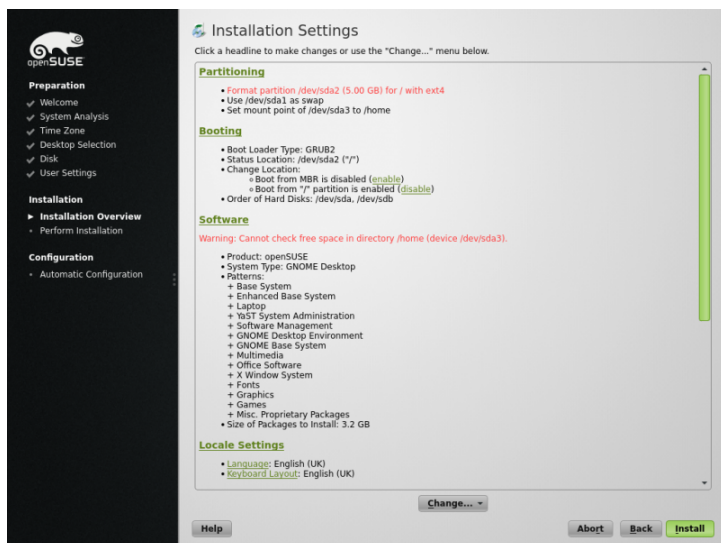
WARNING: The root User

The `root` user 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.

1.12 Installation Settings

On the last step before the real installation takes place, you can alter installation settings suggested by YaST and also review the settings you made so far. To modify the suggestions, either click *Change* and select the category to change or click on one of the headlines. After configuring any of the items presented in these dialogs, you are always returned to the Installation Settings window, which is updated accordingly.

Figure 1.9: *Installation Settings*



TIP: Restoring the Default Settings

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

1.12.1 Partitioning

Review and, if necessary, change the partition setup you configured earlier. Modifying the partition setup opens the Expert Partitioner described in Section 3.1, “Using the YaST Partitioner” (page 71).

1.12.2 Booting

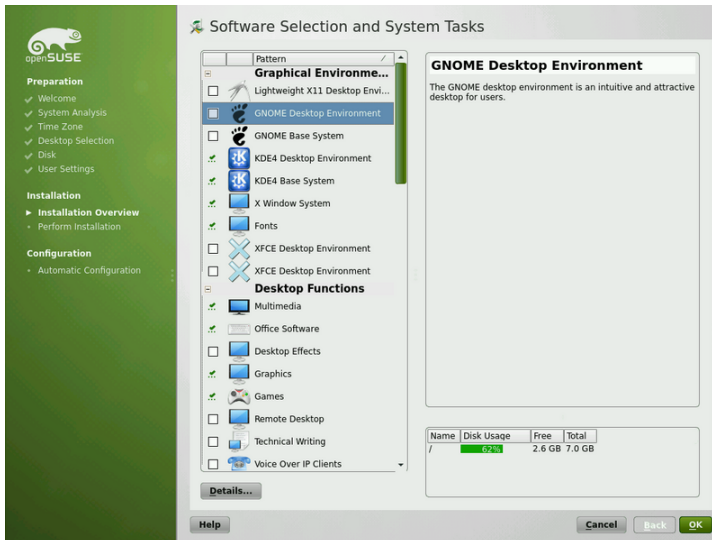
YaST proposes a boot configuration for your system. Other operating systems found on your computer, such as Microsoft Windows or other Linux installations, will automatically be detected and added to the boot loader. However, openSUSE will be booted by default. Normally, you can leave these settings unchanged. If you need a custom setup, modify the proposal for your system. For information, see Section 9.2, “Configuring the Boot Loader with YaST” (page 160). The boot method should only be changed by experienced users.

1.12.3 Software

openSUSE contains a number of software patterns for various application purposes. Click *Software* to start the pattern selection and modify the installation scope according to your needs. Select your pattern from the list and see a pattern description in the right part of the window. Each pattern contains a number of software packages needed for specific functions (e.g. Multimedia or Office software). For a more detailed selection based on software packages to install, select *Details* to switch to the YaST Software Manager.

You can also install additional software packages or remove software packages from your system at any later time with the YaST Software Manager. For more information, refer to Chapter 5, *Installing or Removing Software* (↑Start-Up).

Figure 1.10: *Software Selection and System Tasks*



1.12.4 Locale Settings

Here you can change the system *Language* and *Keyboard Layout* you defined in the first step of the installation. It is also possible to add additional languages. To adjust the system language settings, select *Language*. Select a language from the list. The primary language is used as the system language. You can also adapt keyboard layout and

time zone to the primary language if the current settings differ. *Details* lets you modify language settings for the user `root`, set UTF-8 support, or further specify the language (e.g. select South African English).

Choose secondary languages to be able to switch to one of these languages at any time without having to install additional packages. For more information, see Chapter 11, *Changing Language and Country Settings with YaST* (↑Start-Up).

To change the keyboard layout, select *Keyboard Layout*. By default, the layout corresponds to the language chosen for installation. Select the keyboard layout from the list. Use the *Test* field at the bottom of the dialog to check if you can enter special characters of that layout correctly. Options to fine-tune various settings are available under *Expert Mode*. When finished, click *Accept* to return to the installation summary.

1.12.5 Time Zone

Adjust time zone and clock settings here. Provided a network is configured, you can also set up a Network Time Protocol (NTP) client that automatically synchronizes your computer with a time server. This is the same configuration as shown earlier in Section 1.8, “Clock and Time Zone” (page 16).

1.12.6 User Settings

Change the current *User* settings and change or set the *Root Password* here. This is the same configuration as shown earlier in Section 1.11, “Create New User” (page 20).

1.12.7 Default Runlevel

openSUSE can boot to different runlevels. Normally, there should be no need to change anything here, but if necessary set the default runlevel with this dialog.

1.12.8 System

This dialog presents all the hardware information YaST could obtain about your computer. When called, the hardware detection routine is started. Depending on your sys-

tem, this may take some time. Select any item in the list and click *Details* to see detailed information about the selected item. Use *Save to File* to save a detailed list to either the local file system or a floppy. Advanced users can also change the PCI ID setup and Kernel Settings by choosing *Kernel Settings*.

1.12.9 *Installation from Images*

Installing from images considerably speeds up the installation. Images contain compressed snapshots of installed systems matching your selection of patterns. Packages not contained in the images deployed will be installed conventionally.

Unless your custom software selection does not match any of the available images, this feature is *Enabled* by default. In case of problems, *Disable* it for debugging purposes.

NOTE: Installation Time Stamps

The RPM database keeps track of the date a package was last installed or updated (check with for example `rpm -qa --last`). When installing from images, the time stamps of all packages originating from the images do not match the installation date but rather the date the image was created.

1.12.10 *Firewall*

By default SuSEFirewall2 is enabled on all configured network interfaces. To globally disable the firewall for this computer, click on *Disable*. If the firewall is enabled, you may *Open* the SSH port in order to allow remote connections via secure shell.

1.13 Performing the Installation

After configuring all installation settings, click *Install* in the Installation Settings window to start the installation. Some software may require a license confirmation. If your software selection includes such software, license confirmation dialogs are displayed. Click *Accept* to install the software package. When not agreeing to the license, click *I Disagree* and the software package will not be installed. In the dialog that follows, confirm with *Install* again.

The installation usually takes between 15 and 30 minutes, depending on the system performance and the selected software scope. After having prepared the hard disk, having saved and restored the user settings, and having deployed the installation images, the software installation starts. During this procedure a slide show introduces the features of openSUSE. Choose *Details* to switch to the installation log or *Release Notes* to read important up-to-date information which was not available when the manuals were printed.

NOTE: Release Notes

The release notes that can be viewed during this step are the ones printed on the installation CD. A newer version may be available on the Internet. When manually configuring network and Internet access as described in Section 1.14.2.2, “Network Configuration” (page 30), the latest version of the release notes will be displayed at the end of the installation.

After the software installation has completed, the basic system is set up. Among others, “Finishing the Basic Installation” includes installing the boot manager, initializing fonts and more. Next YaST boots into the new Linux system to start the system configuration.

TIP: Existing SSH Host Keys

If you install openSUSE on a machine with existing Linux installations, the installation routine automatically imports the SSH host key with the most recent access time from an existing installation.

1.14 Configuration of the Installed System

The system is now installed, but not yet configured for use. The hardware, the network and other services are not yet set up. If you follow the default installation path, the system will be automatically configured. If you have deselected the *Automatic Configuration*, the manual system configuration starts.

1.14.1 Automatic System Configuration

Having rebooted, the system starts the Automatic Configuration. This routine attempts to configure your network and Internet access and sets up your hardware. This process does not need any interaction. You can change the settings made by Automatic Configuration at any time on the installed system with YaST. Continue with Section 1.15, “Graphical Login” (page 34).

1.14.2 Manual System Configuration

Having rebooted, the system starts the manual configuration. If the configuration fails at one of the steps of this stage, it restarts and continues from the last successful step.

1.14.2.1 Hostname and Domain Name

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

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

To change hostname settings at any time after installation, use YaST *Network Devices > Network Settings*. For more information, see Section 13.4.1, “Configuring the Network Card with YaST” (page 233).

1.14.2.2 Network Configuration

If you are installing openSUSE on a laptop computer, *Interfaces Controlled by NetworkManager* is enabled. NetworkManager is a tool that enables automatic connection with minimal user intervention. It is ideal for WLAN and mobile computing. If you want to use the traditional method without NetworkManager, click *Disable NetworkManager*. Find detailed information about NetworkManager in Chapter 25, *Using NetworkManager* (page 435). If you are installing openSUSE on any other type

of machine, the traditional method without NetworkManager is selected by default. This configuration step also lets you configure the network devices of your system and make security settings, for example, for a firewall or proxy.

The network can also be configured after the system installation has been completed. If you skip it now, your system is left offline unable to retrieve any available updates. To configure your network connection later, select *Skip Configuration* and click *Next*.

The following network settings can be configured in this step:

General Network Settings

Enable or disable the use of NetworkManager as described above. Also change the IPv6 support here. By default the IPv6 support is enabled. To disable it, click *Disable IPv6*. For more information about IPv6, see Section 13.2, “IPv6—The Next Generation Internet” (page 221).

Firewall

By default SuSEFirewall2 is enabled on all configured network interfaces. To globally disable the firewall for this computer, click on *Disable*. If the firewall is enabled, you may *Open* the SSH port in order to allow remote connections via secure shell. To open the detailed firewall configuration dialog, click on *Firewall*. See Section “Configuring the Firewall with YaST” (Chapter 13, *Masquerading and Firewalls*, ↑Security Guide) for detailed information.

Network Interfaces

All network cards detected by YaST are listed here. If you have already set up a network connection during the installation (as described in Section 1.7.1.1, “Network Setup” (page 15)) the card used for this connection is listed as *Configured*. A click on *Network Interfaces* opens the *Network Settings* dialog, where you can change existing configurations, set up networks cards not configured yet, or add and configure additional cards.

DSL Connections, ISDN Adapters, and Modems

If your computer is equipped with an internal DSL modem, an internal ADSL Fritz Card, an ISDN card or a modem, clicking on the respective headline opens the configuration dialog.

VNC Remote Administration

To enable remote administration of your machine via VNC, click *VNC Remote Administration*. Choose *Allow Remote Administration* in the following dialog and adjust your firewall settings accordingly.

Proxy

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

TIP: Resetting the Network Configuration to the Default Values

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

Test Internet Connection

After having configured a network connection, you can test it. For this purpose, YaST establishes a connection to the openSUSE server and downloads the latest release notes. Read them at the end of the installation process. A successful test is also a prerequisite for the automatic addition of the default repositories and for updating online.

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

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

Proceed with *Next*. If the test was successful, the official software repositories for openSUSE and the update repository will be configured. Downloading the repository data for the first time may take some time.

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

1.14.2.3 Online Update

If an Internet connection has been established, and updates are available, 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. For detailed instructions see Chapter 6, *YaST Online Update* (↑Start-Up). Di-

rectives on how to perform an online update in the installed system are available at Section “Keeping the System Up-to-date” (Chapter 5, *Installing or Removing Software*, ↑Start-Up) or Chapter 6, *YaST Online Update* (↑Start-Up). This step is skipped if no updates are available or no Internet connection has been established. Patches fixing security issues and recommended patches applying to your installation are automatically preselected. Click *Accept* to install them and *Next* to proceed with the system configuration.

IMPORTANT: Downloading Software Updates

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

1.14.2.4 New Local User

If no local user was created in step one, you can create one in this dialog. To create more users, manage groups, modify defaults for new users and set up network authentication, launch *User Management*. Refer to Chapter 10, *Managing Users with YaST* (↑Start-Up) for more information about user management. To skip this step, click *Next* without entering any data.

1.14.2.5 Release Notes

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

1.14.2.6 Hardware Configuration

At the end of the installation, YaST opens a dialog for the configuration of *Graphics Cards Printer*. Click the individual components to start the hardware configuration. For the most part, YaST detects and configures the devices automatically.

You can skip any peripheral devices and configure them later, as described in Chapter 13, *Setting Up Hardware Components with YaST* (↑Start-Up). To skip the configuration, select *Skip Configuration* and click *Next*.

TIP: Resetting Hardware Configuration to the Default Values

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

1.14.2.7 Installation Completed

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

AutoYaST is a system for installing one or more openSUSE systems automatically without user intervention. AutoYaST installations are performed using a control file with installation and configuration data. Finish the installation of openSUSE with *Finish* in the final dialog.

1.15 Graphical Login

openSUSE is now fully installed and configured. Unless you enabled the automatic login function or customized the default runlevel, you should see the graphical login on your screen in which to enter a username and password to log into the system. On single user systems with automatic login enabled, the desktop starts automatically.

For a short introduction to the KDE or GNOME desktop environments, refer to the Chapter 3, *GNOME Quick Start* (↑Start-Up) and the Chapter 2, *KDE Quick Start* (↑Start-Up). These manuals can be accessed via the *Help* function in both KDE and GNOME.

Remote Installation

openSUSE® can be installed in different ways. As well as the usual media installation covered in Chapter 1, *Installation with YaST* (page 3), you can choose from various network-based approaches or even take a completely hands-off approach to the installation of openSUSE.

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

NOTE

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

2.1 Installation Scenarios for Remote Installation

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

IMPORTANT

The configuration of the X Window System is not part of any remote installation process. After the installation has finished, log in to the target system as `root`, enter `telinit 3`, and start `SaX2` to configure the graphics hardware.

2.1.1 Simple Remote Installation via VNC —Static Network Configuration

This type of installation still requires some degree of physical access to the target system to boot for installation. The installation itself is entirely controlled by a remote workstation using VNC to connect to the installation program. User interaction is required as with the manual installation in Chapter 1, *Installation with YaST* (page 3).

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

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

To perform this kind of installation, proceed as follows:

- 1 Set up the repository as described in Section 2.2, “Setting Up the Server Holding the Installation Sources” (page 44). Choose an NFS, HTTP, or FTP network server. For an SMB repository, refer to Section 2.2.5, “Managing an SMB Repository” (page 51).

- 2 Boot the target system using a boot medium (DVD, CD, or USB flash drive) of the openSUSE media kit. For more information about the openSUSE media kit, see Section 1.1, “Choosing the Installation Media” (page 3).
- 3 When the boot screen of the target system appears, use the boot options prompt to set the appropriate VNC options and the address of the repository. This is described in detail in Section 2.4, “Booting the Target System for Installation” (page 62).

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

- 4 On the controlling workstation, open a VNC viewing application or Web browser and connect to the target system as described in Section 2.5.1, “VNC Installation” (page 66).
- 5 Perform the installation as described in Chapter 1, *Installation with YaST* (page 3). Reconnect to the target system after it reboots for the final part of the installation.
- 6 Finish the installation.

2.1.2 Simple Remote Installation via VNC —Dynamic Network Configuration

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

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

- Remote repository: NFS, HTTP, FTP, or SMB with working network connection.
- Target system with working network connection.
- Controlling system with working network connection and VNC viewer software or Java-enabled browser (Firefox, Konqueror, Internet Explorer, or Opera).

- Boot the target system using a boot medium (DVD, CD, or USB flash drive) of the openSUSE media kit. For more information about the openSUSE media kit, see Section 1.1, “Choosing the Installation Media” (page 3).
- Running DHCP server providing IP addresses.

To perform this kind of installation, proceed as follows:

- 1** Set up the repository as described in Section 2.2, “Setting Up the Server Holding the Installation Sources” (page 44). Choose an NFS, HTTP, or FTP network server. For an SMB repository, refer to Section 2.2.5, “Managing an SMB Repository” (page 51).
- 2** Boot the target system using a boot medium (DVD, CD, or USB flash drive) of the openSUSE media kit. For more information about the openSUSE media kit, see Section 1.1, “Choosing the Installation Media” (page 3).
- 3** When the boot screen of the target system appears, use the boot options prompt to set the appropriate VNC options and the address of the repository. This is described in detail in Section 2.4, “Booting the Target System for Installation” (page 62).

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

- 4** On the controlling workstation, open a VNC viewing application or Web browser and connect to the target system as described in Section 2.5.1, “VNC Installation” (page 66).
- 5** Perform the installation as described in Chapter 1, *Installation with YaST* (page 3). Reconnect to the target system after it reboots for the final part of the installation.
- 6** Finish the installation.

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

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

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

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

To perform this type of installation, proceed as follows:

- 1** Set up the repository as described in Section 2.2, “Setting Up the Server Holding the Installation Sources” (page 44). Choose an NFS, HTTP, or FTP network server or configure an SMB repository as described in Section 2.2.5, “Managing an SMB Repository” (page 51).
- 2** Set up a TFTP server to hold a boot image that can be pulled by the target system. This is described in Section 2.3.2, “Setting Up a TFTP Server” (page 55).
- 3** Set up a DHCP server to provide IP addresses to all machines and reveal the location of the TFTP server to the target system. This is described in Section 2.3.1, “Setting Up a DHCP Server” (page 53).
- 4** Prepare the target system for PXE boot. This is described in further detail in Section 2.3.5, “Preparing the Target System for PXE Boot” (page 61).
- 5** Initiate the boot process of the target system using Wake on LAN. This is described in Section 2.3.7, “Wake on LAN” (page 62).

- 6 On the controlling workstation, open a VNC viewing application or Web browser and connect to the target system as described in Section 2.5.1, “VNC Installation” (page 66).
- 7 Perform the installation as described in Chapter 1, *Installation with YaST* (page 3). Reconnect to the target system after it reboots for the final part of the installation.
- 8 Finish the installation.

2.1.4 Simple Remote Installation via SSH—Static Network Configuration

This type of installation still requires some degree of physical access to the target system to boot for installation and to determine the IP address of the installation target. The installation itself is entirely controlled from a remote workstation using SSH to connect to the installer. User interaction is required as with the regular installation described in Chapter 1, *Installation with YaST* (page 3).

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

- Remote repository: NFS, HTTP, FTP, or SMB with working network connection.
- Target system with working network connection.
- Controlling system with working network connection and working SSH client software.
- Boot the target system using a boot medium (DVD, CD, or USB flash drive) of the openSUSE media kit. For more information about the openSUSE media kit, see Section 1.1, “Choosing the Installation Media” (page 3).
- Valid static IP addresses already assigned to the repository and the controlling system.
- Valid static IP address to assign to the target system.

To perform this kind of installation, proceed as follows:

- 1 Set up the repository as described in Section 2.2, “Setting Up the Server Holding the Installation Sources” (page 44). Choose an NFS, HTTP, or FTP network server. For an SMB repository, refer to Section 2.2.5, “Managing an SMB Repository” (page 51).

- 2 Boot the target system using a boot medium (DVD, CD, or USB flash drive) of the openSUSE media kit. For more information about the openSUSE media kit, see Section 1.1, “Choosing the Installation Media” (page 3).
- 3 When the boot screen of the target system appears, use the boot options prompt to set the appropriate parameters for network connection, address of the repository, and SSH enablement. This is described in detail in Section 2.4.2, “Using Custom Boot Options” (page 63).

The target system boots to a text-based environment, giving the network address under which the graphical installation environment can be addressed by any SSH client.
- 4 On the controlling workstation, open a terminal window and connect to the target system as described in Section 2.5.2.2, “Connecting to the Installation Program” (page 68).
- 5 Perform the installation as described in Chapter 1, *Installation with YaST* (page 3). Reconnect to the target system after it reboots for the final part of the installation.
- 6 Finish the installation.

2.1.5 Simple Remote Installation via SSH —Dynamic Network Configuration

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

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

- Remote repository: NFS, HTTP, FTP, or SMB with working network connection.
- Target system with working network connection.
- Controlling system with working network connection and working SSH client software.

- Physical boot medium (CD, DVD, or USB flash drive) for booting the target system.
- Running DHCP server providing IP addresses.

To perform this kind of installation, proceed as follows:

- 1 Set up the repository source as described in Section 2.2, “Setting Up the Server Holding the Installation Sources” (page 44). Choose an NFS, HTTP, or FTP network server. For an SMB repository, refer to Section 2.2.5, “Managing an SMB Repository” (page 51).
- 2 Boot the target system using a boot medium (DVD, CD, or USB flash drive) of the openSUSE media kit. For more information about the openSUSE media kit, see Section 1.1, “Choosing the Installation Media” (page 3).
- 3 When the boot screen of the target system appears, use the boot options prompt to pass the appropriate parameters for network connection, location of the installation source, and SSH enablement. See Section 2.4.2, “Using Custom Boot Options” (page 63) for detailed instructions on the use of these parameters.

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

- 4 On the controlling workstation, open a terminal window and connect to the target system as described in Section 2.5.2.2, “Connecting to the Installation Program” (page 68).
- 5 Perform the installation as described in Chapter 1, *Installation with YaST* (page 3). Reconnect to the target system after it reboots for the final part of the installation.
- 6 Finish the installation.

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

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

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

- Remote repository: NFS, HTTP, FTP, or SMB with working network connection.
- TFTP server.
- Running DHCP server for your network, providing a static IP to the host to install.
- Target system capable of PXE boot, networking, and Wake on LAN, plugged in and connected to the network.
- Controlling system with working network connection and SSH client software.

To perform this type of installation, proceed as follows:

- 1** Set up the repository as described in Section 2.2, “Setting Up the Server Holding the Installation Sources” (page 44). Choose an NFS, HTTP, or FTP network server. For the configuration of an SMB repository, refer to Section 2.2.5, “Managing an SMB Repository” (page 51).
- 2** Set up a TFTP server to hold a boot image that can be pulled by the target system. This is described in Section 2.3.2, “Setting Up a TFTP Server” (page 55).
- 3** Set up a DHCP server to provide IP addresses to all machines and reveal the location of the TFTP server to the target system. This is described in Section 2.3.1, “Setting Up a DHCP Server” (page 53).
- 4** Prepare the target system for PXE boot. This is described in further detail in Section 2.3.5, “Preparing the Target System for PXE Boot” (page 61).
- 5** Initiate the boot process of the target system using Wake on LAN. This is described in Section 2.3.7, “Wake on LAN” (page 62).
- 6** On the controlling workstation, start an SSH client and connect to the target system as described in Section 2.5.2, “SSH Installation” (page 68).
- 7** Perform the installation as described in Chapter 1, *Installation with YaST* (page 3). Reconnect to the target system after it reboots for the final part of the installation.
- 8** Finish the installation.

2.2 Setting Up the Server Holding the Installation Sources

Depending on the operating system running on the machine to use as the network installation source for openSUSE, there are several options for the server configuration. The easiest way to set up an installation server is to use YaST on openSUSE 11.1 and higher.

TIP

You can even use a Microsoft Windows machine as the installation server for your Linux deployment. See Section 2.2.5, “Managing an SMB Repository” (page 51) for details.

2.2.1 Setting Up an Installation Server Using YaST

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

- 1 Log in as `root` to the machine that should act as installation server.
- 2 Install the `yast2-instserver` package.
- 3 Start *YaST > Miscellaneous > Installation Server*.
- 4 Select the repository type (HTTP, FTP, or NFS). The selected service is started automatically every time the system starts. If a service of the selected type is already running on your system and you want to configure it manually for the server, deactivate the automatic configuration of the server service with *Do Not Configure Any Network Services*. In both cases, define the directory in which the installation data should be made available on the server.
- 5 Configure the required repository type. This step relates to the automatic configuration of server services. It is skipped when automatic configuration is deactivated.

Define an alias for the root directory of the FTP or HTTP server on which the installation data should be found. The repository will later be located under `ftp://Server-IP/Alias/Name` (FTP) or under `http://Server-IP/Alias/Name` (HTTP). *Name* stands for the name of the repository, which is defined in the following step. If you selected NFS in the previous step, define wild cards and export options. The NFS server will be accessible under `nfs://Server-IP/Name`. Details of NFS and exports can be found in Chapter 18, *Sharing File Systems with NFS* (page 321).

TIP: Firewall Settings

Make sure that the firewall settings of your server system allow traffic on the ports for HTTP, NFS, and FTP. If they currently do not, enable *Open Port in Firewall* or check *Firewall Details* first.

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

TIP

Consider announcing your repository via OpenSLP if your network set-up supports this option. This saves you from entering the network installation path on every target machine. The target systems are just booted using the SLP boot option and find the network repository without any further configuration. For details on this option, refer to Section 2.4, “Booting the Target System for Installation” (page 62).

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

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

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

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

2.2.2 Setting Up an NFS Repository Manually

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

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

- 1 Log in as `root`.
- 2 Create a directory that will later hold all installation data and change into this directory. For example:

```
mkdir install/product/productversion
cd install/product/productversion
```

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

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

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

```
cp -a /media/path_to_your_DVD_drive .
```


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

3b Rename the directory to the DVD number:

```
mv path_to_your_DVD_drive DVDx
```

Replace `x` with the actual number of your DVD.

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

- 1** Log in as `root`.
- 2** Start *YaST > Network Services > NFS Server*.
- 3** Select *Start and Open Port in Firewall* and click *Next*.
- 4** Select *Add Directory* and browse for the directory containing the installation sources, in this case, `productversion`.
- 5** Select *Add Host* and enter the hostnames of the machines to which to export the installation data. Instead of specifying hostnames here, you could also use wild cards, ranges of network addresses, or just the domain name of your network. Enter the appropriate export options or leave the default, which works fine in most setups. For more information about the syntax used in exporting NFS shares, read the `exports` man page.
- 6** Click *Finish*. The NFS server holding the openSUSE repository is automatically started and integrated into the boot process.

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

- 1** Log in as `root`.
- 2** Open the file `/etc/exports` and enter the following line:

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

This exports the directory `/productversion` to any host that is part of this network or to any host that can connect to this server. To limit the access to this serv-

er, use netmasks or domain names instead of the general wild card *. Refer to the `export` man page for details. Save and exit this configuration file.

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

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

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

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

- 1 Log in as `root`.
- 2 Create the `/etc/slp.reg.d/install.suse.nfs.reg` configuration file with the following lines:

```
# Register the NFS Installation Server
service:install.suse:nfs://$HOSTNAME/path_to_repository/DVD1,en,65535
description=NFS Repository
```

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

- 3 Start the OpenSLP daemon with `rcslpd start`.

For more information about OpenSLP, refer to the package documentation located under `/usr/share/doc/packages/openslp/` or refer to Chapter 14, *SLP Services in the Network* (page 275). More Information about NFS, refer to Chapter 18, *Sharing File Systems with NFS* (page 321).

2.2.3 Setting Up an FTP Repository Manually

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

- 1** Create a directory holding the installation sources as described in Section 2.2.2, “Setting Up an NFS Repository Manually” (page 46).
- 2** Configure the FTP server to distribute the contents of your installation directory:

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

2b Enter the FTP server root directory:

```
cd /srv/ftp
```

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

```
mkdir repository
```

Replace *repository* with the product name.

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

```
mount --bind path_to_repository /srv/ftp/repository
```

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

2e Start `vsftpd` with `vsftpd`.

- 3** Announce the repository via OpenSLP, if this is supported by your network setup:

3a Create the `/etc/slp.reg.d/install.suse.ftp.reg` configuration file with the following lines:

```
# Register the FTP Installation Server
service:install.suse:ftp://$HOSTNAME/repository/DVD1,en,65535
description=FTP Repository
```

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

3b Start the OpenSLP daemon with `rcslpd start`.

TIP: Configuring an FTP Server with YaST

If you prefer using YaST over manually configuring the FTP installation server, refer to Chapter 21, *Setting up an FTP server with YaST* (page 389) for more information on how to use the YaST FTP server module.

2.2.4 Setting Up an HTTP Repository Manually

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

- 1 Create a directory holding the installation sources as described in Section 2.2.2, “Setting Up an NFS Repository Manually” (page 46).
- 2 Configure the HTTP server to distribute the contents of your installation directory:

2a Install the Web server Apache as described in Section 20.1.2, “Installation” (page 348).

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

```
mkdir repository
```

Replace *repository* with the product name.

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

```
ln -s /path_to_repository /srv/www/htdocs/repository
```

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

```
Options None
```

with

```
Options Indexes FollowSymLinks
```

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

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

3a Create the `/etc/slp.reg.d/install.suse.http.reg` configuration file with the following lines:

```
# Register the HTTP Installation Server
service:install.suse:http://$HOSTNAME/repository/DVD1/,en,65535
description=HTTP Repository
```

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

3b Start the OpenSLP daemon using `rcslpd restart`.

2.2.5 Managing an SMB Repository

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

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

- 1** Log in to your Windows machine.
- 2** Create a new folder that will hold the entire installation tree and name it `INSTALL`, for example.
- 3** Export this share according the procedure outlined in your Windows documentation.
- 4** Enter this share and create a subfolder, called *product*. Replace *product* with the actual product name.
- 5** Enter the `INSTALL/product` folder and copy each DVD to a separate folder, such as `DVD1` and `DVD2`.

To use a SMB mounted share as a repository, proceed as follows:

- 1** Boot the installation target.
- 2** Select *Installation*.

- 3 Press **F4** for a selection of the repository.
- 4 Choose **SMB** and enter the Windows machine's name or IP address, the share name (`INSTALL/product/DVD1`, in this example), username, and password.

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

2.2.6 Using ISO Images of the Installation Media on the Server

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

- 1 Download the ISO images and save them to the machine to use as the installation server.
- 2 Log in as `root`.
- 3 Choose and create an appropriate location for the installation data, as described in Section 2.2.2, “Setting Up an NFS Repository Manually” (page 46), Section 2.2.3, “Setting Up an FTP Repository Manually” (page 48), or Section 2.2.4, “Setting Up an HTTP Repository Manually” (page 50).
- 4 Create subdirectories for each DVD.
- 5 To mount and unpack each ISO image to the final location, issue the following command:

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

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

- 6 Repeat the previous step to mount all ISO images needed for your product.
- 7 Start your installation server as usual, as described in Section 2.2.2, “Setting Up an NFS Repository Manually” (page 46), Section 2.2.3, “Setting Up an FTP

Repository Manually” (page 48), or Section 2.2.4, “Setting Up an HTTP Repository Manually” (page 50).

To automatically mount the ISO images at boot time, add the respective mount entries to `/etc/fstab`. An entry according to the previous example would look like the following:

```
path_to_iso path_to_repository/productmedium auto loop
```

2.3 Preparing the Boot of the Target System

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

2.3.1 Setting Up a DHCP Server

There are two ways to set up a DHCP server. For openSUSE, YaST provides a graphical interface to the process. Users can also manually edit the configuration files. For more information about DHCP servers, see also Chapter 16, *DHCP* (page 303).

2.3.1.1 Setting Up a DHCP Server with YaST

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

- 1 Log in as `root` to the machine hosting the DHCP server.
- 2 Install the `yast2-dhcp-server` package.
- 3 Start *YaST > Network Services > DHCP Server*.
- 4 Complete the setup wizard for basic DHCP server setup.
- 5 Select *Expert Settings* and select *Yes* when warned about leaving the start-up dialog.

- 6 In the *Configured Declarations* dialog, select the subnet in which the new system should be located and click *Edit*.
- 7 In the *Subnet Configuration* dialog select *Add* to add a new option to the subnet's configuration.
- 8 Select filename and enter `pxelinux.0` as the value.
- 9 Add another option (`next-server`) and set its value to the address of the TFTP server.
- 10 Select *OK* and *Finish* to complete the DHCP server configuration.

To configure DHCP to provide a static IP address to a specific host, enter the *Expert Settings* of the DHCP server configuration module (Step 5 (page 53)) and add a new declaration of the host type. Add the options `hardware` and `fixed-address` to this host declaration and provide the appropriate values.

2.3.1.2 Setting Up a DHCP Server Manually

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

- 1 Log in as `root` to the machine hosting the DHCP server.
- 2 Append the following lines to a subnet configuration of your DHCP server's configuration file located under `/etc/dhcpd.conf`:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range dynamic-bootp 192.168.1.200 192.168.1.228;
    # PXE related stuff
    #
    # "next-server" defines the tftp server that will be used
    next-server ip_tftp_server;
    #
    # "filename" specifies the pxelinux image on the tftp server
    # the server runs in chroot under /srv/tftpboot
    filename "pxelinux.0";
}
```

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

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

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

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

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

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

2.3.2 Setting Up a TFTP Server

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

2.3.2.1 Setting Up a TFTP Server Using YaST

- 1 Log in as `root`.
- 2 Install the `yast2-tftp-server` package.
- 3 Start *YaST > Network Services > TFTP Server* and install the requested package.

- 4 Click *Enable* to make sure that the server is started and included in the boot routines. No further action from your side is required to secure this. `xinetd` starts `tftpd` at boot time.
- 5 Click *Open Port in Firewall* to open the appropriate port in the firewall running on your machine. If there is no firewall running on your server, this option is not available.
- 6 Click *Browse* to browse for the boot image directory. The default directory `/tftpbboot` is created and selected automatically.
- 7 Click *Finish* to apply your settings and start the server.

2.3.2.2 Setting Up a TFTP Server Manually

- 1 Log in as `root` and install the packages `tftp` and `xinetd`.
- 2 If unavailable, create `/srv/tftpbboot` and `/srv/tftpbboot/pxelinux.cfg` directories.
- 3 Add the appropriate files needed for the boot image as described in Section 2.3.3, “Using PXE Boot” (page 57).
- 4 Modify the configuration of `xinetd` located under `/etc/xinetd.d` to make sure that the TFTP server is started on boot:

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

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

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

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

2.3.3 Using PXE Boot

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

- 1 Change to the directory `boot/<architecture>/loader` of your installation repository and copy the `linux`, `initrd`, `message`, `biostest`, and `memtest` files to the `/srv/tftpboot` directory by entering the following:

```
cp -a linux initrd message biostest memtest /srv/tftpboot
```

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

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

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

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

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

- 5 Edit the `/srv/tftpboot/pxelinux.cfg/default` file and remove the lines beginning with `readinfo` and `framebuffer`.
- 6 Insert the following entries in the append lines of the default `failsafe` and `apic` labels:

```
insmod=kernel module
```

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

```
netdevice=interface
```

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

```
install=nfs://ip_instserver/path_to_repository/DVD1
```

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

IMPORTANT

If you need to pass other boot options to the installation routines, such as SSH or VNC boot parameters, append them to the *install* entry. An overview of parameters and some examples are given in Section 2.4, “Bootting the Target System for Installation” (page 62).

TIP: Changing Kernel and initrd Filenames

It is possible to use different filenames for Kernel and initrd images. This is useful if you want to provide different operating systems from the same boot server. However, you should be aware that only one dot is permitted in the filenames that are provided by TFTP for the PXE boot.

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

```
default harddisk

# default
label linux
    kernel linux
    append initrd=initrd ramdisk_size=65536 \
        install=nfs://ip_instserver/path_to_repository/product/DVD1

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

```

# bios test
label firmware
    kernel linux
    append initrd=biostest,initrd splash=silent install=exec:/bin/
run_biostest showopts

# memory test
label memtest
    kernel memtest

# hard disk
label haddisk
    localboot 0

implicit      0
display      message
prompt       1
timeout      100

```

- 7 Replace *ip_instserver* and *path_to_repository* with the values used in your setup.

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

2.3.4 PXELINUX Configuration Options

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

DEFAULT *kernel options...*

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

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

APPEND *options...*

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

`LABEL label KERNEL image APPEND options...`

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

Note that GRUB uses the following syntax:

```
title mytitle
  kernel my_kernelmy_kernel_options
  initrd myinitrd
```

PXELINUX uses the following syntax:

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

Labels are mangled as if they were filenames and they must be unique after mangling. For example, the two labels “v2.6.30” and “v2.6.31” would not be distinguishable under PXELINUX because both mangle to the same DOS filename.

The Kernel does not have to be a Linux Kernel; it can be a boot sector or a COM-BOOT file.

APPEND -

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

LOCALBOOT *type*

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

Argument	Description
0	Perform a normal boot
4	Perform a local boot with the Universal Network Driver Interface

Argument	Description
	(UNDI) driver still resident in memory
5	Perform a local boot with the entire PXE stack, including the UNDI driver, still resident in memory

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

TIMEOUT *time-out*

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

PROMPT *flag_val*

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

F2 *filename*
F1 *filename*
...etc...
F9 *filename*
F10 *filename*

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

2.3.5 Preparing the Target System for PXE Boot

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

WARNING: BIOS Boot Order

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

2.3.6 Preparing the Target System for Wake on LAN

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

2.3.7 Wake on LAN

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

IMPORTANT: Wake on LAN across Different Network Segments

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

2.4 Booting the Target System for Installation

Basically, there are two different ways to customize the boot process for installation apart from those mentioned under Section 2.3.7, “Wake on LAN” (page 62) and Section 2.3.3, “Using PXE Boot” (page 57). You can either use the default boot options and function keys or use the boot options prompt of the installation boot screen to pass any boot options that the installation Kernel might need on this particular hardware.

2.4.1 Using the Default Boot Options

The boot options are described in detail in Chapter 1, *Installation with YaST* (page 3). Generally, just selecting *Installation* starts the installation boot process.

If problems occur, use *Installation—ACPI Disabled* or *Installation—Safe Settings*. For more information about troubleshooting the installation process, refer to Section “Installation Problems” (Appendix A, *Help and Troubleshooting*, ↑Start-Up).

The menu bar at the bottom screen offers some advanced functionality needed in some setups. Using the F keys, you can specify additional options to pass to the installation routines without having to know the detailed syntax of these parameters (see Section 2.4.2, “Using Custom Boot Options” (page 63)). A detailed description of the available function keys is available at Section 1.5, “The Boot Screen” (page 9).

2.4.2 Using Custom Boot Options

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

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

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

Replace all the values `xxx` in this string with the values appropriate for your setup.

Chapter 1, *Installation with YaST* (page 3)

Parameters Needed for Booting None

Boot Options None needed

Section 2.1.1, “Simple Remote Installation via VNC—Static Network Configuration” (page 36)

Parameters Needed for Booting

- Location of the installation server

- Network device
- IP address
- Netmask
- Gateway
- VNC enablement
- VNC password

Boot Options

- `install=(nfs,http,ftp,smb):// path_to_instmedia`
- `netdevice=some_netdevice` (only needed if several network devices are available)
- `hostip=some_ip`
- `netmask=some_netmask`
- `gateway=ip_gateway`
- `vnc=1`
- `vncpassword=some_password`

Section 2.1.2, “Simple Remote Installation via VNC—Dynamic Network Configuration” (page 37)

Parameters Needed for Booting

- Location of the installation server
- VNC enablement
- VNC password

Boot Options

- `install=(nfs,http,ftp,smb):// path_to_instmedia`
- `vnc=1`
- `vncpassword=some_password`

Section 2.1.3, “Remote Installation via VNC—PXE Boot and Wake on LAN” (page 39)

Parameters Needed for Booting

- Location of the installation server
- Location of the TFTP server
- VNC enablement
- VNC password

Boot Options Not applicable; process managed through PXE and DHCP

Section 2.1.4, “Simple Remote Installation via SSH—Static Network Configuration”
(page 40)

Parameters Needed for Booting

- Location of the installation server
- Network device
- IP address
- Netmask
- Gateway
- SSH enablement
- SSH password

Boot Options

- `install=(nfs,http,ftp,smb):: path_to_instmedia`
- `netdevice=some_netdevice` (only needed if several network devices are available)
- `hostip=some_ip`
- `netmask=some_netmask`
- `gateway=ip_gateway`
- `usessh=1`
- `sshpassword=some_password`

Section 2.1.5, “Simple Remote Installation via SSH—Dynamic Network Configuration” (page 41)

Parameters Needed for Booting

- Location of the installation server
- SSH enablement
- SSH password

Boot Options

- `install=(nfs,http,ftp,smb):: path_to_instmedia`
- `usessh=1`
- `sshpassword=some_password`

Section 2.1.6, “Remote Installation via SSH—PXE Boot and Wake on LAN”
(page 42)

- Location of the installation server
- Location of the TFTP server

- SSH enablement
- SSH password

Boot Options Not applicable; process managed through PXE and DHCP

TIP: More Information about linuxrc Boot Options

Find more information about the linuxrc boot options used for booting a Linux system at <http://en.opensuse.org/SDB:Linuxrc>.

2.5 Monitoring the Installation Process

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

2.5.1 VNC Installation

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

2.5.1.1 Preparing for VNC Installation

All you need to do on the installation target to prepare for a VNC installation is to provide the appropriate boot options at the initial boot for installation (see Section 2.4.2, “Using Custom Boot Options” (page 63)). The target system boots into a text-based environment and waits for a VNC client to connect to the installation program.

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

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

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

2.5.1.2 Connecting to the Installation Program

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

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

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

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

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

ip_address:display_number

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

Using a Web browser to connect to the installation program makes you totally independent of any VNC software or the underlying operating system. As long as the browser application has Java support enabled, you can use any browser (Firefox, In-

ternet Explorer, Konqueror, Opera, etc.) to perform the installation of your Linux system.

To perform a VNC installation, proceed as follows:

- 1 Launch your preferred Web browser.
- 2 Enter the following at the address prompt:
`http://ip_address_of_target:5801`
- 3 Enter your VNC password when prompted to do so. The browser window now displays the YaST screens as in a normal local installation.

2.5.2 SSH Installation

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

2.5.2.1 Preparing for SSH Installation

Apart from installing the appropriate software package (OpenSSH for Linux and PuTTY for Windows), you just need to pass the appropriate boot options to enable SSH for installation. See Section 2.4.2, “Using Custom Boot Options” (page 63) for details. OpenSSH is installed by default on any SUSE Linux-based operating system.

2.5.2.2 Connecting to the Installation Program

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

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

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

- 3** When prompted for a username, enter `root`.
- 4** When prompted for the password, enter the password that has been set with the SSH boot option. After you have successfully authenticated, a command line prompt for the installation target appears.
- 5** Enter `yast` to launch the installation program. A window opens showing the normal YaST screens as described in Chapter 1, *Installation with YaST* (page 3).

Advanced Disk Setup

Sophisticated system configurations require specific disk setups. All common partitioning tasks can be done with YaST. To get persistent device naming with block devices, use the block devices below `/dev/disk/by-id` or `/dev/disk/by-uuid`. Logical Volume Management (LVM) is a disk partitioning scheme that is designed to be much more flexible than the physical partitioning used in standard setups. Its snapshot functionality enables easy creation of data backups. Redundant Array of Independent Disks (RAID) offers increased data integrity, performance, and fault tolerance. openSUSE also supports multipath I/O , and there is also the option to use iSCSI as a networked disk.

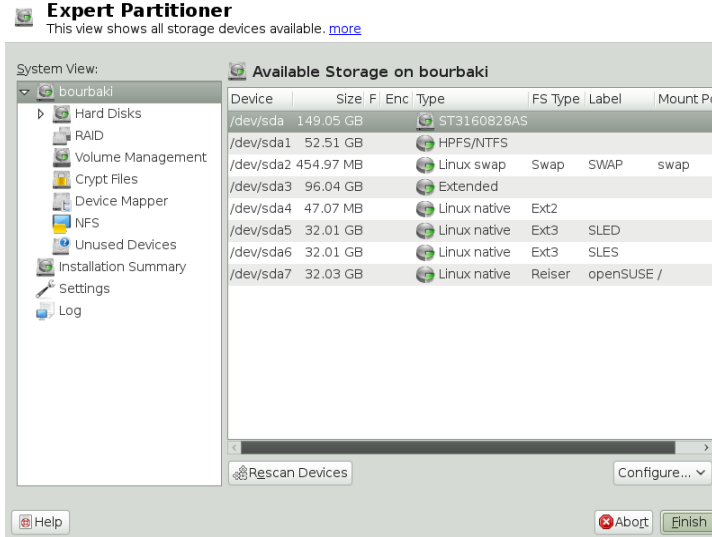
3.1 Using the YaST Partitioner

With the expert partitioner, shown in Figure 3.1, “The YaST Partitioner” (page 72), manually modify the partitioning of one or several hard disks. You can add, delete, resize, and edit partitions, as well as access the soft RAID, and LVM configuration.

WARNING: Repartitioning the Running System

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

Figure 3.1: *The YaST Partitioner*



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

Several functional views are available on the lefthand *System View*. Use these views to gather information about existing storage configurations, or to configure functions like RAID, Volume Management, Crypt Files, or view filesystems with additional features, such as BTRFS, NFS, or TMPFS.

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

3.1.1 Partition Types

Every hard disk has a partition table with space for four entries. Every entry in the partition table corresponds 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 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 be divided into *logical partitions* itself. 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 occupy the entire remaining free cylinder range. Then create multiple logical partitions within the extended partition. The maximum number of logical partitions is 63, independent of the disk type. It does not matter which types of partitions are used for Linux. Primary and logical partitions both function normally.

3.1.2 Creating a Partition

To create a partition from scratch select *Hard Disks* and then a hard disk with free space. The actual modification can be done in the *Partitions* tab:

- 1 Select *Add* and specify the partition type (primary or extended). Create up to four primary partitions or up to three primary partitions and one extended partition. Within the extended partition, create several logical partitions (see Section 3.1.1, “Partition Types” (page 72)).
- 2 Specify the size of the new partition. You can either choose to occupy all the free unpartitioned space, or enter a custom size.
- 3 Select the file system to use and a mount point. YaST suggests a mount point for each partition created. To use a different mount method, like mount by label, select *Fstab Options*.
- 4 Specify additional file system options if your setup requires them. This is necessary, for example, if you need persistent device names. For details on the available options, refer to Section 3.1.3, “Editing a Partition” (page 76).

5 Click *Finish* to apply your partitioning setup and leave the partitioning module.

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

3.1.2.1 Btrfs Partitioning

If you want to use Btrfs and *Storage Administration Guide*, chapter *Overview of File Systems in Linux* for more information on Btrfs) as your default filesystem for a newly installed system, click *Partitioning* on the *Installation Settings* screen, and check *Use Btrfs as Default Filesystem*. The installation system then suggests creating the `/boot` partition formatted with Ext3 filesystem, and the root `/` partition formatted with Btrfs holding a default set of subvolumes, which you can modify with the *Expert Partitioner* tool later.

The root filesystem is the default subvolume and it is not listed in the list of created subvolumes. As a default Btrfs subvolume, it can be mounted as a normal filesystem.

It is possible to create snapshots of Btrfs subvolumes - either manually, or automatically based on system events. For example when making changes to the filesystem, `zypper` invokes the `snapper` command to create snapshots before and after the change. This is useful if you are not satisfied with the change `zypper` made and want to restore the previous state. As `snapper` invoked by `zypper` snapshots the *root* filesystem by default, it is reasonable to exclude specific directories from being snapshot, depending on the nature of data they hold. And that is why YaST suggests creating the following separate subvolumes.

Suggested Btrfs Subvolumes

`/tmp` `/var/tmp` `/var/run`

Directories with frequently changed content.

`/var/spool`

Contains user data, such as mails.

`/var/log`

Contains system and applications' log files which should never be rolled back.

`/var/crash`

Contains memory dumps of crashed kernels.

/srv

Contains data files belonging to FTP and HTTP servers.

/opt

Contains third party software.

TIP: Size of Btrfs Partition

Because saved snapshots require more disk space, it is recommended to reserve more space for Btrfs partition than for a partition not capable of snapshotting (such as Ext3). Recommended size for a root Btrfs partition with suggested subvolumes is 20GB.

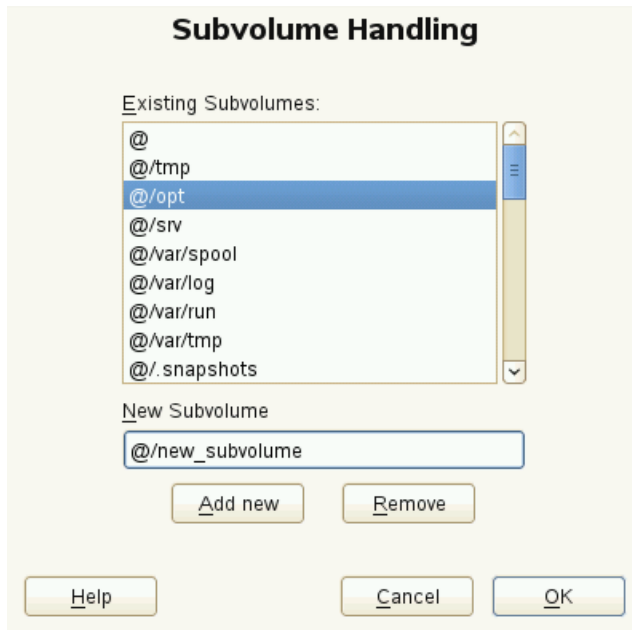
Managing Btrfs Subvolumes using YaST

Subvolumes of a Btrfs partition can be now managed with the YaST *Expert partitioner* module. You can add new or remove existing subvolumes.

Procedure 3.1: *Btrfs Subvolumes with YaST*

- 1** Start the YaST *Expert Partitioner* with *System > Partitioner*.
- 2** Choose *BTRFS* in the left *System View* pane.
- 3** Select the Btrfs partition whose subvolumes you need to manage and click *Edit*.
- 4** Click *Subvolume Handling*. You can see a list off all existing subvolumes of the selected Btrfs partition. You can notice a number of @/.snapshots/xyz/snapshot entries — each of these subvolumes belongs to one existing snapshot.
- 5** Depending on whether you want to add or remove subvolumes, do the following:
 - 5a** To remove a subvolume, select it from the list of *Existing Subvolumes* and click *Remove*.
 - 5b** To add a new subvolume, enter its name to the *New Subvolume* text field and click *Add new*.

Figure 3.2: *Btrfs Subvolumes in YaST Partitioner*



6 Confirm with *OK* and *Finish*.

7 Leave the partitioner with *Finish*.

3.1.3 Editing a Partition

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

1 Select the partition.

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

File System ID

Even if you do not want to format the partition at this stage, assign it a file system ID to ensure that the partition is registered correctly. Typical values are *Linux*, *Linux swap*, *Linux LVM*, and *Linux RAID*.

File System

To change the partition file system, click *Format Partition* and select file system type in the *File System* list.

openSUSE supports several types of filesystems. Btrfs is the Linux filesystem of choice because of its advanced features. It supports copy-on-write functionality, creating snapshots, multi-device spanning, subvolumes, and other useful techniques. ReiserFS, JFS, XFS, and Ext3 are journaling file systems. These file systems are able to restore the system very quickly after a system crash, utilizing write processes logged during the operation. Ext2 is not a journaling file system, but it is adequate for smaller partitions because it does not require much disk space for management.

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

WARNING: Changing the file system

Changing the file system and reformatting partitions irreversibly deletes all data from the partition.

Encrypt Device

If you activate the encryption, all data is written to the hard disk in encrypted form. This increases the security of sensitive data, but reduces the system speed, as the encryption takes some time to process. More information about the encryption of file systems is provided in Chapter 10, *Encrypting Partitions and Files* (↑Security Guide).

Mount Point

Specify the directory where the partition should be mounted in the file system tree. Select from YaST suggestions or enter any other name.

Fstab Options

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

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

If you prefer to mount the partition by its label, you need to define one in the *Volume label* text entry. For example, you could use the partition label `HOME` for a partition intended to mount to `/home`.

If you intend to use quotas on the file system, use the mount option *Enable Quota Support*. This must be done before you can define quotas for users in the YaST *User Management* module. For further information on how to configure user quota, refer to Section “Managing Quotas” (Chapter 10, *Managing Users with YaST*, ↑Start-Up).

3 Select *Finish* to save the changes.

NOTE: Resize Filesystems

To resize an existing file system, select the partition and use *Resize*. Note, that it is not possible to resize partitions while mounted. To resize partitions, unmount the relevant partition before running the partitioner.

3.1.4 Expert Options

After you select a hard disk device (like *sda*) in the *System View* pane, you can access the *Expert...* menu in the lower right part of the *Expert Partitioner* window. The menu contains the following commands:

Create New Partition Table

This option helps you create a new partition table on the selected device.

WARNING: Creating a New Partition Table

Creating a new partition table on a device irreversibly removes all the partitions and their data from that device.

Clone This Disk

This option helps you clone the device partition layout (but not the data) to other available disk devices.

3.1.5 Advanced Options

After you select the hostname of the computer (the top-level of the tree in the *System View* pane), you can access the *Configure...* menu in the lower right part of the *Expert Partitioner* window. The menu contains the following commands:

Configure iSCSI

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

Configure Multipath

Selecting this option helps you configure the multipath enhancement to the supported mass storage devices.

3.1.6 More Partitioning Tips

The following section includes a few hints and tips on partitioning that should help you make the right decisions when setting up your system.

TIP: Cylinder Numbers

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

3.1.6.1 Using swap

Swap is used to extend the available physical memory. It is then possible to use more memory than physical RAM available. The memory management system of kernels

before 2.4.10 needed swap as a safety measure. Then, if you did not have twice the size of your RAM in swap, the performance of the system suffered. These limitations no longer exist.

Linux uses a page called “Least Recently Used” (LRU) to select pages that might be moved from memory to disk. Therefore, running applications have more memory available and caching works more smoothly.

If an application tries to allocate the maximum allowed memory, problems with swap can arise. There are three major scenarios to look at:

System with no swap

The application gets the maximum allowed memory. All caches are freed, and thus all other running applications are slowed. After a few minutes, the kernel's out-of-memory kill mechanism activates and kills the process.

System with medium sized swap (128 MB–512 MB)

At first, the system slows like a system without swap. After all physical RAM has been allocated, swap space is used as well. At this point, the system becomes very slow and it becomes impossible to run commands from remote. Depending on the speed of the hard disks that run the swap space, the system stays in this condition for about 10 to 15 minutes until the out-of-memory kill mechanism resolves the issue. Note that you will need a certain amount of swap if the computer needs to perform a “suspend to disk”. In that case, the swap size should be large enough to contain the necessary data from memory (512 MB–1GB).

System with lots of swap (several GB)

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

If your system is not out of control, but needs more swap after some time, it is possible to extend the swap space online. If you prepared a partition for swap space, just add this partition with YaST. If you do not have a partition available, you may also just use a swap file to extend the swap. Swap files are generally slower than partitions,

but compared to physical ram, both are extremely slow so the actual difference is negligible.

Procedure 3.2: *Adding a Swap File Manually*

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

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

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

- 2** Initialize this swap file with the command

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

- 3** Activate the swap with the command

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

To disable this swap file, use the command

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

- 4** Check the current available swap spaces with the command

```
cat /proc/swaps
```

Note that at this point, it is only temporary swap space. After the next reboot, it is no longer utilized.

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

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

3.1.7 Partitioning and LVM

From the *Expert partitioner*, access the LVM configuration by clicking the *Volume Management* item in the *System View* pane. However, if a working LVM configuration already exists on your system, it is automatically activated upon entering the initial LVM configuration of a session. In this case, all disks containing a partition (belonging to an activated volume group) cannot be repartitioned. The Linux kernel cannot reread the modified partition table of a hard disk when any partition on this disk is in use. If you already have a working LVM configuration on your system, physical

repartitioning should not be necessary. Instead, change the configuration of the logical volumes.

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

WARNING: File System for Booting

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

3.2 LVM Configuration

This section briefly describes the principles behind the Logical Volume Manager (LVM) and its multipurpose features. In Section 3.2.2, “LVM Configuration with YaST” (page 84), learn how to set up LVM with YaST.

WARNING

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

3.2.1 The Logical Volume Manager

The LVM enables flexible distribution of hard disk space over several file systems. It was developed because sometimes the need to change the segmenting of hard disk space arises just after the initial partitioning has been done. Because it is difficult to modify partitions on a running system, LVM provides a virtual pool (volume group, VG for short) of memory space from which logical volumes (LVs) can be created as needed. The operating system accesses these LVs instead of the physical partitions.

Volume groups can occupy more than one disk, so that several disks or parts of them may constitute one single VG. This way, LVM provides a kind of abstraction from the physical disk space that allows its segmentation to be changed in a much easier and safer way than with physical repartitioning. Background information regarding physical partitioning can be found in Section 3.1.1, “Partition Types” (page 72) and Section 3.1, “Using the YaST Partitioner” (page 71).

Figure 3.3: *Physical Partitioning versus LVM*

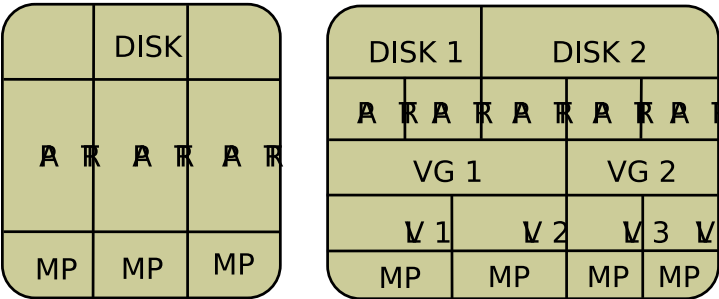


Figure 3.3, “Physical Partitioning versus LVM” (page 83) compares physical partitioning (left) with LVM segmentation (right). On the left side, one single disk has been divided into three physical partitions (PART), each with a mount point (MP) assigned so that the operating system can gain access. On the right side, two disks have been divided into two and three physical partitions each. Two LVM volume groups (VG 1 and VG 2) have been defined. VG 1 contains two partitions from DISK 1 and one from DISK 2. VG 2 contains the remaining two partitions from DISK 2. In LVM, the physical disk partitions that are incorporated in a volume group are called physical volumes (PVs). Within the volume groups, four LVs (LV 1 through LV 4) have been defined. They can be used by the operating system via the associated mount points. The border between different LVs do not need to be aligned with any partition border. See the border between LV 1 and LV 2 in this example.

LVM features:

- Several hard disks or partitions can be combined in a large logical volume.
- Provided the configuration is suitable, an LV (such as `/usr`) can be enlarged if free space is exhausted.
- With LVM, it is possible to add hard disks or LVs in a running system. However, this requires hot-swappable hardware.

- It is possible to activate a "striping mode" that distributes the data stream of a LV over several PVs. If these PVs reside on different disks, the read and write performance is enhanced, as with RAID 0.
- The snapshot feature enables consistent backups (especially for servers) of the running system.

With these features, LVM is ready for heavily used home PCs or small servers. LVM is well-suited for the user with a growing data stock (as in the case of databases, music archives, or user directories). This would allow file systems that are larger than the physical hard disk. Another advantage of LVM is that up to 256 LVs can be added. However, working with LVM is different from working with conventional partitions. Instructions and further information about configuring LVM is available in the official LVM HOWTO at <http://tldp.org/HOWTO/LVM-HOWTO/>.

Starting from Kernel version 2.6, LVM version 2 is available, which is backward-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 backward-compatible version. LVM 2 does not require any kernel patches. It makes use of the device mapper integrated in kernel 2.6. This kernel only supports LVM version 2. Therefore, when talking about LVM, this section always refers to LVM version 2.

3.2.2 LVM Configuration with YaST

The YaST LVM configuration can be reached from the YaST Expert Partitioner (see Section 3.1, "Using the YaST Partitioner" (page 71)) within the *Volume Management* item in the *System View* pane. The Expert Partitioner allows you to edit and delete existing partitions and also create new ones that need to be used with LVM. The first task is to create PVs that provide space to a volume group:

- 1 Select a hard disk from *Hard Disks*.
- 2 Change to the *Partitions* tab.
- 3 Click *Add* and enter the desired size of the PV on this disk.
- 4 Use *Do not format partition* and change the *File System ID* to *0x8E Linux LVM*. Do not mount this partition.


- 5 Repeat this procedure until you have defined all the desired physical volumes on the available disks.

3.2.2.1 Creating Volume Groups

If no volume group exists on your system, you must add one (see Figure 3.4, “Creating a Volume Group” (page 85)). It is possible to create additional groups by clicking on *Volume Management* in the *System View* pane, and then on *Add Volume Group*. One single volume group is usually sufficient.

- 1 Enter a name for the VG, for example, `system`.
- 2 Select the desired *Physical Extend Size*. This value defines the size of a physical block in the volume group. All the disk space in a volume group is handled in blocks of this size.
- 3 Add the prepared PVs to the VG by selecting the device and clicking on *Add*. Selecting several devices is possible by holding Ctrl while selecting the devices.
- 4 Select *Finish* to make the VG available to further configuration steps.





Figure 3.4: *Creating a Volume Group*

 **Add Volume Group**
Enter the name and physical extent size of the new volume group. [more](#)

Volume Group Name:

Physical Extent Size:
4 MB

Available Physical Volumes:

Device	Size	Enc	Type
/dev/sda1	101.94 MB		Linux native
/dev/sda2	62.75 MB		Linux native
/dev/sda6	20.00 GB		Linux native
/dev/sda7	30.01 GB		Linux native

Total size: 50.17 GB

Selected Physical Volumes:

Device	Size	Enc	Type
No entries.			

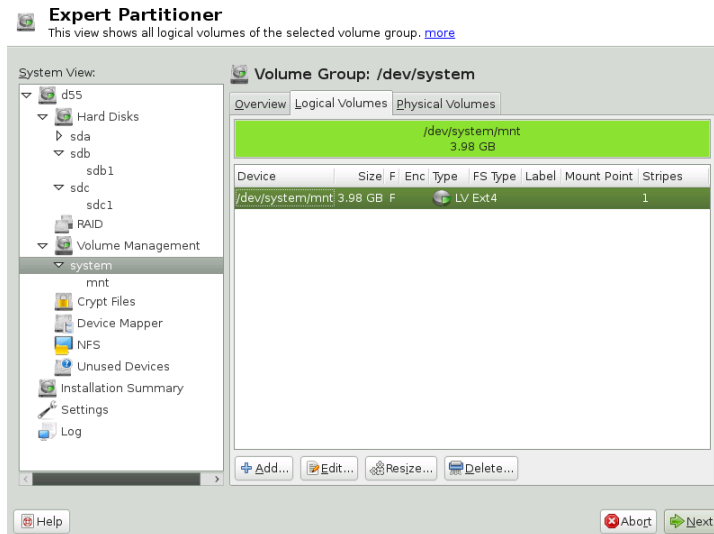
Resulting size: 0.00 B

If you have multiple volume groups defined and want to add or remove PVs, select the volume group in the *Volume Management* list and click *Resize*. In the following window, you can add or remove PVs to the selected volume group.

3.2.2.2 Configuring Logical Volumes

After the volume group has been filled with PVs, define the LVs which the operating system should use in the next dialog. Choose the current volume group and change to the *Logical Volumes* tab. *Add*, *Edit*, *Resize*, and *Delete* LVs as needed until all space in the volume group has been occupied. Assign at least one LV to each volume group.

Figure 3.5: *Logical Volume Management*



Click *Add* and go through the wizard-like pop-up that opens:

1. Enter the name of the LV. For a partition that should be mounted to `/home`, a self-explanatory name like `HOME` could be used.
2. Select the size and the number of stripes of the LV. If you have only one PV, selecting more than one stripe is not useful.
3. Choose the filesystem to use on the LV as well as the mount point.

By using stripes it is possible to distribute the data stream in the LV among several PVs (striping). However, striping a volume can only be done over different PVs,

each providing at least the amount of space of the volume. The maximum number of stripes equals to the number of PVs, where Stripe "1" means "no striping". Striping only makes sense with PVs on different hard disks, otherwise performance will decrease.

WARNING: Striping

YaST cannot, at this point, verify the correctness of your entries concerning striping. Any mistake made here is apparent only later when the LVM is implemented on disk.

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

3.3 Soft RAID Configuration

The purpose of RAID (redundant array of independent disks) is to combine several hard disk partitions into one large *virtual* hard disk to optimize performance and/or data security. 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. It is also more suitable for the parallel command processing. There are some RAID controllers that support IDE or SATA hard disks. Soft RAID provides the advantages of RAID systems without the additional cost of hardware RAID controllers. However, this requires some CPU time and has memory requirements that make it unsuitable for high performance computers.

With openSUSE®, you can combine several hard disks into one soft RAID system. RAID implies several strategies for combining several hard disks in a RAID system, each with different goals, advantages, and characteristics. These variations are commonly known as *RAID levels*.

Common RAID levels are:

RAID 0

This level improves the performance of your data access by spreading out blocks of each file across multiple disk drives. Actually, this is not really a RAID, because it does not provide data backup, but the name *RAID 0* for this type of sys-

tem is commonly used. With RAID 0, two or more hard disks are pooled together. Performance is enhanced, 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 one disk is destroyed, a copy of its contents is available on the other one. All disks but one could be damaged without endangering your data. However, if the damage is not detected, the damaged data can be mirrored to the undamaged disk. This could result in the same loss of data. The writing performance suffers in the copying process compared to using single disk access (10 to 20 % slower), but read access is significantly faster in comparison to any one of the normal physical hard disks. The reason is that the duplicate data can be parallel-scanned. Generally it can be said that Level 1 provides nearly twice the read transfer rate of single disks and almost the same write transfer rate as single disks.

RAID 5

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

RAID 6

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

RAID 10 (RAID 1+0)

This RAID implementation combines features of RAID 0 and RAID 1: the data are first mirrored in separate disk arrays, which are inserted into a new RAID 0; type array. In each RAID 1 sub-array, one disk can fail without any damage to the data. RAID 10 is used for database application where a huge load is expected.

Other RAID Levels

Several other RAID levels have been developed (RAID 2, RAID 3, RAID 4, RAIDn, RAID 10, RAID 0+1, RAID 30, RAID 50, etc.), some of them being proprietary implementations created by hardware vendors. These levels are not very common and therefore are not explained here.

3.3.1 Soft RAID Configuration with YaST

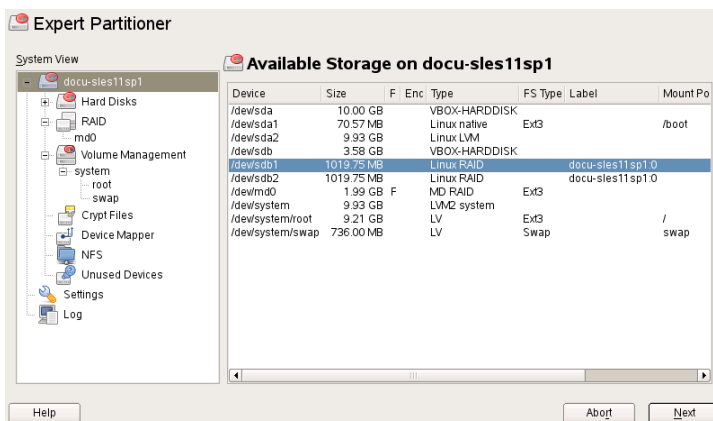
The YaST *RAID* configuration can be reached from the YaST Expert Partitioner, described in Section 3.1, “Using the YaST Partitioner” (page 71). This partitioning tool enables you to edit and delete existing partitions and create new ones to be used with soft RAID:

- 1 Select a hard disk from *Hard Disks*.
- 2 Change to the *Partitions* tab.
- 3 Click *Add* and enter the desired size of the raid partition on this disk.
- 4 Use *Do not Format the Partition* and change the *File System ID* to *0xFD Linux RAID*. Do not mount this partition.
- 5 Repeat this procedure until you have defined all the desired physical volumes on the available disks.

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 utilize partitions of the same size only. The RAID partitions should be located on different hard disks to decrease the risk of losing data if one is defective (RAID 1 and 5) and to optimize the performance of RAID 0. After creating all the partitions to use with RAID, click *RAID > Add RAID* to start the RAID configuration.

In the next dialog, choose between RAID levels 0, 1, 5, 6 and 10. Then, select all partitions with either the “Linux RAID” or “Linux native” type that should be used by the RAID system. No swap or DOS partitions are shown.

Figure 3.6: RAID Partitions



To add a previously unassigned partition to the selected RAID volume, first click the partition then *Add*. Assign all partitions reserved for RAID. Otherwise, the space on the partition remains unused. After assigning all partitions, click *Next* to select the available *RAID Options*.

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

3.3.2 Troubleshooting

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

Note that although you can access all data during the rebuild, you may encounter some performance issues until the RAID has been fully rebuilt.

3.3.3 For More Information

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

- `/usr/share/doc/packages/mdadm/Software-RAID.HOWTO.html`
- <http://raid.wiki.kernel.org>

Linux RAID mailing lists are available, such as <http://marc.info/?l=linux-raid>.

Part II. Advanced Administration

Snapshots/Rollback with Snapper

Being able to do file system snapshots on Linux providing the ability to do rollbacks is a feature that was often requested in the past. Snapper, in conjunction with the `Btrfs` file system now fills that gap.

`Btrfs`, a new copy-on-write file system for Linux, supports file system snapshots (a copy of the state of a subvolume at a certain point of time) of subvolumes (one or more separately mountable file systems within each physical partition). Snapper lets you manage these snapshots. Snapper comes with a command line and a YaST interface.

By default Snapper and `Btrfs` on openSUSE are set up to serve as an “undo tool” for system changes made with YaST and `zypper`. Before and after running a YaST module or `zypper`, a snapshot is created. Snapper lets you compare the two snapshots and provides means to revert the differences between the two snapshots. The tools also provide system backups by creating hourly snapshots of the system subvolumes.

4.1 Requirements

Since `Btrfs` is the only filesystem on openSUSE supporting snapshots, it is required on all partitions or subvolumes you want to “snapshot”.

4.1.1 snapshots and Disk Space

When a snapshot is created, both the snapshot and the original point to the same blocks in the file system. So, initially a snapshot does not occupy additional disk

space. If data in the original file system is modified, changed data blocks are copied while the old data blocks are kept for the snapshot. Therefore, a snapshot occupies the same amount of space as the data modified. So, over time, the amount of space a snapshot allocates, constantly grows. As a consequence, deleting files from a `Btrfs` file system containing snapshots may *not* free disk space!

NOTE: Snapshot Location

Snapshots always reside on the same partition or subvolume that has been snapshotted. It is not possible to store snapshots on a different partition or subvolume.

As a result, partitions containing snapshots need to be larger than “normal” partitions. The exact amount strongly depends on the number of snapshots you keep and the amount of data modifications. As a rule of thumb you should consider using twice the size than you normally would.

TIP: Freeing space / Disk Usage

In order to free space on a `Btrfs` partition containing snapshots you need to delete unneeded snapshots rather than files. Older snapshots occupy more space than recent ones.

Since the `df` does not show the correct disk usage on `Btrfs` file systems, you need to use the command `btrfs filesystem df MOUNT_POINT`. Displaying the amount of disk space a snapshot allocates is currently not supported by the `Btrfs` tools.

4.2 Using Snapper to Undo YaST and zypper Changes

If you set up the root partition with `Btrfs` during the installation, Snapper—pre-configured for doing rollbacks of YaST or zypper changes—will automatically be installed. Every time you start a YaST module or a zypper transaction, two snapshots are created: a “pre-snapshot” capturing the state of the filesystem before the start of the module and a “post-snapshot” after the module has been finished.

Using the YaST Snapper module or the `snapper` command line tool, you can undo the changes made by YaST/zypper by restoring files from the “pre-snapshot”. Comparing two snapshots the tools also allow you to see which files have been changed. You can also display the differences between two versions of a file (`diff`).

Since Linux is a multitasking system, processes other than YaST or zypper may modify data in the timeframe between the pre- and the post-snapshot. If this is the case, completely reverting to the pre-snapshot will also undo these changes by other processes. In most cases this would be unwanted—therefore it is strongly recommended to closely review the changes between two snapshots before starting the rollback. If there are changes from other processes you want to keep, select which files to roll back.

IMPORTANT: Limitations

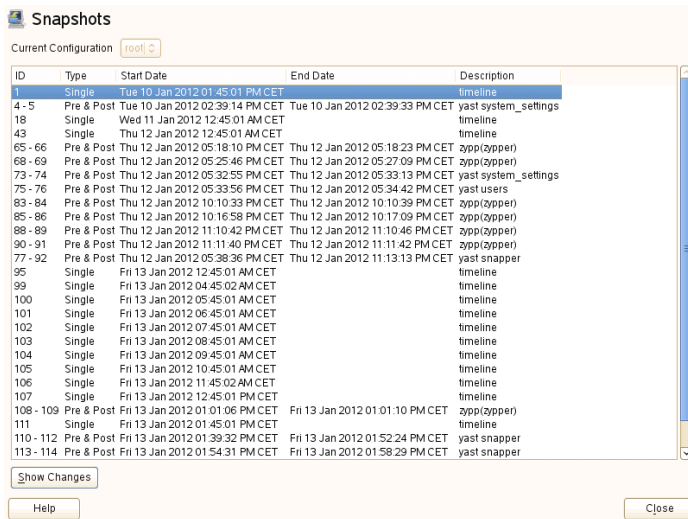
Make sure you know about Snapper's limitations before attempting to use its rollback mechanism. See Section 4.7, “Limitations” (page 108) for details.

NOTE: Storage Time of Snapshots

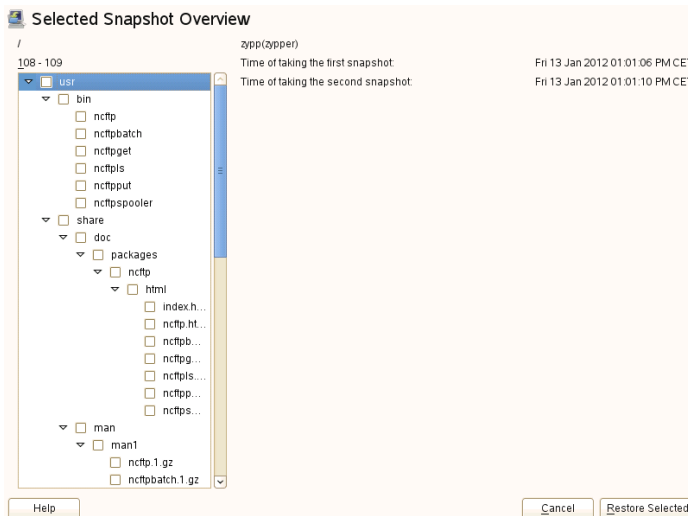
By default, the last 100 YaST and zypper snapshots are being kept. If this number is exceeded, the oldest snapshot(s) will be deleted.

Procedure 4.1: Undoing changes using the YaST Snapper module

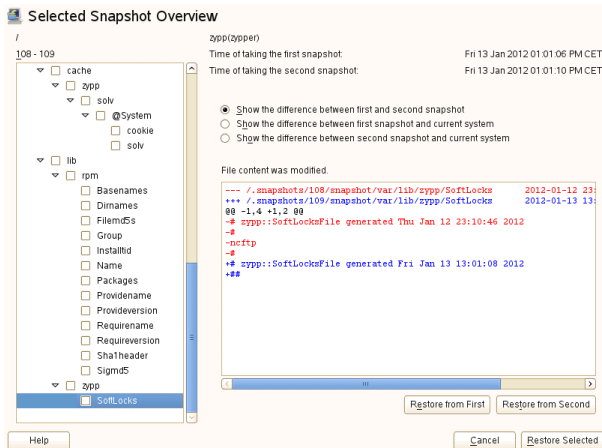
- 1 Start the *Snapper* module from the *Miscellaneous* section in YaST or by entering `yast2 snapper`.
- 2 Make sure *Current Configuration* is set to *root*. This is always the case unless you have manually added own Snapper configurations.
- 3 Choose a pair of pre- and post-snapshots from the list. Both, YaST and zypper snapshot pairs are of the type *Pre & Post*. YaST snapshots are labeled as `yast module_name` in the *Description column*; zypper snapshots are labeled `zypp (zypper)`.



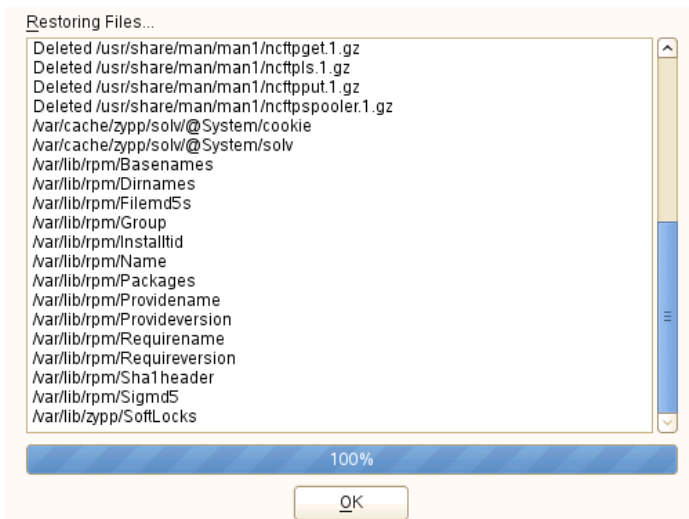
- Click *Show Changes* to open the list of files that differ between the two snapshots. The following image shows a list of files that have changed after having added the user `tester`.



- 5 Review the list of files. To display a “diff” between the pre- and post-version of a file, select it from the list. The following images shows the changes to `/etc/passwd` after having added the user `tester`.



- 6 To restore a set of files, select the relevant files or directories by ticking the respective checkbox. Click *Restore Selected* and confirm the action by clicking *Yes*.



To restore a single file, activate its diff view by clicking on its name. Click *Restore From First* and confirm your choice with *Yes*.

Procedure 4.2: Undoing changes using the snapper command

- 1 Get a list of yast and zypper snapshots by running `snapper list -t pre-post`. YaST snapshots are labeled as `yast module_name` in the *Description* column; zypper snapshots are labeled `zypp (zypper)`.

```
~ # snapper list -t pre-post
Pre # | Post # | Pre Date                | Post Date                | Description
-----+-----+-----+-----+-----
+
4      | 5       | Tue Jan 10 14:39:14 2012 | Tue Jan 10 14:39:33 2012 | yast system_settings
65     | 66      | Thu Jan 12 17:18:10 2012 | Thu Jan 12 17:18:23 2012 | zypp(zypper)
68     | 69      | Thu Jan 12 17:25:46 2012 | Thu Jan 12 17:27:09 2012 | zypp(zypper)
73     | 74      | Thu Jan 12 17:32:55 2012 | Thu Jan 12 17:33:13 2012 | yast system_settings
75     | 76      | Thu Jan 12 17:33:56 2012 | Thu Jan 12 17:34:42 2012 | yast users
77     | 92      | Thu Jan 12 17:38:36 2012 | Thu Jan 12 23:13:13 2012 | yast snapper
83     | 84      | Thu Jan 12 22:10:33 2012 | Thu Jan 12 22:10:39 2012 | zypp(zypper)
85     | 86      | Thu Jan 12 22:16:58 2012 | Thu Jan 12 22:17:09 2012 | zypp(zypper)
88     | 89      | Thu Jan 12 23:10:42 2012 | Thu Jan 12 23:10:46 2012 | zypp(zypper)
90     | 91      | Thu Jan 12 23:11:40 2012 | Thu Jan 12 23:11:42 2012 | zypp(zypper)
108    | 109     | Fri Jan 13 13:01:06 2012 | Fri Jan 13 13:01:10 2012 | zypp(zypper)
```

- 2 Get a list of changed files for a snapshot pair with `snapper status PRE..POST`. Files with content changes are marked with `c`, files that have been added are marked with `+` and deleted files are marked with `-`. The following example shows a snapshot pair for the installation of the package `ncftp`.

```
~ # snapper status 108..109
+... /usr/bin/ncftp
+... /usr/bin/ncftpbatch
+... /usr/bin/ncftpget
+... /usr/bin/ncftpls
[... ]
+... /usr/share/man/man1/ncftpspooler.1.gz
c... /var/cache/zypp/solv/@System/cookie
c... /var/cache/zypp/solv/@System/solv
c... /var/lib/rpm/Basenames
c... /var/lib/rpm/Dirnames
c... /var/lib/rpm/Filemd5s
c... /var/lib/rpm/Group
c... /var/lib/rpm/Installtid
c... /var/lib/rpm/Name
c... /var/lib/rpm/Packages
c... /var/lib/rpm/Providename
c... /var/lib/rpm/Provideversion
c... /var/lib/rpm/Requirename
c... /var/lib/rpm/Requireversion
c... /var/lib/rpm/Shalheader
c... /var/lib/rpm/Sigmd5
c... /var/lib/zypp/SoftLocks
```

- 3** To display the diff for a certain file, run `snapper diff PRE..POST FILE-NAME`. If you do not specify *FILENAME*, a diff for all files will be displayed.

```
~ # snapper diff 108..109 /var/lib/zypp/SoftLocks
--- /.snapshots/108/snapshot/var/lib/zypp/SoftLocks 2012-01-12
    23:15:22.408009164 +0100
+++ /.snapshots/109/snapshot/var/lib/zypp/SoftLocks 2012-01-13
    13:01:08.724009131 +0100
@@ -1,4 +1,2 @@
-# zypp::SoftLocksFile generated Thu Jan 12 23:10:46 2012
-#
-ncftp
-#
+# zypp::SoftLocksFile generated Fri Jan 13 13:01:08 2012
+##
```

- 4** To restore one or more files run `snapper -v undochange PRE..POST FILENAMES`. If you do not specify a *FILENAMES*, all changed files will be restored.

```
~ # snapper -v undochange 108..109
create:0 modify:16 delete:21
undoing change...
deleting /usr/share/man/man1/ncftpspooler.1.gz
deleting /usr/share/man/man1/ncftpput.1.gz
[...]
deleting /usr/bin/ncftpls
deleting /usr/bin/ncftpget
deleting /usr/bin/ncftpbatch
deleting /usr/bin/ncftp
modifying /var/cache/zypp/solv/@System/cookie
modifying /var/cache/zypp/solv/@System/solv
modifying /var/lib/rpm/Basenames
modifying /var/lib/rpm/Dirnames
modifying /var/lib/rpm/Filemd5s
modifying /var/lib/rpm/Group
modifying /var/lib/rpm/Installtid
modifying /var/lib/rpm/Name
modifying /var/lib/rpm/Packages
modifying /var/lib/rpm/Providename
modifying /var/lib/rpm/Provideversion
modifying /var/lib/rpm/Requirename
modifying /var/lib/rpm/Requireversion
modifying /var/lib/rpm/Shalheader
modifying /var/lib/rpm/Sigmd5
modifying /var/lib/zypp/SoftLocks
undoing change done
```

4.3 Using Snapper to Restore Files from Hourly Backups

Apart from the YaST and zypper snapshots, Snapper creates hourly snapshots of the system partition (/). You can use these backup snapshots to restore files that have accidentally been deleted or modified beyond recovery. By making use of Snapper's diff feature you can also find out which modifications have been made at a certain point of time.

Hourly backup snapshots are of the type `Single` and are marked with the description `timeline`. To restore files from these snapshots proceed as described in Procedure 4.1, “Undoing changes using the YaST *Snapper* module” (page 97) or Procedure 4.2, “Undoing changes using the `snapper` command” (page 100)

NOTE: Storage Time of Snapshots

By default, the last ten snapshots of each day are kept. Furthermore ten snapshots are kept for the last ten days, months, and years. For details see Example 4.1, “Example timeline configuration” (page 108).

4.4 Managing Snapshots

With Snapper you can not only create and delete snapshots, but also set and modify the metadata. All Snapper operations are carried out for a certain configuration. If you do not specify a configuration, the default one (`root`) is used. If you want to manage snapshots for your own configuration (see Section 4.6, “Creating Snapper Configurations” (page 106) for details), you need to specify it with the global `snapper` option `-c`:

```
snapper -c myconfig list
```

For details on `snapper` refer to its manpage (`man 1 snapper`).

4.4.1 Snapshot Metadata

Each snapshot consists of the snapshot itself and some metadata. With Snapper you can set and modify some metadata. The following metadata is available for each snapshot:

- **Type:** Snapshot type, see Section 4.4.1.1, “Snapshot Types” (page 103) for details. This data can not be changed.
- **Number:** Unique number of the snapshot. This data can not be changed.
- **Pre Number:** Specifies the number of the corresponding pre snapshot. For snapshots of type post only. This data can not be changed.
- **Description:** A description of the snapshot.
- **Userdata:** An extended description where you can specify custom data in the form or comma-separated key=value list: `reason=testing_stuff, user=tux`
- **Cleanup-Algorithm:** Cleanup-algorithm for the snapshot

4.4.1.1 Snapshot Types

Snapper knows three different types of snapshots: pre, post and single. Physically they do not differ, but Snapper handles them differently.

`pre`

Snapshot of a filesystem *before* a modification. Each `pre` snapshot has got a corresponding `post` snapshot. Used e.g. for the automatic YaST/zypper snapshots.

`post`

Snapshot of a filesystem *after* a modification. Each `post` snapshot has got a corresponding `pre` snapshot. Used e.g. for the automatic YaST/zypper snapshots.

`single`

Stand-alone snapshot. Used e.g. for the automatic hourly snapshots. This is the default type when creating snapshots.

4.4.1.2 Cleanup-algorithms

Snapper provides three algorithms to cleanup old snapshots. The algorithms are executed in a daily cron-job. The cleanup-frequency itself is defined in the Snapper configuration for the partition or subvolume (see Section 4.6.1, “Adjusting the Config File” (page 107) for details).

`number`

Deletes old snapshots when a certain snapshot count is reached.

timeline

Deletes old snapshots having passed a certain age, but keeps a number of hourly, daily, monthly and yearly snapshots.

empty-pre-post

Deletes pre/post snapshot pairs with empty diffs.

4.4.2 Creating Snapshots

Creating a snapshot is done by running `snapper create`. You should always specify a meaningful description with `-d` in order to later be able to identify its purpose.

```
snapper create -d "Before the Apache config cleanup"
```

Creates a stand-alone snapshot (type single) for the default (`root`) configuration with a description. Because no cleanup-algorithm is specified, the snapshot will never be deleted automatically.

```
snapper -c home create -d ".-file cleanup in ~tux"
```

Creates a stand-alone snapshot (type single) for a custom configuration named `home` with a description. Because no cleanup-algorithm is specified, the snapshot will never be deleted automatically.

```
snapper -c home create -d "daily HOME backup" -c timeline
```

Creates a stand-alone snapshot (type single) for a custom configuration named `home` with a description. The file will automatically be deleted when it meets the criteria specified for the timeline cleanup-algorithm in the configuration.

4.4.3 Modifying Snapshot Metadata

Snapper allows to modify the description, the cleanup algorithm and the userdata of a snapshot. All other metadata cannot be changed.

```
snapper modify -c "timeline" 10
```

Modifies the metadata of snapshot 10 for the default (`root`) configuration. The cleanup algorithm is set to `timeline`.

```
snapper -c home modify -d "daily backup" -c "" 120
```

Modifies the metadata of snapshot 120 for a custom configuration named `home`. A new description is set and the cleanup algorithm is unset.

4.4.4 Deleting Snapshots

To delete a snapshot, you need to know its number. Get it by running `snapper list`. To delete a snapshot, run `snapper delete NUMBER`. When deleting a pre snapshot, you should always delete its corresponding post snapshot (and vice-versa).

```
snapper delete 65
```

Deletes snapshot 65 for the default (`root`) configuration.

```
snapper -c home delete 89 90
```

Deletes snapshots 89 and 90 for a custom configuration named `home`.

```
snapper delete 65
```

```
snapper -c myconfig delete 89 90
```

TIP: Old Snapshots Occupy More Disk Space

If you delete snapshots in order to free space on your harddisk (see Section 4.1.1, “snapshots and Disk Space” (page 95) for details), make sure to delete old snapshots first. The older a snapshot is, the more disk space it occupies.

Snapshots are also automatically deleted by a daily cron-job. Refer to Section 4.4.1.2, “Cleanup-algorithms” (page 103) for details.

4.5 Disabling Snapper

If you have set up the root partition with `Btrfs` during the installation, Snapper automatically creates hourly snapshots of the system, as well as pre- and post-snapshots for YaST and zypper transactions. Each of these tasks can be disabled as follows:

Disabling Hourly Snapshots

Edit `/etc/snapper/configs/root` and set `TIMELINE_CREATE` to `no`:

```
TIMELINE_CREATE="no"
```

Disabling zypper snapshots

Uninstall the package `snapper-zypp-plugin`

Disabling YaST snapshots

Edit `/etc/sysconfig/yast2` and set `USE_SNAPPER` to `no`:

```
USE_SNAPPER="no"
```

4.6 Creating Snapper Configurations

The way Snapper behaves is defined in a config file that is specific for each partition or Btrfs subvolume. These config files reside under `/etc/snapper/configs/`. The default config installed with Snapper for the `/` directory is named `root`. It creates and manages the YaST and zypper snapshots as well as the hourly backup snapshot for `/`.

You may create your own configurations for other partitions formatted with Btrfs or existing subvolumes on a Btrfs partition. In the following example we will set up a Snapper configuration for backing up the webserver data residing on a separate, Btrfs-formatted partition mounted at `/srv/www`.

You can use either `snapper` itself or the YaST *Snapper* module to restore files from these snapshots. In YaST you need to select your *Current Configuration*, while you need to specify your config for `snapper` with the global switch `-c` (e.g. `snapper -c myconfig list`).

To create a new Snapper configuration, run `snapper create-config`:

```
snapper -c www-data❶ create-config  
/srv/www❷
```

- ❶ Name of config file.
- ❷ Mount point of the partition or Btrfs subvolume to snapshot.

This command will create a new config file `/etc/snapper/config-templates/www-data` with reasonable default values (taken from `/etc/snapper/config-templates/default`).

TIP: Config Defaults

Default values for a new config are taken from `/etc/snapper/config-templates/default`. To use your own set of defaults, create a copy of this file in the same directory and adjust it to your needs. To use it, specify the `-t` option with the `create-config` command:

```
snapper -c www-data create-config -t my_defaults /srv/www
```

4.6.1 Adjusting the Config File

To adjust the config file, you need to modify it with an editor. It contains key/value pairs in the form of *key=value*. You may only change the *value*.

SUBVOLUME

Mount point of the partition or subvolume to snapshot. Do not change.

FSTYPE

Filesystem type of the partition. Do not change.

NUMBER_CLEANUP

Defines whether to automatically delete old snapshots when the total snapshot count exceeds a number specified with `NUMBER_LIMIT` *and* an age specified with `NUMBER_MIN_AGE`. Valid values: `yes`, `no`

NOTE: Limit and Age

`NUMBER_LIMIT` and `NUMBER_MIN_AGE` are always evaluated both. Snapshots are only deleted when *both* conditions are met. If you always want to keep a certain number of snapshots regardless of their age, set `NUMBER_MIN_AGE` to 0. On the other hand, if you do not want to keep snapshots beyond a certain age, set `NUMBER_LIMIT` to 0.

NUMBER_LIMIT

Defines how many snapshots to keep if `NUMBER_CLEANUP` is set to `yes`.

NUMBER_MIN_AGE

Defines the minimum age in seconds a snapshot must have before it can automatically be deleted.

TIMELINE_CREATE

If set to `yes`, hourly snapshots are created. This is currently the only way to automatically create snapshots, therefore setting it to `yes` is strongly recommended. Valid values: `yes`, `no`

TIMELINE_CLEANUP

Defines whether to automatically delete old snapshots when the snapshot count exceeds a number specified with the `TIMELINE_LIMIT_*` options *and* an age specified with `TIMELINE_MIN_AGE`. Valid values: `yes`, `no`

`TIMELINE_MIN_AGE`

Defines the minimum age in seconds a snapshot must have before it can automatically be deleted.

`TIMELINE_LIMIT_HOURLY`, `TIMELINE_LIMIT_DAILY`,
`TIMELINE_LIMIT_MONTHLY`, `TIMELINE_LIMIT_YEARLY`

Number of snapshots to keep for hour, day, month, year.

Example 4.1: Example timeline configuration

```
TIMELINE_CREATE="yes"
TIMELINE_CLEANUP="yes"
TIMELINE_MIN_AGE="1800"
TIMELINE_LIMIT_HOURLY="10"
TIMELINE_LIMIT_DAILY="10"
TIMELINE_LIMIT_MONTHLY="10"
TIMELINE_LIMIT_YEARLY="10"
```

This example configuration enables hourly screenshots which are automatically cleaned up. `TIMELINE_MIN_AGE` and `TIMELINE_LIMIT_*` are always evaluated both. In this example, the minimum age of a snapshot, before it can be deleted is set to 30 minutes (1800 seconds). Since we create hourly snapshots, this ensures that only the latest snapshots are kept. If `TIMELINE_LIMIT_DAILY` is set to not zero, this means that the first screenshot of the day is kept, too.

Snapshots to be Kept

- Hourly: The last ten snapshots that have been made
- Daily: The first daily snapshot that has been made is kept for the last ten days.
- Monthly: The first snapshot made on the last day of the month is kept for the last ten months.
- Yearly: The first snapshot made on Dec 31st is kept for the last ten years.

4.7 Limitations

Although being ready for production, `Btrfs` as well as `Snapper` are constantly developed further. The following limitations exist at the moment. It is planned to solve these issues in future releases.

4.7.1 Data Consistency

There is no mechanism to ensure data consistency when creating snapshot. Whenever a file is written (e.g. a database) at the same time the snapshot is created, it will result in a broken or partly written file. Restoring such a file will cause problems. Therefore it is strongly recommended to *always* closely review the list of changed files and their diffs. Only restore files that really need to belonging to the action you want to roll back.

4.7.2 Reverting User Additions

Usually `/home` resides on a separate partition. Such a separate partition is not part of the default configuration for doing YaST rollbacks. Therefore the user's home partition will not be deleted when reverting a user addition using Snapper. It is strongly recommended to use the YaST *User and Group Management* tool to remove users.

4.7.3 No Rollback on `/boot` and Boot Loader Changes

Currently openSUSE cannot boot from `Btrfs` partitions. Therefore a separate partition for `/boot` is created upon the installation when using `Btrfs` for the system partition. Since `/boot` does not support snapshots, the following restrictions apply for YaST/zypper rollbacks:

no rollback for any configuration changes on the bootloader

The only file that can be rolled back is the bootloader configuration file in `/etc`.

The main configuration files reside under `/boot` and cannot be rolled back.

no complete rollback for Kernel installations

The Kernel itself and its `initrd` are installed in the `/boot` partition, whereas Kernel modules or sources are installed in `/var/lib` and `/usr/src`, respectively. Whatsmore, each Kernel installation also changes the bootloader configuration files in `/boot`. So whenever you do a rollback that involves undoing a Kernel installation, you need to manually remove the Kernel and its `initrd` from `/boot` and adjust the bootloader configuration by removing the boot entry for the Kernel.

4.8 Frequently Asked Questions

Why does Snapper never show changes in `/var/log`, `/tmp` and other directories?

For some directories we decided to disable snapshotting, e.g. `/var/log` since reverting logs makes searching for problems difficult. To exclude a path from snapshotting we create a subvolume for that path. The following mount points are excluded from snapshotting on openSUSE:

- `/opt`
- `/srv`
- `/tmp`
- `/var/crash`
- `/var/log`
- `/var/run`
- `/var/spool`
- `/var/tmp`

Can I boot a Snapshot from the Bootloader?

This is currently not possible. The bootloader on openSUSE currently does not support booting from a `Btrfs` partition.

Remote Access with VNC

Virtual Network Computing (VNC) enables you to control a remote computer via a graphical desktop (as opposed to a remote shell access). VNC is platform-independent and lets you access the remote machine from any operating system.

openSUSE supports two different kinds of VNC sessions: One-time sessions that “live” as long as the VNC connection from the client is kept up, and persistent sessions that “live” until they are explicitly terminated.

NOTE: Session Types

A machine can offer both kinds of sessions simultaneously on different ports, but an open session cannot be converted from one type to the other.

5.1 One-time VNC Sessions

A one-time session is initiated by the remote client. It starts a graphical login screen on the server. This way you can choose the user which starts the session and, if supported by the login manager, the desktop environment. Once you terminate the client connection to such a VNC session, all applications started within that session will be terminated, too. One-time VNC sessions cannot be shared, but it is possible to have multiple sessions on a single host at the same time.

Procedure 5.1: *Enabling One-time VNC Sessions*

- 1 Start *YaST > Network Services > Remote Administration (VNC)*.

- 2 Check *Allow Remote Administration*.
- 3 If necessary, also check *Open Port in Firewall* (for example, when your network interface is configured to be in the External Zone). If you have more than one network interface, restrict opening the firewall ports to a specific interface via *Firewall Details*.
- 4 Confirm your settings with *Finish*.
- 5 In case not all needed packages are available yet, you need to approve the installation of missing packages.

NOTE: Available Configurations

The default configuration on openSUSE serves sessions with a resolution of 1024x768 pixels at a color depth of 16-bit. The sessions are available on ports 5901 for “regular” VNC viewers (equivalent to VNC display 1) and on port 5801 for Web browsers.

Other configurations can be made available on different ports.

VNC display numbers and X display numbers are independent in one-time sessions. A VNC display number is manually assigned to every configuration that the server supports (:1 in the example above). Whenever a VNC session is initiated with one of the configurations, it automatically gets a free X display number.

5.1.1 Initiating a One-time VNC Session

To initiate a one-time VNC session, a VNC viewer must be installed on the client machine. The standard viewer on SUSE Linux products is `vncviewer`, provided by the package `tightvnc`. You may also view a VNC session using your Web browser and a Java applet.

To start your VNC viewer and initiate a session with the server's default configuration, use the command:

```
vncviewer jupiter.example.com:1
```

Instead of the VNC display number you can also specify the port number with two colons:

```
vncviewer jupiter.example.com::5901
```

Alternatively use a Java-capable Web browser to view the VNC session by entering the following URL: `http://jupiter.example.com:5801`

5.1.2 Configuring One-time VNC Sessions

You can skip this section, if you do not need or want to modify the default configuration.

One-time VNC sessions are started via the `xinetd` daemon. A configuration file is located at `/etc/xinetd.d/vnc`. By default it offers six configuration blocks: three for VNC viewers (`vnc1` to `vnc3`), and three serving a Java applet (`vnchttpd1` to `vnchttpd3`). By default only `vnc1` and `vnchttpd1` are active.

To activate a configuration, comment the line `disable = yes` with a `#` character in the first column, or remove that line completely. To deactivate a configuration uncomment or add that line.

The `Xvnc` server can be configured via the `server_args` option—see `Xvnc --help` for a list of options.

When adding custom configurations, make sure they are not using ports that are already in use by other configurations, other services, or existing persistent VNC sessions on the same host.

Activate configuration changes by entering the following command:

```
rcxinetd reload
```

IMPORTANT: Firewall and VNC Ports

When activating Remote Administration as described in Procedure 5.1, “Enabling One-time VNC Sessions” (page 111), the ports `5801` and `5901` are opened in the firewall. If the network interface serving the VNC sessions is protected by a firewall, you need to manually open the respective ports when activating additional ports for VNC sessions. See Chapter 13, *Masquerading and Firewalls* (↑Security Guide) for instructions.

5.2 Persistent VNC Sessions

A persistent VNC session is initiated on the server. The session and all applications started in this session run regardless of client connections until the session is terminated.

A persistent session can be accessed from multiple clients simultaneously. This is ideal for demonstration purposes where one client has full access and all other clients have view-only access. Another usecase are trainings where the trainer might need access to the trainee's desktop. However, most of the times you probably do not want to share your VNC session.

In contrast to one-time sessions that start a display manager, a persistent session starts a ready-to-operate desktop that runs as the user that started the VNC session.

Access to persistent sessions is protected by two possible types of passwords:

- a regular password that grants full access or
- an optional view-only password that grants a non-interactive (view-only) access.

A session can have multiple client connections of both kinds at once.

Procedure 5.2: *Starting a Persistent VNC Session*

- 1 Open a shell and make sure you are logged in as the user that should own the VNC session.
- 2 If the network interface serving the VNC sessions is protected by a firewall, you need to manually open the port used by your session in the firewall. If starting multiple sessions you may alternatively open a range of ports. See Chapter 13, *Masquerading and Firewalls* (↑Security Guide) for details on how to configure the firewall.

`vncserver` uses the ports 5901 for display :1, 5902 for display :2, and so on. For persistent sessions, the VNC display and the X display usually have the same number.

- 3 To start a session with a resolution of 1024x769 pixel and with a color depth of 16-bit, enter the following command:

```
vncserver -geometry 1024x768 -depth 16
```

The `vncserver` command picks an unused display number when none is given and prints out its choice. See `man 1 vncserver` for more options.

When running `vncviewer` for the first time, it asks for a password for full access to the session. If needed, you can also provide a password for view-only access to the session.

The password(s) you are providing here are also used for future sessions started by the same user. They can be changed with the `vncpasswd` command.

IMPORTANT: Security Considerations

Make sure to use strong passwords of significant length (eight or more characters). Do not share these passwords.

VNC connections are unencrypted, so people who can sniff the network(s) between the two machines can read the password when it gets transferred at the beginning of a session.

To terminate the session shut down the desktop environment that runs inside the VNC session from the VNC viewer as you would shut it down if it was a regular local X session.

If you prefer to manually terminate a session, open a shell on the VNC server and make sure you are logged in as the user that owns the VNC session you want to terminate. Run the following command to terminate the session that runs on display `:1`:

```
vncserver -kill :1
```

5.2.1 Connecting to a Persistent VNC Session

To connect to a persistent VNC session, a VNC viewer must be installed. The standard viewer on SUSE Linux products is `vncviewer`, provided by the package `tightvnc`. You may also view a VNC session using your Web browser and a Java applet.

To start your VNC viewer and connect to display `:1` of the VNC server, use the command

```
vncviewer jupiter.example.com:1
```

Instead of the VNC display number you can also specify the port number with two colons:

```
vncviewer jupiter.example.com::5901
```

Alternatively use a Java-capable Web browser to view the VNC session by entering the following URL: `http://jupiter.example.com:5801`

5.2.2 Configuring Persistent VNC Sessions

Persistent VNC sessions can be configured by editing `$HOME/.vnc/xstartup`. By default this shell script starts an `xterm` and the `twm` Window Manager. To start either GNOME or KDE instead, replace the line starting `twm` with one of the following:

```
/usr/bin/gnome      # GNOME  
/usr/bin/startkde   # KDE
```

NOTE: One Configuration for Each User

Persistent VNC sessions are configured in a single per-user configuration. Multiple sessions started by a user will all use the same startup and password files.

Part III. System

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

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

openSUSE for the 64-bit platforms amd64 and Intel 64 is designed so that existing 32-bit applications run in the 64-bit environment “out-of-the-box.” This support means that you can continue to use your preferred 32-bit applications without waiting for a corresponding 64-bit port to become available.

6.1 Runtime Support

IMPORTANT: Conflicts between Application Versions

If an application is available both for 32-bit and 64-bit environments, parallel installation of both versions is bound to lead to problems. In such cases, decide on one of the two versions and install and use this.

An exception to this rule is PAM (pluggable authentication modules). openSUSE uses PAM in the authentication process as a layer that mediates between user and application. On a 64-bit operating system that also runs

32-bit applications it is necessary to always install both versions of a PAM module.

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 that you would normally expect to find under `/lib` and `/usr/lib` are now found under `/lib64` and `/usr/lib64`. This means that there is space for the 32-bit libraries under `/lib` and `/usr/lib`, so the filename for both versions can remain unchanged.

Subdirectories of 32-bit `/lib` directories which contain data content that does not depend on the word size are not moved. This scheme conforms to LSB (Linux Standards Base) and FHS (File System Hierarchy Standard).

6.2 Software Development

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

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

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 `x86_64` system with `x86` as the second architecture.

1 Use the 32-bit compiler:

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

2 Instruct the linker to process 32-bit objects (always use `gcc` as the linker frontend):

```
LD="gcc -m32"
```

3 Set the assembler to generate 32-bit objects:

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

4 Specify linker flags, such as the location of 32-bit libraries, for example:

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

5 Specify the location for the 32-bit object code libraries:

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

6 Specify the location for the 32-bit X libraries:

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

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

```
CC="gcc -m32"
LDFLAGS="-L/usr/lib;"
./configure --prefix=/usr --libdir=/usr/lib --x-libraries=/usr/lib
make
make install
```

6.4 Kernel Specifications

The 64-bit kernels for x86_64 offer both a 64-bit and a 32-bit kernel ABI (application binary interface). The latter is identical with the ABI for the corresponding 32-bit kernel. This means that the 32-bit application can communicate with the 64-bit kernel in the same way as with the 32-bit kernel.

The 32-bit emulation of system calls for a 64-bit kernel does not support all the APIs used by system programs. This depends on the platform. For this reason, a small number of applications, like `lspci`, must be compiled.

A 64-bit kernel can only load 64-bit kernel modules that have been specially compiled for this kernel. It is not possible to use 32-bit kernel modules.

TIP: Kernel-loadable Modules

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.

Booting a Linux System

Booting a Linux system involves different components and tasks. The hardware itself is initialized by the BIOS or the UEFI, which starts the Kernel by means of a boot loader. After this point, the boot process is completely controlled by the operating system and handled by `systemd`. `systemd` provides a set of “targets” that boot setups for everyday usage, maintenance or emergencies.

7.1 The Linux Boot Process

The Linux boot process consists of several stages, each represented by a different component. The following list briefly summarizes the boot process and features all the major components involved:

1. **BIOS/UEFI** After turning on the computer, the BIOS or the UEFI initializes the screen, and keyboard and tests the main memory. Up to this stage, the machine does not access any mass storage media. Subsequently, the information about the current date, time, and the most important peripherals are loaded from the CMOS values. When the first hard disk and its geometry are recognized, the system control passes from the BIOS to the boot loader. If the BIOS supports network booting, it is also possible to configure a boot server that provides the boot loader. On x86 systems, PXE boot is needed. Other architectures commonly use the BOOTP protocol to get the boot loader.
2. **Boot Loader** The first physical 512-byte data sector of the first hard disk is loaded into the main memory and the *boot loader* that resides at the beginning of

this sector takes over. The commands executed by the boot loader determine the remaining part of the boot process. Therefore, the first 512 bytes on the first hard disk are referred to as the *Master Boot Record* (MBR). The boot loader then passes control to the actual operating system, in this case, the Linux Kernel. More information about GRUB, the Linux boot loader, can be found in Chapter 9, *The Boot Loader GRUB* (page 149). For a network boot, the BIOS acts as the boot loader. It gets the boot image from the boot server and starts the system. This is completely independent of local hard disks.

3. **Kernel and `initramfs`** To pass system control, the boot loader loads both the Kernel and an initial RAM-based file system (`initramfs`) into memory. The contents of the `initramfs` can be used by the Kernel directly. `initramfs` contains a small executable called `init` that handles the mounting of the real root file system. If special hardware drivers are needed before the mass storage can be accessed, they must be in `initramfs`. For more information about `initramfs`, refer to Section 7.1.1, “`initramfs`” (page 125). If the system does not have a local hard disk, the `initramfs` must provide the root file system for the Kernel. This can be done with the help of a network block device like iSCSI or SAN, but it is also possible to use NFS as the root device.

NOTE: The `init` Process Naming

Two different programs are commonly named “`init`”:

- a. the `initramfs` process mounting the root filesystem
- b. the operating system process setting up the system

In this chapter we will therefore refer to them as “`init on initramfs`” and “`systemd`”, respectively.

4. **`init on initramfs`** This program performs all actions needed to mount the proper root file system. It provides Kernel functionality for the needed file system and device drivers for mass storage controllers with `udev`. After the root file system has been found, it is checked for errors and mounted. If this is successful, the `initramfs` is cleaned and the `systemd` daemon on the root file system is executed. For more information about `init on initramfs`, refer to Section 7.1.2, “`init on initramfs`” (page 126). Find more information about `udev` in Chapter 12, *Dynamic Kernel Device Management with `udev`* (page 199).

5. **systemd** By starting services and mounting file systems `systemd` handles the actual booting of the system. `systemd` is described in Chapter 8, *The systemd daemon* (page 129).

7.1.1 `initramfs`

`initramfs` is a small `cpio` archive that the Kernel can load into a RAM disk. It provides a minimal Linux environment that enables the execution of programs before the actual root file system is mounted. This minimal Linux environment is loaded into memory by BIOS or UEFI routines and does not have specific hardware requirements other than sufficient memory. The `initramfs` archive must always provide an executable named `init` that executes the `systemd` daemon on the root file system for the boot process to proceed.

Before the root file system can be mounted and the operating system can be started, the Kernel needs the corresponding drivers to access the device on which the root file system is located. These drivers may include special drivers for certain kinds of hard drives or even network drivers to access a network file system. The needed modules for the root file system may be loaded by `init` on `initramfs`. After the modules are loaded, `udev` provides the `initramfs` with the needed devices. Later in the boot process, after changing the root file system, it is necessary to regenerate the devices. This is done by the `systemd` unit `udev.service` with the command `udevtrigger`.

If you need to change hardware (for example hard disks) in an installed system and this hardware requires different drivers to be present in the Kernel at boot time, you must update `initramfs`. This is done by calling `mkinitrd`. Calling `mkinitrd` without any argument creates an `initramfs`. Calling `mkinitrd -R` creates an `init` executable. In openSUSE®, the modules to load are specified by the variable `INITRD_MODULES` in `/etc/sysconfig/kernel`. After installation, this variable is automatically set to the correct value. The modules are loaded in exactly the order in which they appear in `INITRD_MODULES`.

IMPORTANT: Updating `initramfs` or `init`

The boot loader loads `initramfs` or `init` in the same way as the Kernel. It is not necessary to re-install GRUB after updating `initramfs` or `init`, because GRUB searches the directory for the right file when booting.

7.1.2 init on `initramfs`

The main purpose of `init` on `initramfs` is to prepare the mounting of and access to the real root file system. Depending on your system configuration, `init` on `initramfs` is responsible for the following tasks.

Loading Kernel Modules

Depending on your hardware configuration, special drivers may be needed to access the hardware components of your computer (the most important component being your hard drive). To access the final root file system, the Kernel needs to load the proper file system drivers.

Providing Block Special Files

For each loaded module, the Kernel generates device events. `udev` handles these events and generates the required special block files on a RAM file system in `/dev`. Without those special files, the file system and other devices would not be accessible.

Managing RAID and LVM Setups

If you configured your system to hold the root file system under RAID or LVM, `init` on `initramfs` sets up LVM or RAID to enable access to the root file system later. Find information about RAID and LVM in Chapter 3, *Advanced Disk Setup* (page 71).

Managing Network Configuration

If you configured your system to use a network-mounted root file system (mounted via NFS), `init` on `initramfs` must make sure that the proper network drivers are loaded and that they are set up to allow access to the root file system.

If the file system resides on a network block device like iSCSI or SAN, the connection to the storage server is also set up by `init` on `initramfs`.

When `init` on `initramfs` is called during the initial boot as part of the installation process, its tasks differ from those mentioned above:

Finding the Installation Medium

When starting the installation process, your machine loads an installation Kernel and a special `init` containing the YaST installer. The YaST installer is running in a RAM file system and needs to have information about the location of the installation medium in order to access it for installing the operating system.

Initiating Hardware Recognition and Loading Appropriate Kernel Modules

As mentioned in Section 7.1.1, “`initramfs`” (page 125), the boot process starts with a minimum set of drivers that can be used with most hardware configurations. `init` starts an initial hardware scanning process that determines the set of drivers suitable for your hardware configuration. The names of the modules needed for the boot process are written to `INITRD_MODULES` in `/etc/sysconfig/kernel`. These names are used to generate a custom `initramfs` that is needed to boot the system. If the modules are not needed for boot but for coldplug, the modules are written to `/etc/sysconfig/hardware/hwconfig-*`. All devices that are described with configuration files in this directory are initialized in the boot process.

Loading the Installation System

As soon as the hardware is properly recognized, the appropriate drivers are loaded. The `udev` program creates the special device files and `init` starts the installation system with the YaST installer.

Starting YaST

Finally, `init` starts YaST, which starts package installation and system configuration.

The `systemd` daemon

The program `systemd` is the process with process ID 1. It is responsible for initializing the system in the required way. `systemd` is started directly by the Kernel and resists signal 9, which normally kills processes. All other programs are either started directly by `systemd` or by one of its child processes.

NOTE: Using System V init rather than `systemd`

Being fully compatible, both `systemd` and System V init (SysV init) can act as a drop-in replacement for each other. If you prefer to use the SysV init rather than `systemd`, press F5 on the boot screen and choose *System V*. To make this a permanent choice, install the package `sysvinit-init` which will replace the `systemd-init` package.

Starting with openSUSE 12 `systemd` is a replacement for the popular SysV init daemon. `systemd` is fully compatible with SysV init (by supporting init scripts). One of the main advantages of `systemd` is the fact that it considerably speeds up boot time by aggressively paralleling service starts. Furthermore, `systemd` only starts a service when it is really needed. For example, the printing daemon `cupsd` is not started during boot time, but rather when a user prints a document for the first time after having booted the system. `systemd` also supports Kernel Control Groups (cgroups), snapshotting and restoring the system state and more. See <http://www.freedesktop.org/wiki/Software/systemd/> for details.

8.1 Basic Usage

The SysV init system utilized several different commands to handle services— the `init` scripts, `insserv`, `telinit` and others. `systemd` makes it easier to manage services, since there is only one command to memorize for the majority of service handling tasks: `systemctl`. It uses the command plus subcommand notation like `git` or `zypper`:

```
systemctl [general OPTIONS] subcommand [subcommand OPTIONS]
```

See `man 1 systemctl` for a complete manual.

TIP: Terminal Output and Bash Completion

If the output goes to a terminal (and not to a pipe or a file, for example) `systemd` commands send long output to a pager by default. Use the `--no-pager` option to turn off paging mode.

`systemd` also supports bash-completion, allowing you to enter the first letters of a subcommand and then hit `→|` to automatically complete it. This feature is only available in the `bash` Shell and requires the installation of the package `bash-completion`.

8.1.1 Managing Services in a Running System

Subcommands for managing services are the same as for managing a service with SysV init (`start`, `stop`, ...). The general syntax for service management commands is as follows:

`systemd`

```
systemctl reload|restart|start|status|stop|... <my_service(s)>.service
```

SysV init

```
rc<my_service(s)> reload|restart|start|status|stop|...
```

`systemd` allows to manage several services in one go. Instead of executing init scripts one after the other as with SysV init, execute a command like the following:

```
systemctl start <my_1st_service>.service <my_2nd_service>.service
```

The following table shows a list of the most important service management commands for `systemd` and SysV `init`:

Table 8.1: *Service Management Commands*

Task	systemd Subcommand	SysV init Subcommand
Starting	<code>start</code>	<code>start</code>
Stopping	<code>stop</code>	<code>stop</code>
Restarting <i>Shuts down services and starts them afterwards. If a service is not yet running it will be started.</i>	<code>restart</code>	<code>restart</code>
Restarting conditionally <i>Restarts services if they are currently running. Does nothing for services that are not running.</i>	<code>try-restart</code>	<code>try-restart</code>
Reloading <i>Tells services to reload their configuration files without interrupting operation. Use case: Tell Apache to reload a modified <code>httpd.conf</code> configuration file. Note that not all services support reloading.</i>	<code>reload</code>	<code>reload</code>
Reloading or restarting <i>Reloads services if reloading is supported, otherwise restarts them. Services is not yet running it will be started.</i>	<code>reload-or-restart</code>	n/a

Task	systemd Subcom- mand	SysV init Subcom- mand
Reloading or restarting conditionally <i>Reloads services if reloading is supported, otherwise restarts them if currently running. Does nothing for services that are not running.</i>	<code>reload-or-try-restart</code>	n/a
Getting detailed status information <i>Lists information about the status of services. The <code>systemd</code> command shows details such as description, executable, status, cgroup, and messages last issued by a service (see Section 8.1.2, “Debugging Services” (page 132)). The level of details displayed with the SysV init differs from service to service.</i>	<code>status</code>	<code>status</code>
Getting short status information <i>Shows whether services are active or not.</i>	<code>is-active</code>	<code>status</code>

8.1.2 Debugging Services

By default, `systemd` is not overly verbose. If a service was started successfully, no output will be produced. In case of a failure, a short error message will be displayed. However, `systemctl status` provides means to debug start-up as well as operation of a service.

`systemd` comes with its own logging mechanism (“The Journal”) that logs system messages. This allows to display the service messages together with status messages. The `status` command works similar to `tail` and can also display the log messages in different formats, making it a powerful debugging tool.

Show Service Start-Up Failure

Whenever a service fails to start, use `systemctl status <my_service>.service` to get a detailed error message:

```
www.example.com: ~ # systemctl start apache2.service
Job failed. See system journal and 'systemctl status' for details.
www.example.com: ~ # systemctl status apache2.service
    Loaded: loaded (/lib/systemd/system/apache2.service; disabled)
    Active: failed (Result: exit-code) since Mon, 04 Jun 2012 16:52:26
    +0200; 29s ago
    Process: 3088 ExecStart=/usr/sbin/start_apache2 -D SYSTEMD -k start
    (code=exited, status=1/FAILURE)
    CGroup: name=systemd:/system/apache2.service

Jun 04 16:52:26 g144 start_apache2[3088]: httpd2-prefork: Syntax error
on line
205 of /etc/apache2/httpd.conf: Syntax error on li...alHost>
```

Show Last *n* Service Messages

The default behavior of the `status` subcommand is displaying the last ten messages a service issued. To change the number of messages to show, use the `--lines=n` parameter,

```
systemctl status ntp.service
systemctl --lines=20 status ntp.service
```

Show Service Messages in Append Mode

To display a “live stream” of service messages, use the `--follow` option, which works just like `tail -f`:

```
systemctl --follow status ntp.service
```

Messages Output Format

The `--output=mode` parameter allows to change the output format of service messages. The most important modes available are:

`short`

The default format. Shows the log messages with a human readable time-stamp.

`verbose`

Full output with all fields.

`cat`

Terse output without time stamps.

8.1.3 Permanently Enabling/Disabling Services

The service management commands mentioned in the previous section let you manipulate services for the current session. `systemd` also lets you permanently enable or disable services, so they are automatically started when requested or are always unavailable. You can either do this by using YaST, or on the command line.

8.1.3.1 Enabling/Disabling Services on the Command Line

The following table shows a list of enabling and disabling commands for `systemd` and SysV init:

IMPORTANT: Service Start

When enabling a service on the command line, it is not started automatically. It is scheduled to be started with the next system start-up or runlevel/target change. In order to immediately start a service after having enabled it, explicitly run `systemctl start <my_service>.service` or `rc<my_service> start`.

Table 8.2: *Commands for Enabling and Disabling Services*

Task	systemd subcommand	SysV init sub-command
Enabling	<code>systemctl enable <my_service(s)>.service</code>	<code>insserv <my_service(s)></code>
Disabling	<code>systemctl disable <my_service(s)>.service</code>	<code>insserv -r <my_service(s)></code>
Checking <i>Shows whether a service is enabled or not.</i>	<code>systemctl is-enabled <my_service>.service</code>	n/a

Task	systemd subcommand	SysV init sub-command
Re-Enabling <i>Similar to restarting a service, this commands first disables and then enables a service. Useful to re-enable a service with its defaults.</i>	<code>systemctl reenab <my_service>.service</code>	n/a
Masking <i>When “disabling” a service, it can still be started manually. To completely disable a service, you need to mask it. Use with care.</i>	<code>systemctl mask <my_service>.service</code>	n/a
Unmasking <i>A service that has been masked can only be used again after it has been unmasked.</i>	<code>systemctl unmask <my_service>.service</code>	n/a

8.1.3.2 Enabling/Disabling Services with YaST

Start the YaST module with *YaST > System > System Services (Runlevel)*. In the default view, the *Simple Mode*, it displays an overview listing all the available services and the current status of each service (see Figure 8.1, “System Services (Runlevel)” (page 136)). 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.

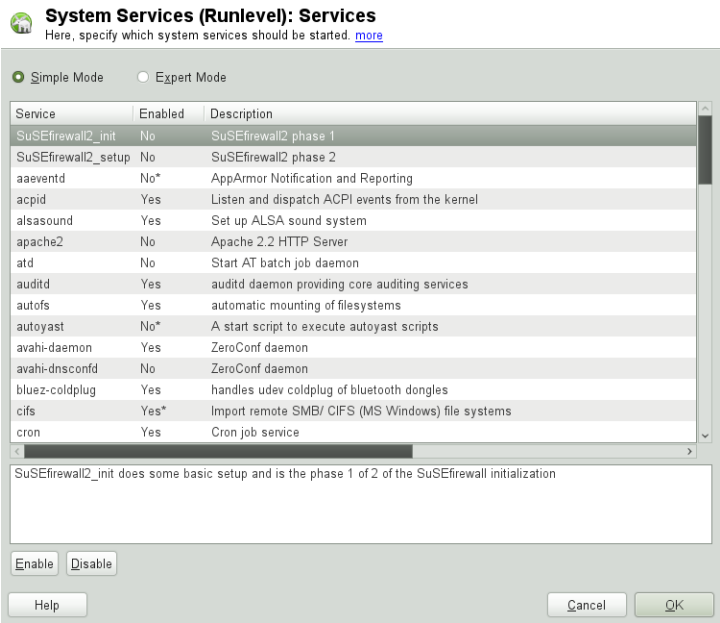
The *Expert Mode* provides detailed control over the runlevels (refer to Section 8.2.1, “Targets vs. Runlevels” (page 137) for more information on runlevels). The runlevel into which the system boots by default is displayed at the top. Normally it defaults to runlevel 5 (full multiuser mode with network and X). A suitable alternative might be runlevel 3 (full multiuser mode with network). Runlevel 4 is undefined to allow creation of a custom runlevel.

To enable or disable individual services use the table in this window. It lists the services available, shows whether they are currently enabled on your system and, if so, for which runlevels. To define the runlevels in which a service should be started, select the respective row with the mouse and activate the checkboxes below that represent the runlevels. A brief description of the currently selected service or daemon is provided below the table overview.

WARNING: Faulty Runlevel Settings May Damage Your System

Faulty runlevel settings may make your system unusable. Before applying your changes, make absolutely sure that you know their consequences.

Figure 8.1: *System Services (Runlevel)*



With *Start/Stop/Refresh*, decide whether a service should be activated. *Refresh status* checks the current status. *Set/Reset* lets you enable or disable a service in the same manner as with the *Simple Mode* interface. Selecting *OK* saves the changed settings to disk.

8.2 System Start and Target Management

The entire process of starting the system and shutting it down is maintained by `systemd`. From this point of view, the Kernel can be considered a background process to maintain all other processes and adjust CPU time and hardware access according to requests from other programs.

8.2.1 Targets vs. Runlevels

With SysV `init` the system was booted into a so called “Runlevel”. A runlevel defines how the system is started and what services are available in the running system. Runlevels are numbered, the most commonly known ones are 0 (shutting down the system), 3 (multiuser with network) and 5 (multiuser with network and displaymanager).

`systemd` introduces a new concept by using so-called “target units”. However, it remains fully compatible to the runlevel concept. Target units are named rather than numbered and serve specific purposes. For example, the targets `local-fs.target` and `swap.target` mount local filesystems and swap spaces.

The target `graphical.target` provides a multiuser system with network and displaymanager capabilities and is equivalent to runlevel 5. Complex targets, such as `graphical.target` act as “meta” targets by combining a subset of other targets. Since `systemd` makes it easy to create custom targets by combining existing targets, it offers great flexibility.

The following list shows a list of the most important `systemd` target units. For a full list refer to `man 7 systemd.special`.

Selected `systemd` Target Units

`default.target`

The target that is booted by default. Not a “real” target, but rather a symlink to another target like `graphic.target`. Can be permanently

changed via YaST (see Section 8.1.3.2, “Enabling/Disabling Services with YaST” (page 135)). To change it for a session, use the Kernel command line option `systemd.unit=<my_target>.target` at the boot prompt.

- `emergency.target`
Starts an emergency shell on the console. Only use it at the boot prompt as `systemd.unit=emergency.target`.
- `graphical.target`
Starts a system with network, multi-user support and a displaymanager.
- `halt.target`
Shuts down the system.
- `mail-transfer-agent.target`
Starts all services necessary for sending and receiving mails.
- `multi-user.target`
Starts a multi-user system with network.
- `reboot.target`
Reboots the system.
- `rescue.target`
Starts a single user system without network.

In order to remain compatible to the SysV init runlevel system, `systemd` provides special targets named `runlevelX.target` mapping the corresponding runlevels numbered `X`.

Table 8.3: *SysV Runlevels and systemd Target Units*

SysV run-level	systemd target	Purpose
0	runlevel0.target, halt.target, poweroff.target	System shutdown
1, S	runlevel1.target, rescue.target,	Single user mode

SysV run-level	systemd target	Purpose
2	runlevel2.target, multi-user.target,	Local multiuser without remote network
3	runlevel3.target, multi-user.target,	Full multiuser with network
4	runlevel4.target	Unused/User Defined
5	runlevel5.target, graphical.target,	Full multiuser with network and displaymanager
6	runlevel6.target, reboot.target,	System reboot

IMPORTANT: systemd ignores /etc/inittab

The runlevels in a SysV init system are configured in `/etc/inittab`. `systemd` *does not* use this configuration. Please refer to Section 8.2.2, “Custom Targets” (page 141) for instructions on how to create your own bootable target.

8.2.1.1 Commands to Change Targets

Use the following commands to operate with target units:

Task	systemd command	SysV init command
Change the current target/runlevel	<code>systemctl isolate <my_target>.target</code>	<code>telinit X</code>
Change to the default target/runlevel	<code>systemctl default</code>	n/a

Task	systemd command	SysV init command
Get the current target/runlevel	<pre>systemctl list-units --type=target</pre> <p>With <code>systemd</code> there is usually more than one active target. The command lists all targets currently active.</p>	<pre>who -r</pre> <p>or</p> <pre>runlevel</pre>
Change the default runlevel persistently	<p>Use the YaST runlevel editor or run the following command:</p> <pre>ln -sf /lib/systemd/system/<my_target>.target /etc/systemd/system/default.target</pre>	<p>Use the YaST runlevel editor or change the line</p> <pre>id:X:initdefault:</pre> <pre>in /etc/inittab</pre>
Change the default runlevel for the current boot process	<p>Enter the following option at the boot prompt</p> <pre>systemd.unit=<my_target.target></pre>	<p>Enter the desired runlevel number at the boot prompt.</p>
Show a target's/runlevel's dependencies	<pre>systemctl show -p "Requires" <my_target.target></pre> <pre>systemctl show -p "Wants" <my_target.target></pre> <p>“Requires” lists the hard dependencies (the ones that must be resolved), whereas “Wants” lists the soft dependencies (the ones that get resolved if possible).</p>	n/a

8.2.2 Custom Targets

On SysV init SUSE systems, runlevel 4 is unused to allow administrators to create their own runlevel configuration. `systemd` allows to create any number of custom targets. It's suggested to start by adapting an existing target such as `graphical.target`.

WARNING: Avoiding Overwritten Customizations

Always do `systemd` customization in `/etc/systemd`, *never* in `/lib/systemd`. Otherwise your changes will be overwritten by the next update of `systemd`.

Procedure 8.1: *Create a Custom Target*

- 1 Copy the configuration file `/lib/systemd/system/graphical.target` to `/etc/systemd/system/<my_target>.target` and adjust it according to your needs.
- 2 The configuration file copied in the previous step already covers the required (“hard”) dependencies for the target. To also cover the wanted (“soft”) dependencies, create a directory `/etc/systemd/system/<my_target>.target.wants`.
- 3 For each wanted service create a symbolic link from `/lib/systemd/system` into `/etc/systemd/system/<my_target>.target.wants`.
- 4 Once you have finished setting up the target, reload the `systemd` configuration to make the new target available:

```
systemctl daemon-reload
```

8.2.3 Debugging System Start-Up

`systemd` offers means to analyze the system start-up process. You can conveniently review the list of all services and their status (rather than having to parse `/var-log/`). `systemd` also allows to scan the start-up procedure to find out how much time each service start-up consumes.

8.2.3.1 Review Start-Up of Services

To review the complete list of services that have been started since booting the system, just enter the command `systemctl`. It lists all active services like shown below (shortened). To get more information on a specific service, use `systemctl status <my_service>.service`.

Example 8.1: *List Active Services*

```
jupiter.example.com: ~ # systemctl
UNIT                                LOAD    ACTIVE SUB    JOB DESCRIPTION
[...]
systemd-random-seed-load.path      loaded active waiting    Random Seed
acpid.service                      loaded active running    ACPI Event
  Daemon
apache2.service                   loaded failed failed      apache
avahi-daemon.service              loaded active running    Avahi mDNS/
DNS-SD Stack
bluetooth.service                 loaded active exited     LSB: handles
udev coldplug of bluetooth dongles
console-kit...-system-start.service loaded active exited     Console
  System Startup Logging
cron.service                      loaded active running    Command
  Scheduler
cups.service                      loaded active running    CUPS
  Printing Service
[...]
LOAD    = Reflects whether the unit definition was properly loaded.
ACTIVE  = The high-level unit activation state, i.e. generalization of SUB.
SUB      = The low-level unit activation state, values depend on unit type.
JOB      = Pending job for the unit.
```

107 units listed. Pass `--all` to see inactive units, too.

To restrict the output to services that failed to start, use the `--failed` option:

Example 8.2: *List Failed Services*

```
jupiter.example.com: ~ # systemctl --failed
UNIT                                LOAD    ACTIVE SUB    JOB DESCRIPTION
apache2.service                   loaded failed failed      apache
NetworkManager.service           loaded failed failed      Network Manager
plymouth-start.service            loaded failed failed      Show Plymouth Boot Screen
[...]

```


8.2.3.2 Debug Start-Up Time

To debug system start-up time, `systemd` offers the `systemd-analyze` command. It shows the total start-up time, a list of services ordered by start-up time and can also generate an SVG graphic showing the time services took to start in relation to the other services.

Listing the System's Start-Up Time

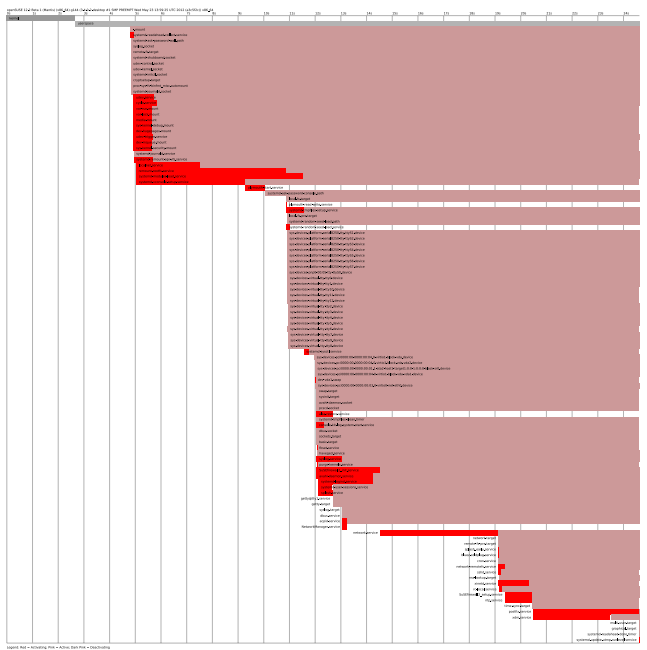
```
jupiter.example.com: ~ # systemd-analyze
Startup finished in 2666ms (kernel) + 21961ms (userspace) = 24628ms
```

Listing the Services Start-Up Time

```
jupiter.example.com: ~ # systemd-analyze blame
6472ms systemd-modules-load.service
5833ms remount-rootfs.service
4597ms network.service
4254ms systemd-vconsole-setup.service
4096ms postfix.service
2998ms xdm.service
2483ms localnet.service
2470ms SuSEfirewall12_init.service
2189ms avahi-daemon.service
2120ms systemd-logind.service
1210ms xinetd.service
1080ms ntp.service
[...]
75ms fbset.service
72ms purge-kernels.service
47ms dev-vda1.swap
38ms bluez-coldplug.service
35ms splash_early.service
```

Services Start-Up Time Graphics

```
jupiter.example.com: ~ # systemd-analyze plot > jupiter.example.com-
startup.svg
```



8.2.3.3 Review the Complete Start-Up Process

Above mentioned commands let you review the services that started and the time it took to start them. If you need to know more details, you can tell `systemd` to ver-
bositely log the complete start-up procedure by entering the following parameters at the
boot prompt:

```
systemd.log_level=debug systemd.log_target=kmsg
```

Now `systemd` writes its log messages into the kernel ring buffer. View that buffer
with `dmesg`:

```
dmesg | less
```

8.3 Advanced Usage

The following sections cover advanced topics for system administrators. For even
more advanced `systemd` documentation, please refer Lennart Pöttering's se-

ries about `systemd` for administrators at <http://0pointer.de/blog/projects>.

8.3.1 System Log

Section 8.1.2, “Debugging Services” (page 132) explains how to view log messages for given service. However, displaying log messages is not restricted to service logs. You can also access and query the complete log written by `systemd`—the so-called “Journal”. Use the command `systemd-journalctl` to display the complete log starting with the oldest entries. Refer to `man 1 systemd-journalctl` for options such as applying filters or changing the output format.

8.3.2 Snapshots

You can save the current state of `systemd` to a named snapshot and later revert to it with the `isolate` subcommand. This is useful when testing services or custom targets, because it allows you to return to a defined state at any time. A snapshot is only available in the current session and will automatically be deleted on reboot. A snapshot name must end in `.snapshot`.

Create a Snapshot

```
systemctl snapshot <my_snapshot>.snapshot
```

Delete a Snapshot

```
systemctl delete <my_snapshot>.snapshot
```

View a Snapshot

```
systemctl show <my_snapshot>.snapshot
```

Activate a Snapshot

```
systemctl isolate <my_snapshot>.snapshot
```

8.3.3 Kernel Control Groups (cgroups)

On a traditional SysV init system it is not always possible to clearly assign a process to the service that spawned it. Some services such as Apache, spawn a lot of 3rd party

processes (e.g. CGI or Java processes), which themselves spawn more processes. This makes a clear assignment difficult or even impossible. Additionally, a service may not terminate correctly, leaving some of its children alive.

`systemd` solves this problem by placing each service into its own cgroup. cgroups are a Kernel feature that allows aggregating processes and all their children into hierarchical organized groups. `systemd` names each cgroup after its service. Since a non-privileged process is not allowed to “leave” its cgroup, this provides an effective way to label all processes spawned by a service with the name of the service.

To list all processes belonging to a service, use the command `systemd-cgls`. The result will look like the following (shortened) example:

Example 8.3: *List all Processes Belonging to a Service*

```
~ # systemd-cgls --no-pager
├─ user
│   └─ root
│       └─ 1
│           ├── 2279 sshd: root@pts/0
│           ├── 2282 -bash
│           └─ 2541 systemd-cgls --no-pager
└─ system
    ├── 1 /sbin/init splash showopts
    ├── apache2.service
    │   ├── 2535 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D SYSTEMD -
    │   k start
    │   ├── 2536 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D SYSTEMD -
    │   k start
    │   ├── 2537 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D SYSTEMD -
    │   k start
    │   ├── 2538 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D SYSTEMD -
    │   k start
    │   ├── 2539 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D SYSTEMD -
    │   k start
    │   └─ 2540 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D SYSTEMD -
    │   k start
    ├── xdm.service
    │   ├── 2250 /usr/bin/xdm
    │   ├── 2253 /usr/bin/X -nolisten tcp -br vt7 -auth /var/lib/xdm/authdir/
    │   authfiles/A:0-ii8Goo
    │   ├── 2263 -:0
    │   └─ 2276 /usr/bin/xconsole -notify -nostdin -verbose -exitOnFail
    ├── ntp.service
    │   └─ 2202 /usr/sbin/ntpd -p /var/run/ntp/ntpd.pid -g -u ntp:ntp -c /etc/
    ntp.conf
    ├── sshd.service
    │   └─ 1743 /usr/sbin/sshd -D
```

See Chapter 10, *Kernel Control Groups* (↑System Analysis and Tuning Guide) for more information about cgroups.

8.3.4 Killing Services (Sending Signals)

As explained in Section 8.3.3, “Kernel Control Groups (cgroups)” (page 145), it is not always possible to assign a process to its parent service process in a SysV init system. This makes it difficult to terminate a service and all of its children. Child processes that have not been killed will remain as zombie processes.

`systemd`'s concept of to confine each service into a cgroup makes it possible to clearly identify all child processes of a service and therefore allows to send a signal to each of these processes. Use `systemctl kill` to send signals to services. For a list of available signals refer to `man 7 signals`.

Sending `SIGTERM` to a Service

`SIGTERM` is the default signal that is sent.

```
systemctl kill <my_service>.service
```

Sending `SIGNAL` to a Service

Use the `-s` option to specify the signal that should be sent.

```
systemctl kill -s SIGNAL <my_service>.service
```

Selecting Processes

By default the `kill` command send the signal to all processes of the specified cgroup. You can restrict it to the `control` or the `main` process. The latter is for example useful to force a service to reload its configuration by sending `SIGHUP`:

```
systemctl kill -s SIGHUP --kill-who=main <my_service>.service
```

8.4 More information

For more information on `systemd` refer to the following online resources:

Homepage

<http://www.freedesktop.org/wiki/Software/systemd>

`systemd` for Administrators

Lennart Pöttering, one of the `systemd` authors, has written a series of blog entries (13 at the time of writing this chapter). Find them at <http://0pointer.de/blog/projects>.

Control Centre: The systemd Linux init system

<http://www.h-online.com/open/features/Control-Centre-The-systemd-Linux-init-system-1565543.html>

Booting up: Tools and tips for systemd, a Linux init tool

<http://www.h-online.com/open/features/Booting-up-Tools-and-tips-for-systemd-1570630.html>

The Boot Loader GRUB

This chapter describes how to configure GRUB (Grand Unified Bootloader), the boot loader used in openSUSE®. A special YaST module is available for configuring all settings. If you are not familiar with the subject of booting in Linux, read the following sections to acquire some background information. This chapter also describes some of the problems frequently encountered when booting with GRUB and their solutions.

This chapter focuses on boot management and the configuration of the boot loader GRUB. The boot procedure as a whole is outlined in Chapter 7, *Booting a Linux System* (page 123). A boot loader represents the interface between the machine (BIOS) and the operating system (openSUSE). The configuration of the boot loader directly impacts the start of the operating system.

The following terms appear frequently in this chapter and might need some explanation:

MBR (Master Boot Record)

The structure of the MBR is defined by an operating system–independent convention. The first 446 bytes are reserved for the program code. They typically hold part of a boot loader program or an operating system selector. The next 64 bytes provide space for a partition table of up to four entries. The partition table contains information about the partitioning of the hard disk and the file system types. The operating system needs this table for handling the hard disk. With conventional generic code in the MBR, exactly one partition must be marked *active*. The last two bytes of the MBR must contain a static “magic number”

(AA55). An MBR containing a different value is regarded as invalid by some BIOSes, so is not considered for booting.

Boot Sectors

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

9.1 Booting with GRUB

GRUB comprises two stages. Stage 1 consists of 512 bytes and its only task is to load the second stage of the boot loader. Subsequently, stage 2 is loaded. This stage contains the main part of the boot loader.

In some configurations, an intermediate stage 1.5 can be used, which locates and loads stage 2 from an appropriate file system. If possible, this method is chosen by default on installation or when initially setting up GRUB with YaST.

Stage 2 is able to access many file systems. Currently, ext2, ext3, ReiserFS, Minix, and the DOS FAT file system used by Windows are supported. To a certain extent, XFS, and UFS and FFS used by BSD systems are also supported. Since version 0.95 GRUB is also able to boot from a CD or DVD containing an ISO 9660 standard file system pursuant to the “El Torito” specification. Even before the system is booted, GRUB can access file systems of supported BIOS disk devices (floppy disks or hard disks, CD drives and DVD drives detected by the BIOS). Therefore, changes to the GRUB configuration file (`menu.lst`) do not require a new installation of the boot manager. When the system is booted, GRUB reloads the menu file with the valid paths and partition data of the kernel or the initial RAM disk (`initrd`) and locates these files.

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

`/boot/grub/menu.lst`

This file contains all information about partitions or operating systems that can be booted with GRUB. Without this information, the GRUB command line prompts the user for how to proceed. For details, see Section 9.1.1.3, “Editing Menu Entries during the Boot Procedure” (page 155).

`/boot/grub/device.map`

This file translates device names from the GRUB and BIOS notation to Linux device names.

`/etc/grub.conf`

This file contains the commands, parameters and options the GRUB shell needs for installing the boot loader correctly.

`/etc/sysconfig/bootloader`

This file is read by the perl-bootloader library which is used when configuring the bootloader with YaST and every time a new kernel is installed. It contains configuration options (such as kernel parameters) that will be added by default to the bootloader configuration file.

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

In GRUB, all boot parameters can be changed prior to booting. For example, errors made when editing the menu file can be corrected in this way. Boot commands can also be entered interactively at a kind of input prompt. For details, see Section 9.1.1.3, “Editing Menu Entries during the Boot Procedure” (page 155). GRUB offers the possibility of determining the location of the kernel and the `initrd` prior to booting. In this way, you can even boot an installed operating system for which no entry exists in the boot loader configuration.

GRUB actually exists in two versions: as a boot loader and as a normal Linux program in `/usr/sbin/grub`. The latter is referred to as the *GRUB shell*. It provides an emulation of GRUB in the installed system and can be used to install GRUB or test new settings before applying them. The functionality to install GRUB as the boot loader on a hard disk or floppy disk is integrated in GRUB in the form of the command `setup`. This is available in the GRUB shell when Linux is loaded.

9.1.1 The File `/boot/grub/menu.lst`

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

Every time the system is booted, GRUB loads the menu file from the file system. For this reason, GRUB does not need to be reinstalled after every change to the file. Use the YaST boot loader to modify the GRUB configuration as described in Section 9.2, “Configuring the Boot Loader with YaST” (page 160).

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

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

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

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

The device names in GRUB are explained in Section 9.1.1.1, “Naming Conventions for Hard Disks and Partitions” (page 153). This example specifies the first block of the fourth partition of the first hard disk.

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

If the kernel does not have built-in drivers for access to the root partition or a recent Linux system with advanced hotplug features is used, `initrd` must be specified with a separate GRUB command whose only argument is the path to the `initrd` file. Because the loading address of the `initrd` is written into the loaded kernel image, the command `initrd` must follow after the `kernel` command.

The command `root` simplifies the specification of kernel and `initrd` files. The only argument of `root` is a device or a partition. This device is used for all kernel,

`initrd`, or other file paths for which no device is explicitly specified until the next `root` command.

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

After writing all menu entries, define one of them as the `default` entry. Otherwise, the first one (entry 0) is used. You can also specify a time-out in seconds after which the default entry should boot. `timeout` and `default` usually precede the menu entries. An example file is described in Section 9.1.1.2, “An Example Menu File” (page 154).

9.1.1.1 Naming Conventions for Hard Disks and Partitions

The naming convention GRUB uses for hard disks and partitions differ from that used for normal Linux devices. It more closely resembles the simple disk enumeration the BIOS does and the syntax is similar to that used in some BSD derivatives. In GRUB, the numbering of the partitions start with zero. This means that `(hd0, 0)` is the first partition of the first hard disk. On a common desktop machine with a hard disk connected as primary master, the corresponding Linux device name is `/dev/sda1`.

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

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

Being dependent on BIOS devices, GRUB does not distinguish between PATA (IDE), SATA, SCSI, and hardware RAID devices. All hard disks recognized by the BIOS or other controllers are numbered according to the boot sequence preset in the BIOS.

Unfortunately, it is often not possible to map the Linux device names to BIOS device names exactly. It generates this mapping with the help of an algorithm and saves it to

the file `device.map`, which can be edited if necessary. Information about the file `device.map` is available in Section 9.1.2, “The File `device.map`” (page 156).

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 PATA (IDE) hard disk containing Linux in its first partition:

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

9.1.1.2 An Example Menu File

The following example shows the structure of a GRUB menu file. The example installation has a Linux boot partition under `/dev/sda5`, a root partition under `/dev/sda7` and a Windows installation under `/dev/sda1`.

```
gfxmenu (hd0,4)/boot/message❶
color white/blue black/light-gray❷
default 0❸
timeout 8❹

title linux❺
    root (hd0,4)
    kernel /boot/vmlinuz root=/dev/sda7 vga=791 resume=/dev/sda9
    initrd /boot/initrd

title windows❻
    rootnoverify (hd0,0)
    chainloader +1

title floppy❼
    rootnoverify (hd0,0)
    chainloader (fd0)+1

title failsafe❽
    root (hd0,4)
    kernel /boot/vmlinuz.shipped root=/dev/sda7 ide=nodma \
    apm=off acpi=off vga=normal nosmp maxcpus=0 3 noresume
    initrd /boot/initrd.shipped
```

The first block defines the configuration of the splash screen:

- ❶ The background image `message` is located in the `/boot` directory of the `/dev/sda5` partition.
- ❷ 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`.

- ❸ The first (0) menu entry `title linux` is booted by default.
- ❹ After eight seconds without any user input, GRUB automatically boots the default entry. To deactivate automatic boot, delete the `timeout` line. If you set `timeout 0`, GRUB boots the default entry immediately.

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

- ❺ The first entry (`title linux`) is responsible for booting openSUSE. The kernel (`vmlinuz`) is located in the first logical partition (the boot partition) of the first hard disk. Kernel parameters, such as the root partition and VGA mode, are appended here. The root partition is specified according to the Linux naming convention (`/dev/sda7/`) because this information is read by the kernel and has nothing to do with GRUB. The `initrd` is also located in the first logical partition of the first hard disk.
- ❻ 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.
- ❼ The next entry enables booting from floppy disk without modifying the BIOS settings.
- ❽ The boot option `failsafe` starts Linux with a selection of kernel parameters that enables Linux to boot even on problematic systems.

The menu file can be changed whenever necessary. GRUB then uses the modified settings during the next boot. Edit the file permanently using YaST or an editor of your choice. Alternatively, make temporary changes interactively using the `edit` function of GRUB. See Section 9.1.1.3, “Editing Menu Entries during the Boot Procedure” (page 155).

9.1.1.3 Editing Menu Entries during the Boot Procedure

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

IMPORTANT: Keyboard Layout during the Boot Procedure

The US keyboard layout is the only one available when booting. See Figure “US Keyboard Layout” (↑Start-Up).

Editing menu entries facilitates the repair of a defective system that can no longer be booted, because the faulty configuration file of the boot loader can be circumvented by manually entering parameters. Manually entering parameters during the boot procedure is also useful for testing new settings without impairing the native system.

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

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

```
title linux
    root (hd0,0)
    kernel /vmlinuz root=/dev/sda3 additional parameter
    initrd /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.

9.1.2 The File `device.map`

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

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

or

```
(fd0)  /dev/fd0  
(hd0)  /dev/disk-by-id/DISK1 ID  
(hd1)  /dev/disk-by-id/DISK2 ID
```

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

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

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

9.1.3 The File `/etc/grub.conf`

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

```
setup --stage2=/boot/grub/stage2 --force-lba (hd0,1) (hd0,1)  
quit
```

This command tells GRUB to automatically install the boot loader to the second partition on the first hard disk (`hd0,1`) using the boot images located on the same partition. The `--stage2=/boot/grub/stage2` parameter is needed to install the `stage2` image from a mounted file system. Some BIOSes have a faulty LBA support implementation, `--force-lba` provides a solution to ignore them.

9.1.4 The File `/etc/sysconfig/bootloader`

This configuration file is only used when configuring the bootloader with YaST and every time a new kernel is installed. It is evaluated by the `perl-bootloader` library which modifies the bootloader configuration file (for example `/boot/grub/menu.lst` for GRUB) accordingly. `/etc/sysconfig/bootloader` is not a GRUB specific configuration file - the values are applied to any bootloader installed on openSUSE.

NOTE: Bootloader Configuration after a Kernel Update

Every time a new kernel is installed, the perl bootloader writes a new bootloader configuration file (for example `/boot/grub/menu.lst` for GRUB) using the defaults specified in `/etc/sysconfig/bootloader`. If you are using a customized set of kernel parameters, make sure to adjust the relevant defaults in `/etc/sysconfig/bootloader` according to your needs.

LOADER_TYPE

Specifies the bootloader installed on the system (e.g. GRUB or LILO). Do not modify—use YaST to change the bootloader as described in Procedure 9.6, “Changing the Boot Loader Type” (page 165).

DEFAULT_VGA / FAILSAFE_VGA / XEN_VGA

Screen resolution and color depth of the framebuffer used during booting are configured with the kernel parameter `vga`. These values define which resolution and color depth to use for the default boot entry, the failsafe and the XEN entry. The following values are valid:

Table 9.1: *Screen Resolution and Color Depth Reference*

	640x480	800x600	1024x768	1280x1024	1600x1200
8bit	0x301	0x303	0x305	0x307	0x31C
15bit	0x310	0x313	0x316	0x319	0x31D
16bit	0x311	0x314	0x317	0x31A	0x31E
24bit	0x312	0x315	0x318	0x31B	0x31F

DEFAULT_APPEND / FAILSAFE_APPEND / XEN_KERNEL_APPEND

Kernel parameters (other than `vga`) that are automatically appended to the default, failsafe and XEN boot entries in the bootloader configuration file.

CYCLE_DETECTION / CYCLE_NEXT_ENTRY

Configure whether to use boot cycle detection and if so, which alternative entry from `/boot/grub/menu.lst` to boot in case of a reboot cycle (e.g. Failsafe). See `/usr/share/doc/packages/bootcycle/README` for detailed information.

9.1.5 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, set a boot password.

IMPORTANT: Boot Password and Splash Screen

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

As the user `root`, proceed as follows to set a boot password:

- 1 At the root prompt, encrypt the password using `grub-md5-crypt`:

```
# grub-md5-crypt
Password: ****
Retype password: ****
Encrypted: $1$lS2dv/$JOYcdxIn7CJk9xShzzJVw/
```

- 2 Paste the encrypted string into the global section of the file `menu.lst`:

```
gfxmenu (hd0,4)/message
color white/blue black/light-gray
default 0
timeout 8
password --md5 $1$lS2dv/$JOYcdxIn7CJk9xShzzJVw/
```

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

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

```
title linux
    kernel (hd0,4)/vmlinuz root=/dev/sda7 vga=791
    initrd (hd0,4)/initrd
    lock
```

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

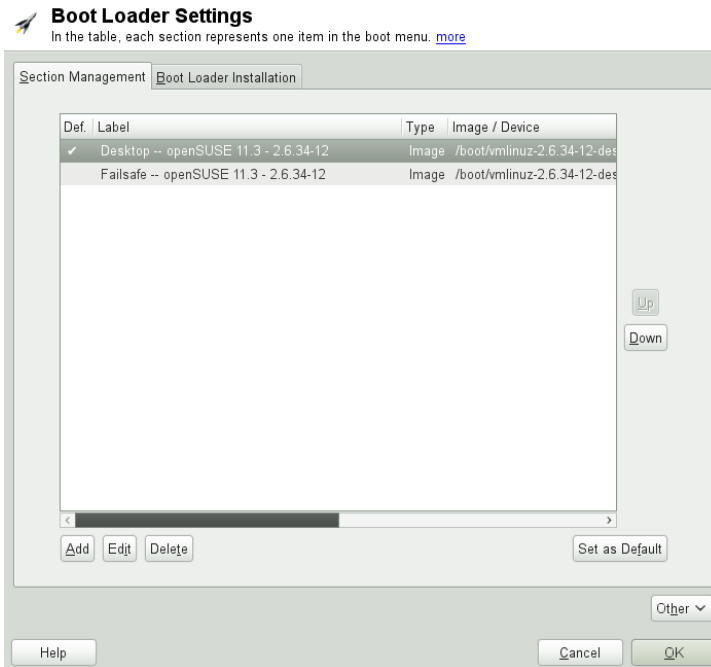
```
Error 32: Must be authenticated
```

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

9.2 Configuring the Boot Loader with YaST

The easiest way to configure the boot loader in your openSUSE system is to use the YaST module. In the YaST Control Center, select *System > Boot Loader*. As in Figure 9.1, “Boot Loader Settings” (page 160), this shows the current boot loader configuration of your system and allows you to make changes.

Figure 9.1: *Boot Loader Settings*



Use the *Section Management* tab to edit, change and delete boot loader sections for the individual operating systems. To add an option, click *Add*. To change the value of an

existing option, select it with the mouse and click *Edit*. To remove an existing entry, select it and click *Delete*. If you are not familiar with boot loader options, read Section 9.1, “Bootting with GRUB” (page 150) first.

Use the *Boot Loader Installation* tab to view and change settings related to type, location and advanced loader settings.

Click *Other* to access advanced configuration options. The build-in editor lets you change the GRUB configuration files. For details, see Section 9.1, “Bootting with GRUB” (page 150). You can also delete the existing configuration and *Start from Scratch* or let YaST *Propose a New Configuration*. It is also possible to write the configuration to disk or reread the configuration from the disk. To restore the original Master Boot Record (MBR) that was saved during the installation, choose *Restore MBR of Hard Disk*.

9.2.1 Adjusting the Default Boot Entry

To change the system that is booted by default, proceed as follows:

Procedure 9.1: *Setting the Default System*

- 1 Open the *Section Management* tab.
- 2 Select the desired entry from the list.
- 3 Click *Set as Default*.
- 4 Click *OK* to activate these changes.

9.2.2 Modifying the Boot Loader Location

To modify the location of the boot loader, follow these steps:

Procedure 9.2: *Changing the Boot Loader Location*

- 1 Select the *Boot Loader Installation* tab and then choose one of the following options for *Boot Loader Location*:

Boot from Master Boot Record

This installs the boot loader in the MBR of the first disk (according to the boot sequence preset in the BIOS).

Boot from Root Partition

This installs the boot loader in the boot sector of the `/` partition (this is the default).

Boot from Boot Partition

This installs the boot loader in the boot sector of the `/boot` partition.

Boot from Extended Partition

This installs the boot loader in the extended partition container.

Custom Boot Partition

Use this option to specify the location of the boot loader manually.

- 2 Click *OK* to apply your changes.

9.2.3 Changing the Boot Loader Time-Out

The boot loader does not boot the default system immediately. During the time-out, you can select the system to boot or write some kernel parameters. To set the boot loader time-out, proceed as follows:

Procedure 9.3: *Changing the Boot Loader Time-Out*

- 1 Open the *Boot Loader Installation* tab.
- 2 Click *Boot Loader Options*.
- 3 Change the value of *Time-Out in Seconds* by typing in a new value and clicking the appropriate arrow key with your mouse, or by using the arrow keys on the keyboard.
- 4 Click *OK* twice to save the changes.

WARNING: Timeout of 0 Seconds

When setting the timeout to 0 seconds, you will not be able to access GRUB during boot time. When having set the default boot option to a non-Linux operation system at the same time, this effectively disables access to the Linux system.

9.2.4 Setting a Boot Password

Using this YaST module, you can also set a password to protect booting. This gives you an additional level of security.

Procedure 9.4: *Setting a Boot Loader Password*

- 1 Open the *Boot Loader Installation* tab.
- 2 Click *Boot Loader Options*.
- 3 Activate the *Protect Boot Loader with Password* option with a click and type in your *Password* twice.
- 4 Click *OK* twice to save the changes.

9.2.5 Adjusting the Disk Order

If your computer has more than one hard disk, you can specify the boot sequence of the disks to match the BIOS setup of the machine (see Section 9.1.2, “The File device.map” (page 156)). To do so, proceed as follows:

Procedure 9.5: *Setting the Disk Order*

- 1 Open the *Boot Loader Installation* tab.
- 2 Click *Boot Loader Installation Details*.
- 3 If more than one disk is listed, select a disk and click *Up* or *Down* to reorder the displayed disks.
- 4 Click *OK* two times to save the changes.

9.2.6 Configuring Advanced Options

Advanced boot options can be configured via *Boot Loader Installation > Boot Loader Options*. Normally, it should not be necessary to change the default settings.

Set Active Flag in Partition Table for Boot Partition

Activates the partition that contains the boot loader. Some legacy operating systems (such as Windows 98) can only boot from an active partition.

Write Generic Boot Code to MBR

Replaces the current MBR with generic, operating system independent code.

Debugging Flag

Sets GRUB in debug mode where it displays messages to show disk activity.

Hide Boot Menu

Hides the boot menu and boots the default entry.

WARNING

When hiding the boot menu, you will not be able to access GRUB during boot time. When having set the default boot option to a non-Linux operation system at the same time, this effectively disables access to the Linux system.

Use Trusted GRUB

Starts the Trusted GRUB which supports trusted computing functionality.

Enable Acoustic Signals

Enables or disables acoustic signals in GRUB.

Graphical Menu File

Path to the graphics file used when displaying the boot screen.

Use Serial Console

If your machine is controlled via a serial console, activate this option and specify which COM port to use at which speed. See `info grub` or <http://www.gnu.org/software/grub/manual/grub.html#Serial-terminal>

9.2.7 Changing Boot Loader Type

Set the boot loader type in *Boot Loader Installation*. The default boot loader in openSUSE is GRUB. To use LILO or ELILO, proceed as follows:

WARNING: LILO is unsupported

Using LILO is not recommended—it is unsupported on openSUSE. Only use it in special cases.

Procedure 9.6: *Changing the Boot Loader Type*

- 1 Select the *Boot Loader Installation* tab.
- 2 For *Boot Loader*, select *LILO*.
- 3 In the dialog box that opens, select one of the following actions:

Propose New Configuration

Have YaST propose a new configuration.

Convert Current Configuration

Have YaST convert the current configuration. When converting the configuration, some settings may be lost.

Start New Configuration from Scratch

Write a custom configuration. This action is not available during the installation of openSUSE.

Read Configuration Saved on Disk

Load your own `/etc/lilo.conf`. This action is not available during the installation of openSUSE.

- 4 Click *OK* two times to save the changes.

During the conversion, the old GRUB configuration is saved to the disk. To use it, simply change the boot loader type back to GRUB and choose *Restore Configuration Saved before Conversion*. This action is available only on an installed system.

NOTE: Custom Boot Loader

To use a boot loader other than GRUB or LILO, select *Do Not Install Any Boot Loader*. Read the documentation of your boot loader carefully before choosing this option.

9.3 Uninstalling the Linux Boot Loader

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

To uninstall GRUB, start YaST and click *System > Boot Loader* to start the boot loader module. Select *Other > Restore MBR of Hard Disk* and confirm with *Yes, Rewrite*.

9.4 Creating Boot CDs

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

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

Procedure 9.7: *Creating Boot CDs*

- 1 Change into a directory in which to create the ISO image, for example: `cd /tmp`
- 2 Create a subdirectory for GRUB and change into the newly created `iso` directory:

```
mkdir -p iso/boot/grub && cd iso
```
- 3 Copy the kernel, the files `stage2_eltorito`, `initrd`, `menu.lst` and `message` to `iso/boot/`:

```
cp /boot/vmlinuz boot/  
cp /boot/initrd boot/  
cp /boot/message boot/  
cp /usr/lib/grub/stage2_eltorito boot/grub  
cp /boot/grub/menu.lst boot/grub
```
- 4 Replace the `root (hdx, y)` entries with `root (cd)` to point to the CD-ROM device. You may also need to adjust the paths to the message file, the kernel and the `initrd`—they need to point to `/boot/message`, `/boot/vmlinuz` and `/boot/initrd`, respectively. After having made the adjustments, `menu.lst` should look similar to the following example:

```
timeout 8  
default 0
```



```
gfxmenu (cd) /boot/message

title Linux
    root (cd)
    kernel /boot/vmlinuz root=/dev/sda5 vga=794 resume=/dev/sda1 \
    splash=verbose showopts
    initrd /boot/initrd
```

Use `splash=silent` instead of `splash=verbose` to prevent the boot messages from appearing during the boot procedure.

5 Create the ISO image with the following command:

```
genisoimage -R -b boot/grub/stage2_eltorito -no-emul-boot \
-bboot-load-size 4 -boot-info-table -iso-level 2 -input-charset utf-8 \
-o grub.iso /tmp/iso
```

6 Write the resulting file `grub.iso` to a CD using your preferred utility. Do not burn the ISO image as a data file, but use the option for burning a CD image in your burning utility.

9.5 The Graphical SUSE Screen

The graphical SUSE screen is displayed on the first console if the option `vga=value` is used as a kernel parameter. If you install using YaST, this option is automatically activated in accordance with the selected resolution and the graphics card. There are three ways to disable the SUSE screen, if desired:

Disabling the SUSE Screen When Necessary

Enter the command `echo 0 >/proc/splash` on the command line to disable the graphical screen. To activate it again, enter `echo 1 >/proc/splash`.

Disabling the SUSE screen by default

Add the kernel parameter `splash=0` to your boot loader configuration. Chapter 9, *The Boot Loader GRUB* (page 149) provides more information about this. However, if you prefer the text mode (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*. Disabling framebuffer support in the kernel automatically disables the splash screen, as well.

WARNING: No Support

SUSE cannot provide any support for your system if you run it with a custom kernel.

9.6 Troubleshooting

This section lists some of the problems frequently encountered when booting with GRUB and a short description of possible solutions. Some of the problems are covered in articles in the Support Database at http://en.opensuse.org/Portal:Support_database. Use the search dialog to search for keywords like *GRUB*, *boot* and *boot loader*.

GRUB and XFS

XFS leaves no room for `stage1` in the partition boot block. Therefore, do not specify an XFS partition as the location of the boot loader. This problem can be solved by creating a separate boot partition that is not formatted with XFS.

GRUB Reports GRUB Geom Error

GRUB checks the geometry of connected hard disks when the system is booted. Sometimes, the BIOS returns inconsistent information and GRUB reports a GRUB Geom Error. In this case, update the BIOS.

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 Several Hard Disks Does Not Boot

During the installation, YaST may have incorrectly determined the boot sequence of the hard disks. For example, GRUB may regard the PATA (IDE) disk as `hd0` and the SCSI disk as `hd1`, although the boot sequence in the BIOS is reversed (SCSI *before* PATA).

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

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

Booting Windows from the Second Hard Disk

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

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

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

9.7 For More Information

Extensive information about GRUB is available at <http://www.gnu.org/software/grub/>. Also refer to the `grub` info page. You can also search for the keyword “GRUB” in the Support Database at http://en.opensuse.org/Portal:Support_database to get information about special issues.

The Boot Loader GRUB2

This chapter describes how to configure GRUB2 (Grand Unified Bootloader), the boot loader used in openSUSE®. It is a successor of the traditional GRUB boot loader — now called “GRUB Legacy” — which is described in Chapter 9, *The Boot Loader GRUB* (page 149). GRUB2 has become the default boot loader in openSUSE® since version 12.2. A special YaST module is available for configuring major GRUB2 settings. If you are not familiar with the subject of booting in Linux, read the initial sections of Chapter 9, *The Boot Loader GRUB* (page 149) to acquire some background information. The boot procedure as a whole is outlined in Chapter 7, *Booting a Linux System* (page 123).

10.1 Main Differences from GRUB Legacy

- The configuration is stored in different files. The configuration syntax has changed.
- Partition numbers start at 1 (instead of 0 as in GRUB Legacy).
- More file-systems are supported.
- GRUB2 can directly read files stored on LVM or RAID devices.
- The user interface can be translated, including menu entry names.
- GRUB2 includes mechanism for loading modules to support specific features, such as filesystems etc.

- So called “Stages” were dropped and the images that make up GRUB2 were re-organized.

10.2 Configuration File Structure

The actual configuration of GRUB2 is based on the following files:

`/boot/grub2/grub.cfg`

This file contains all information about the GRUB2 menu items. It replaces `menu.lst` used in GRUB Legacy. `grub.cfg` is built by the `grub2-mkconfig` command, and normally is not edited manually.

`/etc/default/grub`

This file controls the user settings of GRUB2 and usually includes additional environmental settings such as backgrounds and themes.

Scripts under `/etc/grub.d/`

The scripts in this directory are read during execution of the `grub2-mkconfig` command. Their instructions are then incorporated into the main configuration file `/boot/grub/grub.cfg`.

`/etc/sysconfig/bootloader`

This file is read by the `perl-bootloader` library which is used when configuring the bootloader with YaST and every time a new kernel is installed. It contains configuration options (such as kernel parameters) that will be added by default to the bootloader configuration file.

GRUB2 can be controlled in various ways. Boot entries from an existing configuration can be selected from the graphical menu (splash screen). The configuration is loaded from the file `grub.cfg` which is compiled from other configuration files (see below). All GRUB2 configuration files are considered system files, and you need root privileges to edit them. Remember to run `grub2-mkconfig -o /boot/grub2/grub.cfg` after any change to GRUB2 configuration files.

10.2.1 The File `/boot/grub2/grub.cfg`

The graphical splash screen with the boot menu is based on the GRUB2 configuration file `/boot/grub2/grub.cfg`, which contains all information about all partitions or operating systems that can be booted by the menu.

Every time the system is booted, GRUB2 loads the menu file from the file system. For this reason, GRUB2 does not need to be re-installed after every change to the file. `grub.cfg` is automatically rebuilt with kernel installations or removals.

`grub.cfg` is compiled from the file `/etc/default/grub` and scripts found in the `/etc/grub.d/` directory with the `grub2-mkconfig -o /boot/grub2/grub.cfg` command. Therefore you should never edit the file manually. Instead, edit the related source files under `/etc/grub.d/` or use the YaST boot loader to modify the GRUB2 configuration as described in Section 10.3, “Configuring the Boot Loader with YaST” (page 180).

10.2.2 The File `/etc/default/grub`

More general options of GRUB2 belong here, such as the time the menu is displayed, or the default OS to boot. To list all available options, see the output of the following command:

```
grep "export GRUB_DEFAULT" -A50 /usr/sbin/grub2-mkconfig | grep GRUB_
```

In addition to already defined variables, the user may introduce their own variables, and use them later in the scripts found in the `/etc/grub.d` directory.

After you edit `/etc/default/grub`, run `grub2-mkconfig -o /boot/grub2/grub.cfg` to update the main configuration file.

10.2.2.1 General Options

This section details some of the options commonly used in the `/etc/default/grub` file. For their complete list, see GNU GRUB manual [<http://www.gnu.org/software/grub/manual/grub.html#Simple-configuration>].

GRUB_DEFAULT

Sets the default menu entry that will be booted next time the computer is rebooted. It can be a numeric value, a complete menu entry quotation, or “saved”. A few examples follow:

`GRUB_DEFAULT=2` boots the third (counted from zero) boot menu entry.

`GRUB_DEFAULT="2>0"` boots the first submenu entry of the third top level menu entry.

`GRUB_DEFAULT="Example boot menu entry"` boots the menu entry whose title matches the quotation.

`GRUB_DEFAULT=saved` boots the entry specified by the `grub2-reboot` or `grub2-set-default` commands. While `grub2-reboot` sets the default boot entry for the next reboot only, `grub2-set-default` sets the default boot entry until changed.

`GRUB_SAVEDEFAULT`

If set to `true`, it will automatically choose the last selected OS from the boot menu as the default boot entry on the next boot. For this to work, you also need to specify `GRUB_DEFAULT=saved`.

`GRUB_HIDDEN_TIMEOUT`

Waits the specified number of seconds for the user to press a key. During the period no menu is shown unless the user presses a key. If no key is pressed during the time specified, the control is passed to `GRUB_TIMEOUT`. `GRUB_HIDDEN_TIMEOUT=0` first checks whether `Shift` is pressed and shows the boot menu if yes, otherwise immediately boots the default menu entry. This is the default when only one bootable OS is identified by `GRUB2`.

`GRUB_HIDDEN_TIMEOUT_QUIET`

If `false` is specified, a countdown timer is displayed on a blank screen when the `GRUB_HIDDEN_TIMEOUT` feature is active.

`GRUB_TIMEOUT`

Time period in seconds the boot menu is displayed before automatically booting the default boot entry. If you press a key, the timeout is cancelled and `GRUB2` waits for you to make the selection manually. `GRUB_TIMEOUT=-1` will cause the menu to be displayed until you select the boot entry manually.

`GRUB_CMDLINE_LINUX`

Entries on this line are added to the end of the booting command line for both normal and recovery modes. It is used to pass options to the kernel.

`GRUB_CMDLINE_LINUX_DEFAULT`

Same as `GRUB_CMDLINE_LINUX` but the entries are passed and appended in the normal mode only.

GRUB_TERMINAL

Enables and specifies input/output terminal device. Can be `console` (PC BIOS and EFI consoles), `serial` (serial terminal), `ofconsole` (Open Firmware console), or the default `gfxterm` (graphics-mode output).

GRUB_GFXMODE

The resolution used for the `gfxterm` graphical terminal. Note that you can only use modes supported by your graphics card (VBE). The default is 'auto', which tries to select a preferred resolution. You can display the screen resolutions available to GRUB2 by typing `vbeinfo` in the GRUB2 command line. The command line is accessed by typing `c` when the GRUB2 boot menu screen is displayed.

You can also specify a color bit depth by appending it to the resolution setting, for example `GRUB_GFXMODE=1280x1024x24`.

TIP

Setting the same resolution in GRUB2 and the operating system will slightly reduce the boot time.

GRUB_BACKGROUND

Set a background image for the `gfxterm` graphical terminal. The image must be a file readable by GRUB2 at boot time, and it must end with the `.png`, `.tga`, `.jpg`, or `.jpeg` suffix. If necessary, the image will be scaled to fit the screen.

10.2.3 Scripts in /etc/grub.d

The scripts in this directory are read during execution of the `grub2-mkconfig` command, and their instructions are incorporated into `/boot/grub2/grub.cfg`. The order of menu items in `grub.cfg` is determined by the order in which the files in this directory are run. Files with a leading numeral are executed first, beginning with the lowest number. `00_header` is run before `10_linux`, which would run before `40_custom`. If files with alphabetic names are present, they are executed after the numerically-named files. Only executable files generate output to `grub.cfg` during execution of `grub2-mkconfig`. By default all files in the `/etc/grub.d` directory are executable.

A list of default scripts follows.

00_header

Sets environmental variables such as system file locations, video settings, themes, and previously saved entries. It also imports preferences stored in the `/etc/default/grub`. Normally you do not need to make changes to this file.

10_linux

Identifies Linux kernels on the root device and creates relevant menu entries.

This includes the associated recovery mode option if enabled. Only the latest kernel is displayed on the main menu page, with additional kernels included in a submenu.

30_os-prober

This script uses OS-prober to search for Linux and other operating systems and places the results in the GRUB2 menu. There are sections to identify specific operating systems, such as Linux, Windows, Hurd, or Mac OS X.

40_custom

A template for custom menu entries to be inserted into `grub.cfg`. The contents of this file, below the `exec tail -n +3 $0` line and the default comments, are imported directly into `grub.cfg` without any changes.

90_persistent

This is a special script which copies a corresponding part of the `grub.cfg` file and outputs it back unchanged. This way you can modify that part of `grub.cfg` directly and the change survives the execution of `grub2-mkconfig`.

10.2.4 The File `/etc/sysconfig/bootloader`

This configuration file is only used when configuring the bootloader with YaST and every time a new kernel is installed. It is evaluated by the perl-bootloader library which modifies the bootloader configuration file (for example `/boot/grub2/grub.cfg` for GRUB2) accordingly. `/etc/sysconfig/bootloader` is not a GRUB2 specific configuration file - the values are applied to any bootloader installed on openSUSE.

NOTE: Bootloader Configuration after a Kernel Update

Every time a new kernel is installed, the perl bootloader writes a new bootloader configuration file using the defaults specified in `/etc/syscon-`

`fig/bootloader`. If you are using a customized set of kernel parameters, make sure to adjust the relevant defaults in `/etc/sysconfig/bootloader` according to your needs.

LOADER_TYPE

Specifies the bootloader installed on the system (e.g. GRUB2, GRUB, or LILO). Do not modify—use YaST to change the bootloader as described in Section 10.3.5, “Changing Boot Loader Type” (page 184).

LOADER_LOCATION

Specifies the location of the bootloader. Do not modify—use YaST to change the bootloader as described in Section 10.3.2, “Modifying the Boot Loader Location” (page 181).

DEFAULT_VGA / FAILSAFE_VGA / XEN_VGA

Screen resolution and color depth of the framebuffer used during booting are configured with the kernel parameter `vga`. These values define which resolution and color depth to use for the default boot entry, the failsafe and the XEN entry. The following values are valid:

Table 10.1: *Screen Resolution and Color Depth Reference*

	640x480	800x600	1024x768	1280x1024	1600x1200
8bit	0x301	0x303	0x305	0x307	0x31C
15bit	0x310	0x313	0x316	0x319	0x31D
16bit	0x311	0x314	0x317	0x31A	0x31E
24bit	0x312	0x315	0x318	0x31B	0x31F

DEFAULT_APPEND / FAILSAFE_APPEND / XEN_KERNEL_APPEND

Kernel parameters (other than `vga`) that are automatically appended to the default, failsafe and XEN boot entries in the bootloader configuration file.

CYCLE_DETECTION / CYCLE_NEXT_ENTRY

Configure whether to use boot cycle detection and if so, which alternative boot entry from to boot in case of a reboot cycle (e.g. Failsafe). See `/usr/share/doc/packages/bootcycle/README` for detailed information.

10.2.5 Mapping between BIOS drives and Linux devices

In GRUB Legacy, the `device.map` configuration file was used to derive Linux device names from BIOS drive numbers. The map between BIOS drives and Linux devices cannot always be guessed correctly: for example, GRUB will get the order wrong if you exchange the boot sequence between IDE and SCSI in your BIOS.

GRUB2 avoids this problem by using device ID strings (UUIDs) or file system labels when generating `grub.cfg`. GRUB2 utilities create a temporary device map on the fly, which is usually sufficient, particularly in the case of single-disk systems.

However, if you need to override the GRUB2's automatic device mapping mechanism, create your custom mapping file `/boot/grub2/device.map` similarly as in GRUB Legacy (see Section 9.1.2, “The File `device.map`” (page 156) for more details). Please note that GRUB2 partition number start at 1 and not at 0 as in GRUB Legacy.

10.2.6 Editing Menu Entries during the Boot Procedure

Editing menu entries facilitates the repair of a defective system that can no longer be booted, because the faulty configuration file of the boot loader can be circumvented by manually entering parameters. Manually entering parameters during the boot procedure is also useful for testing new settings without impairing the native system.

In the graphical boot menu, select the operating system to boot with the arrow keys. If you select a Linux system, you can enter additional boot parameters at the boot prompt. To edit individual menu entries directly, press `Esc` to exit the splash screen and get to the GRUB2 text-based menu then press `E`. Changes made in this way only apply to the current boot and are not adopted permanently.

NOTE: Password-protected Bootloader

If you protected your bootloader with a password as described in Section 10.2.7, “Setting a Boot Password” (page 179), you need to first enter the specified username and password to “unlock” the bootloader.

IMPORTANT: Keyboard Layout during the Boot Procedure

The US keyboard layout is the only one available when booting. See Figure “US Keyboard Layout” (↑Start-Up).

Configuration options and commands related to the selected boot entry appears in a simple text editor interface. Use arrows to move in the text and modify the menu entry as desired. The TAB key suggests possible completions, while ESC discards all changes and returns to the GRUB2 menu. Once you finish editing the menu-entry, press F10 to boot it.

10.2.7 Setting a Boot Password

Even before the operating system is booted, GRUB2 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, set a boot password.

IMPORTANT: Boot Password and Splash Screen

If you use a boot password for GRUB2, the usual splash screen is not displayed.

As the user `root`, proceed as follows to set a boot password:

- 1 At the root prompt, encrypt the password using `grub2-mkpasswd-pbkdf2`:

```
# grub2-mkpasswd-pbkdf2
Password: ****
Reenter password: ****
PBKDF2 hash of your password is
grub.pbkdf2.sha512.10000.9CA4611006FE96BC77A...
```

- 2 Paste the encrypted long string into the file `/etc/grub.d/40_custom` together with the `set superusers` command. Remember to keep the commented lines at the beginning:

```
set superusers="root"
password_pbkdf2 root grub.pbkdf2.sha512.10000.9CA4611006FE96BC77A...
```

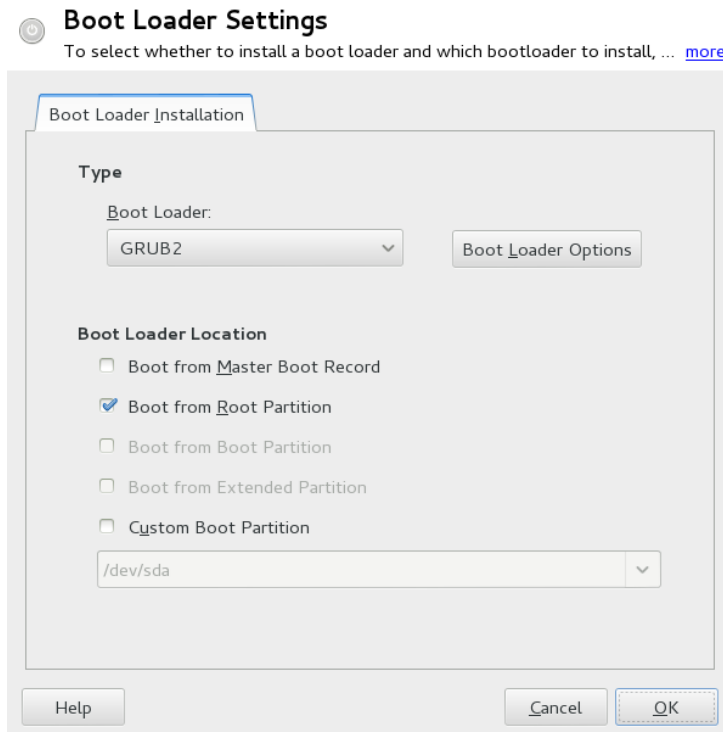
- 3 Run `grub2-mkconfig -o /boot/grub2/grub.cfg` to import the changes into the main configuration file.

After you reboot, you will be prompted for username and password when trying to boot any menu entry. Enter `root` and the password you typed during the `grub2-mkpasswd-pbkdf2` command. If the credentials are correct, the system will boot the selected boot entry.

10.3 Configuring the Boot Loader with YaST

The easiest way to configure general options of the boot loader in your openSUSE system is to use the YaST module. In the YaST Control Center, select *System > Boot Loader*. As in Figure 9.1, “Boot Loader Settings” (page 160), this shows the current boot loader configuration of your system and allows you to make changes.

Figure 10.1: *Boot Loader Settings*



Use the *Boot Loader Installation* tab to view and change settings related to type, location and advanced loader settings. To use the GRUB2 boot loader, make sure it is selected from the list of available boot loaders.

10.3.1 Adjusting the Default Boot Entry

To change the system that is booted by default, proceed as follows:

Procedure 10.1: *Setting the Default System*

- 1 Click *Boot Loader Options* and open the *Default Boot Section* list.
- 2 Select the desired entry from the list. Note that the “>” sign in the boot entry name delimits boot section and its subsection.
- 3 Click *OK* to activate these changes.

10.3.2 Modifying the Boot Loader Location

To modify the location of the boot loader, follow these steps:

Procedure 10.2: *Changing the Boot Loader Location*

- 1 Select the *Boot Loader Installation* tab and then choose one of the following options for *Boot Loader Location*:

Boot from Master Boot Record

This installs the boot loader in the MBR of the first disk (according to the boot sequence preset in the BIOS).

Boot from Root Partition

This installs the boot loader in the boot sector of the / partition (this is the default).

Boot from Boot Partition

This installs the boot loader in the boot sector of the /boot partition.

Boot from Extended Partition

This installs the boot loader in the extended partition container.

Custom Boot Partition

Use this option to specify the location of the boot loader manually.

- 2 Click *OK* to apply your changes.

10.3.3 Changing the Boot Loader Time-Out

The boot loader does not boot the default system immediately. During the time-out, you can select the system to boot or write some kernel parameters. To set the boot loader time-out, proceed as follows:

Procedure 10.3: *Changing the Boot Loader Time-Out*

- 1 Open the *Boot Loader Installation* tab.
- 2 Click *Boot Loader Options*.
- 3 Change the value of *Time-Out in Seconds* by typing in a new value and clicking the appropriate arrow key with your mouse, or by using the arrow keys on the keyboard.
- 4 Click *OK* twice to save the changes.

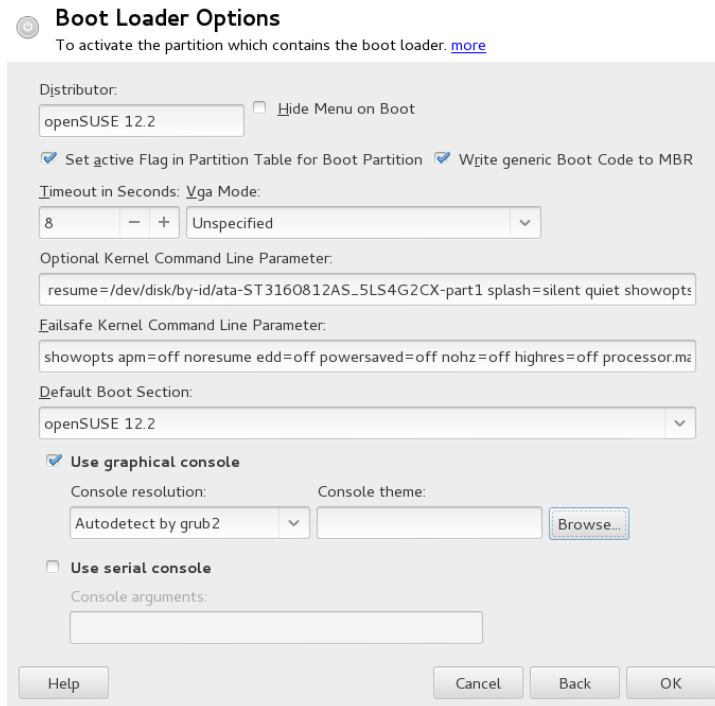
WARNING: Timeout of 0 Seconds

When setting the timeout to 0 seconds, you will not be able to access GRUB2 during boot time. When having set the default boot option to a non-Linux operation system at the same time, this effectively disables access to the Linux system.

10.3.4 Configuring Advanced Options

Advanced boot options can be configured via *Boot Loader Installation > Boot Loader Options*. Normally, it should not be necessary to change the default settings.

Figure 10.2: *Boot Loader Options*



Boot Loader Options
To activate the partition which contains the boot loader: [more](#)

Distributor:
openSUSE 12.2 ☐ Hide Menu on Boot

☒ Set active Flag in Partition Table for Boot Partition ☒ Write generic Boot Code to MBR

Timeout in Seconds: Vga Mode:
8 - + Unspecified

Optional Kernel Command Line Parameter:
resume=/dev/disk/by-id/ata-ST3160812AS_5LS4G2CX-part1 splash=silent quiet showopts

Failsafe Kernel Command Line Parameter:
showopts apm=off noresume edd=off powersaved=off nohz=off highres=off processor.m

Default Boot Section:
openSUSE 12.2

☒ Use graphical console
Console resolution: Autodetect by grub2 Console theme:

☐ Use serial console
Console arguments:

Set Active Flag in Partition Table for Boot Partition

Activates the partition that contains the boot loader. Some legacy operating systems (such as Windows 98) can only boot from an active partition.

Write Generic Boot Code to MBR

Replaces the current MBR with generic, operating system independent code.

Hide Menu on Boot

Hides the boot menu and boots the default entry.

Use graphical console

When checked, the boot menu appears on a graphical splash screen rather than in a text mode. The resolution of the boot screen can be then set from the *Console*

resolution list, and also graphical theme definition file can be specified with the *Console theme* file-chooser.

Use Serial Console

If your machine is controlled via a serial console, activate this option and specify which COM port to use at which speed. See `info grub` or <http://www.gnu.org/software/grub/manual/grub.html#Serial-terminal>

10.3.5 Changing Boot Loader Type

Set the boot loader type in *Boot Loader Installation*. The default boot loader in openSUSE is GRUB2. To use GRUB, LILO or ELILO, proceed as follows:

WARNING: LILO is unsupported

Using LILO is not recommended—it is unsupported on openSUSE. Only use it in special cases.

Procedure 10.4: *Changing the Boot Loader Type*

- 1 Select the *Boot Loader Installation* tab.
- 2 For *Boot Loader*, select *GRUB* or *LILO*.
- 3 In the dialog box that opens, select one of the following actions:

Propose New Configuration

Have YaST propose a new configuration.

Convert Current Configuration

Have YaST convert the current configuration. When converting the configuration, some settings may be lost.

Start New Configuration from Scratch

Write a custom configuration. This action is not available during the installation of openSUSE.

Read Configuration Saved on Disk

Load your own bootloader configuration file. This action is not available during the installation of openSUSE.

4 Click *OK* two times to save the changes.

During the conversion, the old GRUB2 configuration is saved to the disk. To use it, simply change the boot loader type back to GRUB2 and choose *Restore Configuration Saved before Conversion*. This action is available only on an installed system.

NOTE: Custom Boot Loader

To use a boot loader other than GRUB2, GRUB, or LILO, select *Do Not Install Any Boot Loader*. Read the documentation of your boot loader carefully before choosing this option.

10.4 For More Information

Extensive information about GRUB2 is available at <http://www.gnu.org/software/grub/>. Also refer to the `grub` info page. You can also search for the keyword “GRUB2” in the Support Database at http://en.opensuse.org/Portal:Support_database to get information about special issues.

Special System Features

This chapter starts with information about various software packages, the virtual consoles and the keyboard layout. We talk about software components like `bash`, `cron` and `logrotate`, because they were changed or enhanced during the last release cycles. Even if they are small or considered of minor importance, users may want to change their default behavior, because these components are often closely coupled with the system. The chapter concludes with a section about language and country-specific settings (I18N and L10N).

11.1 Information about Special Software Packages

The programs `bash`, `cron`, `logrotate`, `locate`, `ulimit` and `free` are very important for system administrators and many users. Man pages and info pages are two useful sources of information about commands, but both are not always available. GNU Emacs is a popular and very configurable text editor.

11.1.1 The `bash` Package and `/etc/profile`

Bash is the default system shell. When used as a login shell, it reads several initialization files. Bash processes them in the order they appear in this list:

1. `/etc/profile`

2. `~/.profile`

3. `/etc/bash.bashrc`

4. `~/.bashrc`

Make custom settings in `~/.profile` or `~/.bashrc`. To ensure the correct processing of these files, it is necessary to copy the basic settings from `/etc/skel/.profile` or `/etc/skel/.bashrc` into the home directory of the user. It is recommended to copy the settings from `/etc/skel` after an update. Execute the following shell commands to prevent the loss of personal adjustments:

```
mv ~/.bashrc ~/.bashrc.old
cp /etc/skel/.bashrc ~/.bashrc
mv ~/.profile ~/.profile.old
cp /etc/skel/.profile ~/.profile
```

Then copy personal adjustments back from the `*.old` files.

11.1.2 The cron Package

If you want to run commands regularly and automatically in the background at predefined times, cron is the tool to use. cron is driven by specially formatted time tables. Some of them come with the system and users can write their own tables if needed.

The cron tables are located in `/var/spool/cron/tabs`. `/etc/crontab` serves as a systemwide cron table. Enter the username to run the command directly after the time table and before the command. In Example 11.1, “Entry in `/etc/crontab`” (page 188), `root` is entered. Package-specific tables, located in `/etc/cron.d`, have the same format. See the `cron` man page (`man cron`).

Example 11.1: *Entry in `/etc/crontab`*

```
1-59/5 * * * * root test -x /usr/sbin/atrun && /usr/sbin/atrun
```

You cannot edit `/etc/crontab` by calling the command `crontab -e`. This file must be loaded directly into an editor, then modified and saved.

A number of packages install shell scripts to the directories `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly` and `/etc/cron.monthly`, whose execution is controlled by `/usr/lib/cron/run-crons`. `/usr/lib/cron/run-crons` is run every 15 minutes from the main table (`/etc/crontab`). This guarantees that processes that may have been neglected can be run at the proper time.

To run the `hourly`, `daily` or other periodic maintenance scripts at custom times, remove the time stamp files regularly using `/etc/crontab` entries (see Example 11.2, “`/etc/crontab`: Remove Time Stamp Files” (page 189), which removes the `hourly` one before every full hour, the `daily` one once a day at 2:14 a.m., etc.).

Example 11.2: */etc/crontab: Remove Time Stamp Files*

```
59 * * * * root rm -f /var/spool/cron/lastrun/cron.hourly
14 2 * * * root rm -f /var/spool/cron/lastrun/cron.daily
29 2 * * 6 root rm -f /var/spool/cron/lastrun/cron.weekly
44 2 1 * * root rm -f /var/spool/cron/lastrun/cron.monthly
```

Or you can set `DAILY_TIME` in `/etc/sysconfig/cron` to the time at which `cron.daily` should start. The setting of `MAX_NOT_RUN` ensures that the daily tasks get triggered to run, even if the user did not turn on the computer at the specified `DAILY_TIME` for a longer period of time. The maximum value of `MAX_NOT_RUN` is 14 days.

The daily system maintenance jobs are distributed to various scripts for reasons of clarity. They are contained in the package `aaa_base`. `/etc/cron.daily` contains, for example, the components `suse.de-backup-rpmdb`, `suse.de-clean-tmp` or `suse.de-cron-local`.

11.1.3 Log Files: Package `logrotate`

There are a number of system services (*daemons*) that, along with the kernel itself, regularly record the system status and specific events onto log files. This way, the administrator can regularly check the status of the system at a certain point in time, recognize errors or faulty functions and troubleshoot them with pinpoint precision. These log files are normally stored in `/var/log` as specified by FHS and grow on a daily basis. The `logrotate` package helps control the growth of these files.

Configure `logrotate` with the file `/etc/logrotate.conf`. In particular, the `include` specification primarily configures the additional files to read. Programs that produce log files install individual configuration files in `/etc/logrotate.d`. For example, such files ship with the packages `apache2` (`/etc/logrotate.d/apache2`) and `syslogd` (`/etc/logrotate.d/syslog`).

Example 11.3: *Example for `/etc/logrotate.conf`*

```
# see "man logrotate" for details
# rotate log files weekly
weekly
```

```
# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own lastlog or wtmp - we'll rotate them here
#/var/log/wtmp {
#   monthly
#   create 0664 root utmp
#   rotate 1
#}

# system-specific logs may be also be configured here.
```

logrotate is controlled through cron and is called daily by `/etc/cron.daily/logrotate`.

IMPORTANT

The `create` option reads all settings made by the administrator in `/etc/permissions*`. Ensure that no conflicts arise from any personal modifications.

11.1.4 The locate Command

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

11.1.5 The ulimit Command

With the `ulimit` (*user limits*) command, it is possible to set limits for the use of system resources and to have these displayed. `ulimit` is especially useful for limiting available memory for applications. With this, an application can be prevented from

co-opting too much of the system resources and slowing or even hanging up the operating system.

`ulimit` can be used with various options. To limit memory usage, use the options listed in Table 11.1, “`ulimit`: Setting Resources for the User” (page 191).

Table 11.1: *ulimit: Setting Resources for the User*

<code>-m</code>	The maximum resident set size
<code>-v</code>	The maximum amount of virtual memory available to the shell
<code>-s</code>	The maximum size of the stack
<code>-c</code>	The maximum size of core files created
<code>-a</code>	All current limits are reported

Systemwide entries can be made in `/etc/profile`. There, enable creation of core files (needed by programmers for *debugging*). A normal user cannot increase the values specified in `/etc/profile` by the system administrator, but can make special entries in `~/ .bashrc`.

Example 11.4: *ulimit: Settings in `~/ .bashrc`*

```
# Limits maximum resident set size (physical memory):
ulimit -m 98304

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

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

IMPORTANT

Not all shells support `ulimit` directives. PAM (for instance, `pam_limits`) offers comprehensive adjustment possibilities if you depend on encompassing settings for these restrictions.

11.1.6 The free Command

The `free` command displays the total amount of free and used physical memory and swap space in the system, as well as the buffers and cache consumed by the kernel. The concept of *available RAM* dates back to before the days of unified memory management. The slogan *free memory is bad memory* applies well to Linux. As a result, Linux has always made the effort to balance out caches without actually allowing free or unused memory.

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

The kernel also contains other caches, such as the *slab cache*, where the caches used for network access are stored. This may explain the differences between the counters in `/proc/meminfo`. Most, but not all, of them can be accessed via `/proc/slabinfo`.

However, if your goal is to find out how much RAM is currently being used, find this information in `/proc/meminfo`.

11.1.7 Man Pages and Info Pages

For some GNU applications (such as `tar`), the man pages are no longer maintained. For these commands, use the `--help` option to get a quick overview of the info pages, which provide more in-depth instructions. Info is GNU's hypertext system. Read an introduction to this system by entering `info info`. Info pages can be viewed with Emacs by entering `emacs -f info` or directly in a console with `info`. You can also use `tkinfo`, `xinfo` or the help system to view info pages.

11.1.8 Selecting Man Pages Using the man Command

To read a man page enter `man man_page`. If a man page with the same name exists in different sections, they will all be listed with the corresponding section numbers. Select the one to display. If you don't enter a section number within a few seconds, the first man page will be displayed.

If you want to change this to the default system behavior, set `MAN_POSIXLY_CORRECT=1` in a shell initialization file such as `~/.bashrc`.

11.1.9 Settings for GNU Emacs

GNU Emacs is a complex work environment. The following sections cover the configuration files processed when GNU Emacs is started. More information is available at <http://www.gnu.org/software/emacs/>.

On start-up, Emacs reads several files containing the settings of the user, system administrator and distributor for customization or preconfiguration. The initialization file `~/.emacs` is installed to the home directories of the individual users from `/etc/skel/.emacs`, in turn, reads the file `/etc/skel/.gnu-emacs`. To customize the program, copy `.gnu-emacs` to the home directory (with `cp /etc/skel/.gnu-emacs ~/.gnu-emacs`) and make the desired settings there.

`.gnu-emacs` defines the file `~/.gnu-emacs-custom` as `custom-file`. If users make settings with the `customize` options in Emacs, the settings are saved to `~/.gnu-emacs-custom`.

With openSUSE, the `emacs` package installs the file `site-start.el` in the directory `/usr/share/emacs/site-lisp`. The file `site-start.el` is loaded before the initialization file `~/.emacs`. Among other things, `site-start.el` ensures that special configuration files distributed with Emacs add-on packages, such as `psgml`, are loaded automatically. Configuration files of this type are located in `/usr/share/emacs/site-lisp`, too, and always begin with `suse-start-`. The local system administrator can specify systemwide settings in `default.el`.

More information about these files is available in the Emacs info file under *Init File*: <info:/emacs/InitFile>. Information about how to disable the loading of these files (if necessary) is also provided at this location.

The components of Emacs are divided into several packages:

- The base package `emacs`.
- `emacs-x11` (usually installed): the program *with* X11 support.
- `emacs-nox`: the program *without* X11 support.
- `emacs-info`: online documentation in info format.

- `emacs-el`: the uncompiled library files in Emacs Lisp. These are not required at runtime.
- Numerous add-on packages can be installed if needed: `emacs-auctex` (LaTeX), `psgml` (SGML and XML), `gnuserv` (client and server operation) and others.

11.2 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` through `Alt + F6`. The seventh console is reserved for X and the tenth console shows kernel messages. More or fewer consoles can be assigned by modifying the file `/etc/inittab`.

To switch to a console from X without shutting it down, use `Ctrl + Alt + F1` to `Ctrl + Alt + F6`. To return to X, press `Alt + F7`.

11.3 Keyboard Mapping

To standardize the keyboard mapping of programs, changes were made to the following files:

```
/etc/inputrc
/etc/X11/Xmodmap
/etc/skel/.emacs
/etc/skel/.gnu-emacs
/etc/skel/.vimrc
/etc/csh.cshrc
/etc/termcap
/usr/share/terminfo/x/xterm
/usr/share/X11/app-defaults/XTerm
/usr/share/emacs/VERSION/site-lisp/term/*.el
```

These changes only affect applications that use `terminfo` entries or whose configuration files are changed directly (`vi`, `emacs`, etc.). Applications not shipped with the system should be adapted to these defaults.

Under X, the compose key (multikey) can be enabled as explained in `/etc/X11/Xmodmap`.

Further settings are possible using the X Keyboard Extension (XKB). This extension is also used by the desktop environments GNOME (gswitchit) and KDE (kxkb).

TIP: For More Information

Information about XKB is available in the documents listed in `/usr/share/doc/packages/xkeyboard-config` (part of the `xkeyboard-config` package).

11.4 Language and Country-Specific Settings

The system is, to a very large extent, internationalized and can be modified for local needs. Internationalization (*I18N*) allows specific localizations (*L10N*). The abbreviations *I18N* and *L10N* are derived from the first and last letters of the words and, in between, the number of letters omitted.

Settings are made with `LC_` variables defined in the file `/etc/sysconfig/language`. This refers not only to *native language support*, but also to the categories *Messages* (Language), *Character Set*, *Sort Order*, *Time and Date*, *Numbers* and *Money*. Each of these categories can be defined directly with its own variable or indirectly with a master variable in the file `language` (see the `locale` man page).

`RC_LC_MESSAGES`, `RC_LC_CTYPE`, `RC_LC_COLLATE`, `RC_LC_TIME`,
`RC_LC_NUMERIC`, `RC_LC_MONETARY`

These variables are passed to the shell without the `RC_` prefix and represent the listed categories. The shell profiles concerned are listed below. The current setting can be shown with the command `locale`.

`RC_LC_ALL`

This variable, if set, overwrites the values of the variables already mentioned.

`RC_LANG`

If none of the previous variables are set, this is the fallback. By default, only `RC_LANG` is set. This makes it easier for users to enter their own values.

`ROOT_USES_LANG`

A yes or no variable. If set to no, `root` always works in the POSIX environment.

The variables can be set with 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>]]
```

11.4.1 Some Examples

You should always set the language and country codes together. Language settings follow the standard ISO 639 available at <http://www.evertype.com/standards/iso639/iso639-en.html> and <http://www.loc.gov/standards/iso639-2/>. Country codes are listed in ISO 3166 available at http://www.din.de/gremien/nas/nabd/iso3166ma/cod1-stp1/en_listp1.html.

It only makes sense to set values for which usable description files can be found in `/usr/lib/locale`. Additional description files can be created from the files in `/usr/share/i18n` using the command `localedef`. The description files are part of the `glibc-i18ndata` package. A description file for `en_US.UTF-8` (for English and United States) can be created with:

```
localedef -i en_US -f UTF-8 en_US.UTF-8
```

```
LANG=en_US.UTF-8
```

This is the default setting if American English is selected during installation. If you selected another language, that language is enabled but still with UTF-8 as the character encoding.

```
LANG=en_US.ISO-8859-1
```

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

```
LANG=en_IE@euro
```

The above example explicitly includes the Euro sign in a language setting. This setting is basically obsolete now, as UTF-8 also covers the Euro symbol. It is only useful if an application supports ISO-8859-15 and not UTF-8.

In former releases, it was necessary to run `SuSEconfig` after doing any changes to `/etc/sysconfig/language`. `SuSEconfig` then wrote the changes to `/etc/SuSEconfig/profile` and `/etc/SuSEconfig/csh.login`. Upon login, these files were read by `/etc/profile` (for the `Bash`) or by `/etc/csh.login` (for the `tcsh`) .

In recent releases, `/etc/SuSEconfig/profile` has been replaced with `/etc/profile.d/lang.sh`, and `/etc/SuSEconfig/csh.login` with `/etc/profile.d/lang.csh`. But if they exist, both legacy file are still read upon login.

The process chain is now as follows:

- For the `Bash`: `/etc/profile` reads `/etc/profile.d/lang.sh` which, in turn, analyzes `/etc/sysconfig/language`.
- For `tcsh`: At login, `/etc/csh.login` reads `/etc/profile.d/lang.csh` which, in turn, analyzes `/etc/sysconfig/language`.

This ensures that any changes to `/etc/sysconfig/language` are available at the next login to the respective shell, without having to run `SuSEconfig` first.

Users can override the system defaults by editing their `~/.bashrc` accordingly. For instance, if you do not want to use the systemwide `en_US` for program messages, include `LC_MESSAGES=es_ES` so that messages are displayed in Spanish instead.

11.4.2 Locale Settings in `~/.i18n`

If you are not satisfied with locale system defaults, change the settings in `~/.i18n` according to the `Bash` scripting syntax. Entries in `~/.i18n` override system defaults from `/etc/sysconfig/language`. Use the same variable names but without the `RC_` namespace prefixes. For example, use `LANG` instead of `RC_LANG`:

```
LANG=cs_CZ.UTF-8
LC_COLLATE=C
```

11.4.3 Settings for Language Support

Files in the category *Messages* are, as a rule, only stored in the corresponding language directory (like `en`) to have a fallback. If you set `LANG` to `en_US` and the message file

in `/usr/share/locale/en_US/LC_MESSAGES` does not exist, it falls back to `/usr/share/locale/en/LC_MESSAGES`.

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

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

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

If desired, use the Norwegian variants Nynorsk and Bokmål instead (with additional fallback to `no`):

```
LANG="nn_NO"
```

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

or

```
LANG="nb_NO"
```

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

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

One problem that can arise is a separator used to delimit groups of digits not being recognized properly. This occurs if `LANG` is set to only a two-letter language code like `de`, but the definition file `glibc` uses is located in `/usr/share/lib/de_DE/LC_NUMERIC`. Thus `LC_NUMERIC` must be set to `de_DE` to make the separator definition visible to the system.

11.4.4 For More Information

- *The GNU C Library Reference Manual*, Chapter “Locales and Internationalization”. It is included in `glibc-info`.
- Markus Kuhn, *UTF-8 and Unicode FAQ for Unix/Linux*, currently at <http://www.cl.cam.ac.uk/~mgk25/unicode.html>.
- *Unicode-Howto* by Bruno Haible, available at <http://tldp.org/HOWTO/Unicode-HOWTO-1.html>.

Dynamic Kernel Device Management with `udev`

The kernel can add or remove almost any device in a running system. Changes in the device state (whether a device is plugged in or removed) need to be propagated to userspace. Devices need to be configured as soon as they are plugged in and recognized. Users of a certain device need to be informed about any changes in this device's recognized state. `udev` provides the needed infrastructure to dynamically maintain the device node files and symbolic links in the `/dev` directory. `udev` rules provide a way to plug external tools into the kernel device event processing. This enables you to customize `udev` device handling by, for example, adding certain scripts to execute as part of kernel device handling, or request and import additional data to evaluate during device handling.

12.1 The `/dev` Directory

The device nodes in the `/dev` directory provide access to the corresponding kernel devices. With `udev`, the `/dev` directory reflects the current state of the kernel. Every kernel device has one corresponding device file. If a device is disconnected from the system, the device node is removed.

The content of the `/dev` directory is kept on a temporary file system and all files are rendered at every system start-up. Manually created or modified files do not, by design, survive a reboot. Static files and directories that should always be present in the `/dev` directory regardless of the state of the corresponding kernel device can be placed in the `/lib/udev/devices` directory. At system start-up, the contents of that directory is copied to the `/dev` directory with the same ownership and permissions as the files in `/lib/udev/devices`.

12.2 Kernel uevents and udev

The required device information is exported by the `sysfs` file system. For every device the kernel has detected and initialized, a directory with the device name is created. It contains attribute files with device-specific properties.

Every time a device is added or removed, the kernel sends a uevent to notify `udev` of the change. The `udev` daemon reads and parses all provided rules from the `/etc/udev/rules.d/*.rules` files once at start-up and keeps them in memory. If rules files are changed, added or removed, the daemon can reload the in-memory representation of all rules with the command `udevadm control reload_rules`. This is also done when running `/etc/init.d/boot.udev reload`. For more details on `udev` rules and their syntax, refer to Section 12.6, “Influencing Kernel Device Event Handling with `udev` Rules” (page 203).

Every received event is matched against the set of provided rules. The rules can add or change event environment keys, request a specific name for the device node to create, add symlinks pointing to the node or add programs to run after the device node is created. The driver core `uevents` are received from a kernel netlink socket.

12.3 Drivers, Kernel Modules and Devices

The kernel bus drivers probe for devices. For every detected device, the kernel creates an internal device structure while the driver core sends a uevent to the `udev` daemon. Bus devices identify themselves by a specially-formatted ID, which tells what kind of device it is. Usually these IDs consist of vendor and product ID and other subsystem-specific values. Every bus has its own scheme for these IDs, called `MODALIAS`. The kernel takes the device information, composes a `MODALIAS` ID string from it and sends that string along with the event. For a USB mouse, it looks like this:

```
MODALIAS=usb:v046DpC03Ed2000dc00dsc00dp00ic03isc01ip02
```

Every device driver carries a list of known aliases for devices it can handle. The list is contained in the kernel module file itself. The program `depmod` reads the ID lists and creates the file `modules.alias` in the kernel's `/lib/modules` directory

for all currently available modules. With this infrastructure, module loading is as easy as calling `modprobe` for every event that carries a `MODALIAS` key. If `modprobe $MODALIAS` is called, it matches the device alias composed for the device with the aliases provided by the modules. If a matching entry is found, that module is loaded. All this is automatically triggered by `udev`.

12.4 Booting and Initial Device Setup

All device events happening during the boot process before the `udev` daemon is running are lost, because the infrastructure to handle these events resides on the root file system and is not available at that time. To cover that loss, the kernel provides a `uevent` file located in the device directory of every device in the `sysfs` file system. By writing `add` to that file, the kernel resends the same event as the one lost during boot. A simple loop over all `uevent` files in `/sys` triggers all events again to create the device nodes and perform device setup.

As an example, a USB mouse present during boot may not be initialized by the early boot logic, because the driver is not available at that time. The event for the device discovery was lost and failed to find a kernel module for the device. Instead of manually searching for possibly connected devices, `udev` just requests all device events from the kernel after the root file system is available, so the event for the USB mouse device just runs again. Now it finds the kernel module on the mounted root file system and the USB mouse can be initialized.

From userspace, there is no visible difference between a device coldplug sequence and a device discovery during runtime. In both cases, the same rules are used to match and the same configured programs are run.

12.5 Monitoring the Running `udev` Daemon

The program `udevadm monitor` can be used to visualize the driver core events and the timing of the `udev` event processes.

```

UEVENT[1185238505.276660] add    /devices/pci0000:00/0000:00:1d.2/usb3/3-1
(usb)
UDEV  [1185238505.279198] add    /devices/pci0000:00/0000:00:1d.2/usb3/3-1
(usb)
UEVENT[1185238505.279527] add    /devices/pci0000:00/0000:00:1d.2/
usb3/3-1/3-1:1.0 (usb)
UDEV  [1185238505.285573] add    /devices/pci0000:00/0000:00:1d.2/
usb3/3-1/3-1:1.0 (usb)
UEVENT[1185238505.298878] add    /devices/pci0000:00/0000:00:1d.2/
usb3/3-1/3-1:1.0/input/input10 (input)
UDEV  [1185238505.305026] add    /devices/pci0000:00/0000:00:1d.2/
usb3/3-1/3-1:1.0/input/input10 (input)
UEVENT[1185238505.305442] add    /devices/pci0000:00/0000:00:1d.2/
usb3/3-1/3-1:1.0/input/input10/mouse2 (input)
UEVENT[1185238505.306440] add    /devices/pci0000:00/0000:00:1d.2/
usb3/3-1/3-1:1.0/input/input10/event4 (input)
UDEV  [1185238505.325384] add    /devices/pci0000:00/0000:00:1d.2/
usb3/3-1/3-1:1.0/input/input10/event4 (input)
UDEV  [1185238505.342257] add    /devices/pci0000:00/0000:00:1d.2/
usb3/3-1/3-1:1.0/input/input10/mouse2 (input)

```

The UEVENT lines show the events the kernel has sent over netlink. The UDEV lines show the finished udev event handlers. The timing is printed in microseconds. The time between UEVENT and UDEV is the time udev took to process this event or the udev daemon has delayed its execution to synchronize this event with related and already running events. For example, events for hard disk partitions always wait for the main disk device event to finish, because the partition events may rely on the data that the main disk event has queried from the hardware.

`udevadm monitor --env` shows the complete event environment:

```

ACTION=add
DEVPATH=/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10
SUBSYSTEM=input
SEQNUM=1181
NAME="Logitech USB-PS/2 Optical Mouse"
PHYS="usb-0000:00:1d.2-1/input0"
UNIQ=""
EV=7
KEY=70000 0 0 0 0
REL=103
MODALIAS=input:b0003v046DpC03Ee0110-e0,1,2,k110,111,112,r0,1,8,amlsfw

```

udev also sends messages to syslog. The default syslog priority that controls which messages are sent to syslog is specified in the udev configuration file `/etc/udev/udev.conf`. The log priority of the running daemon can be changed with `udevadm control log_priority=level/number`.

12.6 Influencing Kernel Device Event Handling with `udev` Rules

A `udev` rule can match any property the kernel adds to the event itself or any information that the kernel exports to `sysfs`. The rule can also request additional information from external programs. Every event is matched against all provided rules. All rules are located in the `/etc/udev/rules.d` directory.

Every line in the rules file contains at least one key value pair. There are two kinds of keys, match and assignment keys. If all match keys match their values, the rule is applied and the assignment keys are assigned the specified value. A matching rule may specify the name of the device node, add symlinks pointing to the node or run a specified program as part of the event handling. If no matching rule is found, the default device node name is used to create the device node. Detailed information about the rule syntax and the provided keys to match or import data are described in the `udev` man page. The following example rules provide a basic introduction to `udev` rule syntax. The example rules are all taken from the `udev` default rule set that is located under `/etc/udev/rules.d/50-udev-default.rules`.

Example 12.1: *Example `udev` Rules*

```
# console
KERNEL=="console", MODE="0600", OPTIONS="last_rule"

# serial devices
KERNEL=="ttyUSB*", ATTRS{product}=="[Pp]alm*Handheld*", SYMLINK+="pilot"

# printer
SUBSYSTEM=="usb", KERNEL=="lp*", NAME="usb/%k", SYMLINK+="usb%k", GROUP="lp"

# kernel firmware loader
SUBSYSTEM=="firmware", ACTION=="add", RUN+="firmware.sh"
```

The `console` rule consists of three keys: one match key (`KERNEL`) and two assignment keys (`MODE`, `OPTIONS`). The `KERNEL` match rule searches the device list for any items of the type `console`. Only exact matches are valid and trigger this rule to be executed. The `MODE` key assigns special permissions to the device node, in this case, read and write permissions to the owner of this device only. The `OPTIONS` key makes this rule the last rule to be applied to any device of this type. Any later rule matching this particular device type does not have any effect.

The `serial devices` rule is not available in `50-udev-default.rules` anymore, but it is still worth considering. It consists of two match keys (`KERNEL` and

ATTRS) and one assign key (SYMLINK). The KERNEL key searches for all devices of the `ttyUSB` type. Using the `*` wild card, this key matches several of these devices. The second match key, ATTRS, checks whether the `product` attribute file in `sysfs` for any `ttyUSB` device contains a certain string. The assign key (SYMLINK) triggers the addition of a symbolic link to this device under `/dev/pilot`. The operator used in this key (`+=`) tells `udev` to additionally perform this action, even if previous or later rules add other symbolic links. As this rule contains two match keys, it is only applied if both conditions are met.

The `printer` rule deals with USB printers and contains two match keys which must both apply to get the entire rule applied (SUBSYSTEM and KERNEL). Three assign keys deal with the naming for this device type (NAME), the creation of symbolic device links (SYMLINK) and the group membership for this device type (GROUP). Using the `*` wild card in the KERNEL key makes it match several `lp` printer devices. Substitutions are used in both, the NAME and the SYMLINK keys to extend these strings by the internal device name. For example, the symlink to the first `lp` USB printer would read `/dev/usblp0`.

The `kernel firmware loader` rule makes `udev` load additional firmware by an external helper script during runtime. The SUBSYSTEM match key searches for the `firmware` subsystem. The ACTION key checks whether any device belonging to the `firmware` subsystem has been added. The `RUN+=` key triggers the execution of the `firmware.sh` script to locate the firmware that is to be loaded.

Some general characteristics are common to all rules:

- Each rule consists of one or more key value pairs separated by a comma.
- A key's operation is determined by the operator. `udev` rules support several different operators.
- Each given value must be enclosed by quotation marks.
- Each line of the rules file represents one rule. If a rule is longer than just one line, use `\` to join the different lines just as you would do in shell syntax.
- `udev` rules support a shell-style pattern that matches the `*`, `?`, and `[]` patterns.
- `udev` rules support substitutions.

12.6.1 Using Operators in `udev` Rules

Creating keys you can choose from several different operators, depending on the type of key you want to create. Match keys will normally just be used to find a value that either matches or explicitly mismatches the search value. Match keys contain either of the following operators:

`==`

Compare for equality. If the key contains a search pattern, all results matching this pattern are valid.

`!=`

Compare for non-equality. If the key contains a search pattern, all results matching this pattern are valid.

Any of the following operators can be used with assign keys:

`=`

Assign a value to a key. If the key previously consisted of a list of values, the key resets and only the single value is assigned.

`+=`

Add a value to a key that contains a list of entries.

`:=`

Assign a final value. Disallow any later change by later rules.

12.6.2 Using Substitutions in `udev` Rules

`udev` rules support the use of placeholders and substitutions. Use them in a similar fashion as you would do in any other scripts. The following substitutions can be used with `udev` rules:

`%r, $root`

The device directory, `/dev` by default.

`%p, $devpath`

The value of `DEVPATH`.

`%k, $kernel`

The value of `KERNEL` or the internal device name.

`%n, $number`

The device number.

`%N, $tempnode`

The temporary name of the device file.

`%M, $major`

The major number of the device.

`%m, $minor`

The minor number of the device.

`%s{attribute}, $attr{attribute}`

The value of a `sysfs` attribute (specified by *attribute*).

`%E{variable}, $attr{variable}`

The value of an environment variable (specified by *variable*).

`%c, $result`

The output of PROGRAM.

`%%`

The `%` character.

`$$`

The `$` character.

12.6.3 Using `udev` Match Keys

Match keys describe conditions that must be met before a `udev` rule can be applied. The following match keys are available:

`ACTION`

The name of the event action, for example, `add` or `remove` when adding or removing a device.

`DEVPATH`

The device path of the event device, for example, `DEVPATH=/bus/pci/drivers/ipw3945` to search for all events related to the `ipw3945` driver.

`KERNEL`

The internal (kernel) name of the event device.

SUBSYSTEM

The subsystem of the event device, for example, `SUBSYSTEM=usb` for all events related to USB devices.

ATTR{ *filename* }

`sysfs` attributes of the event device. To match a string contained in the vendor attribute file name, you could use `ATTR{vendor}=="On[sS]tream"`, for example.

KERNELS

Let `udev` search the device path upwards for a matching device name.

SUBSYSTEMS

Let `udev` search the device path upwards for a matching device subsystem name.

DRIVERS

Let `udev` search the device path upwards for a matching device driver name.

ATTRS{ *filename* }

Let `udev` search the device path upwards for a device with matching `sysfs` attribute values.

ENV{ *key* }

The value of an environment variable, for example,

`ENV{ID_BUS}="ieee1394` to search for all events related to the FireWire bus ID.

PROGRAM

Let `udev` execute an external program. To be successful, the program must return with exit code zero. The program's output, printed to `stdout`, is available to the `RESULT` key.

RESULT

Match the output string of the last `PROGRAM` call. Either include this key in the same rule as the `PROGRAM` key or in a later one.

12.6.4 Using `udev` Assign Keys

In contrast to the match keys described above, assign keys do not describe conditions that must be met. They assign values, names and actions to the device nodes maintained by `udev`.

NAME

The name of the device node to be created. Once a rule has set a node name, all other rules with a `NAME` key for this node are ignored.

SYMLINK

The name of a symlink related to the node to be created. Multiple matching rules can add symlinks to be created with the device node. You can also specify multiple symlinks for one node in one rule using the space character to separate the symlink names.

OWNER, GROUP, MODE

The permissions for the new device node. Values specified here overwrite anything that has been compiled in.

ATTR{*key*}

Specify a value to be written to a `sysfs` attribute of the event device. If the `==` operator is used, this key is also used to match against the value of a `sysfs` attribute.

ENV{*key*}

Tell `udev` to export a variable to the environment. If the `==` operator is used, this key is also used to match against an environment variable.

RUN

Tell `udev` to add a program to the list of programs to be executed for this device. Keep in mind to restrict this to very short tasks to avoid blocking further events for this device.

LABEL

Add a label where a `GOTO` can jump to.

GOTO

Tell `udev` to skip a number of rules and continue with the one that carries the label referenced by the `GOTO` key.

IMPORT{*type*}

Load variables into the event environment such as the output of an external program. `udev` imports variables of several different types. If no type is specified, `udev` tries to determine the type itself based on the executable bit of the file permissions.

- `program` tells `udev` to execute an external program and import its output.

- `file` tells `udev` to import a text file.
- `parent` tells `udev` to import the stored keys from the parent device.

WAIT_FOR_SYSFS

Tells `udev` to wait for the specified `sysfs` file to be created for a certain device. For example, `WAIT_FOR_SYSFS="ioerr_cnt"` informs `udev` to wait until the `ioerr_cnt` file has been created.

OPTIONS

The `OPTION` key may have several possible values:

- `last_rule` tells `udev` to ignore all later rules.
- `ignore_device` tells `udev` to ignore this event completely.
- `ignore_remove` tells `udev` to ignore all later remove events for the device.
- `all_partitions` tells `udev` to create device nodes for all available partitions on a block device.

12.7 Persistent Device Naming

The dynamic device directory and the `udev` rules infrastructure make it possible to provide stable names for all disk devices—regardless of their order of recognition or the connection used for the device. Every appropriate block device the kernel creates is examined by tools with special knowledge about certain buses, drive types or file systems. Along with the dynamic kernel-provided device node name, `udev` maintains classes of persistent symbolic links pointing to the device:

```
/dev/disk
|-- by-id
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B -> ../../sda
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part1 -> ../../sda1
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part6 -> ../../sda6
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part7 -> ../../sda7
|   |-- usb-Generic_STORAGE_DEVICE_02773 -> ../../sdd
|   `-- usb-Generic_STORAGE_DEVICE_02773-part1 -> ../../sdd1
|-- by-label
|   |-- Photos -> ../../sdd1
|   |-- SUSE10 -> ../../sda7
|   `-- devel -> ../../sda6
|-- by-path
```

```

| |-- pci-0000:00:1f.2-scsi-0:0:0:0 -> ../../sda
| |-- pci-0000:00:1f.2-scsi-0:0:0:0-part1 -> ../../sda1
| |-- pci-0000:00:1f.2-scsi-0:0:0:0-part6 -> ../../sda6
| |-- pci-0000:00:1f.2-scsi-0:0:0:0-part7 -> ../../sda7
| |-- pci-0000:00:1f.2-scsi-1:0:0:0 -> ../../sr0
| |-- usb-02773:0:0:2 -> ../../sdd
| |-- usb-02773:0:0:2-part1 -> ../../sdd1
|-- by-uuid
| |-- 159a47a4-e6e6-40be-a757-a629991479ae -> ../../sda7
| |-- 3e999973-00c9-4917-9442-b7633bd95b9e -> ../../sda6
`-- 4210-8F8C -> ../../sdd1

```

12.8 Files used by udev

`/sys/*`

Virtual file system provided by the Linux kernel, exporting all currently known devices. This information is used by `udev` to create device nodes in `/dev`

`/dev/*`

Dynamically created device nodes and static content copied at boot time from `/lib/udev/devices/*`

The following files and directories contain the crucial elements of the `udev` infrastructure:

`/etc/udev/udev.conf`

Main `udev` configuration file.

`/etc/udev/rules.d/*`

`udev` event matching rules.

`/lib/udev/devices/*`

Static `/dev` content.

`/lib/udev/*`

Helper programs called from `udev` rules.

12.9 For More Information

For more information about the `udev` infrastructure, refer to the following man pages:

udev

General information about udev, keys, rules and other important configuration issues.

udevadm

udevadm can be used to control the runtime behavior of udev, request kernel events, manage the event queue and provide simple debugging mechanisms.

udevd

Information about the udev event managing daemon.

Part IV. Services

Basic Networking

Linux offers the necessary networking tools and features for integration into all types of network structures. Network access using a network card, modem or other device can be configured with YaST. Manual configuration is also possible. In this chapter only the fundamental mechanisms and the relevant network configuration files are covered.

Linux and other Unix operating systems use the TCP/IP protocol. It is not a single network protocol, but a family of network protocols that offer various services. The protocols listed in Table 13.1, “Several Protocols in the TCP/IP Protocol Family” (page 215), are provided for the purpose of exchanging data between two machines via TCP/IP. Networks combined by TCP/IP, comprising a worldwide network, are also referred to as “the Internet.”

RFC stands for *Request for Comments*. RFCs are documents that describe various Internet protocols and implementation procedures for the operating system and its applications. The RFC documents describe the setup of Internet protocols. To expand your knowledge of any of the protocols, refer to the appropriate RFC documents. These are available at <http://www.ietf.org/rfc.html>.

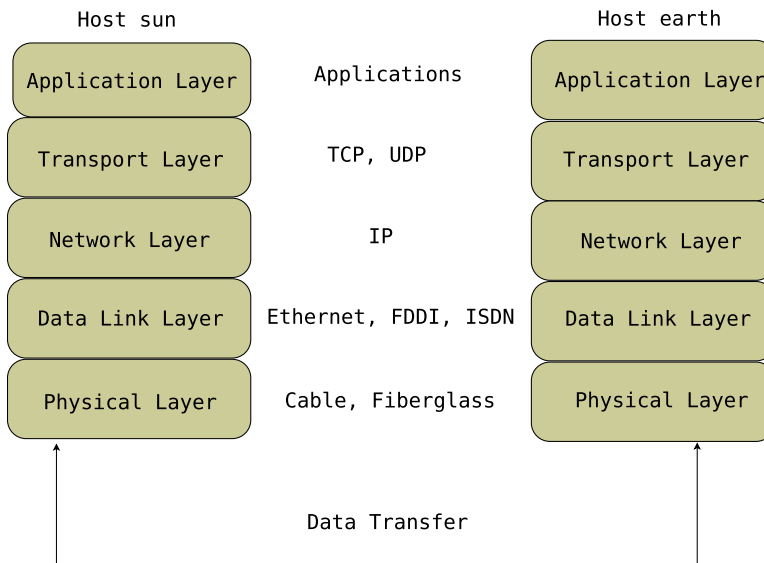
Table 13.1: *Several Protocols in the TCP/IP Protocol Family*

Protocol	Description
TCP	Transmission Control Protocol: a connection-oriented secure protocol. The data to transmit is first sent by the ap-

Protocol	Description
	<p>plication as a stream of data and converted into the appropriate format by the operating system. The data arrives at the respective application on the destination host in the original data stream format it was initially sent. TCP determines whether any data has been lost or jumbled during the transmission. TCP is implemented wherever the data sequence matters.</p>
UDP	<p>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 possible. UDP is suitable for record-oriented applications. It features a smaller latency period than TCP.</p>
ICMP	<p>Internet Control Message Protocol: Essentially, this is not a protocol for the end user, but a special control protocol that issues error reports and can control the behavior of machines participating in TCP/IP data transfer. In addition, it provides a special echo mode that can be viewed using the program ping.</p>
IGMP	<p>Internet Group Management Protocol: This protocol controls machine behavior when implementing IP multicast.</p>

As shown in Figure 13.1, “Simplified Layer Model for TCP/IP” (page 217), data exchange takes place in different layers. The actual network layer is the insecure data transfer via IP (Internet protocol). On top of IP, TCP (transmission control protocol) guarantees, to a certain extent, security of the data transfer. The IP layer is supported by the underlying hardware-dependent protocol, such as ethernet.

Figure 13.1: *Simplified Layer Model for TCP/IP*

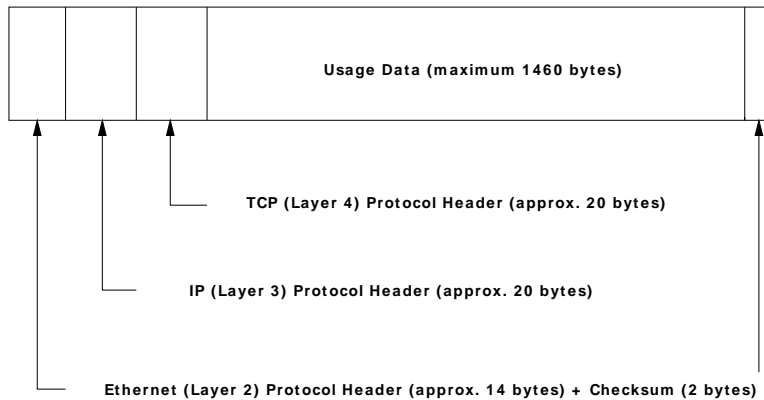


The diagram provides one or two examples for each layer. The layers are ordered according to *abstraction levels*. The lowest layer is very close to the hardware. The uppermost layer, however, is almost a complete abstraction from the hardware. Every layer has its own special function. The special functions of each layer are mostly implicit in their description. The data link and physical layers represent the physical network used, such as ethernet.

Almost all hardware protocols work on a packet-oriented basis. The data to transmit is collected into *packets* (it cannot be sent all at once). The maximum size of a TCP/IP packet is approximately 64 KB. Packets are normally quite smaller, as the network hardware can be a limiting factor. The maximum size of a data packet on an ethernet is about fifteen hundred bytes. The size of a TCP/IP packet is limited to this amount when the data is sent over an ethernet. If more data is transferred, more data packets need to be sent by the operating system.

For the layers to serve their designated functions, additional information regarding each layer must be saved in the data packet. This takes place in the *header* of the packet. Every layer attaches a small block of data, called the protocol header, to the front of each emerging packet. A sample TCP/IP data packet traveling over an ethernet cable is illustrated in Figure 13.2, “TCP/IP Ethernet Packet” (page 218). The proof sum is located at the end of the packet, not at the beginning. This simplifies things for the network hardware.

Figure 13.2: *TCP/IP Ethernet Packet*



When an application sends data over the network, the data passes through each layer, all implemented in the Linux kernel except the physical layer. Each layer is responsible for preparing the data so it can be passed to the next layer. The lowest layer is ultimately responsible for sending the data. The entire procedure is reversed when data is received. Like the layers of an onion, in each layer the protocol headers are removed from the transported data. Finally, the transport layer is responsible for making the data available for use by the applications at the destination. In this manner, one layer only communicates with the layer directly above or below it. For applications, it is irrelevant whether data is transmitted via a 100 Mbit/s FDDI network or via a 56-Kbit/s modem line. Likewise, it is irrelevant for the data line which kind of data is transmitted, as long as packets are in the correct format.

13.1 IP Addresses and Routing

The discussion in this section is limited to IPv4 networks. For information about IPv6 protocol, the successor to IPv4, refer to Section 13.2, “IPv6—The Next Generation Internet” (page 221).

13.1.1 IP Addresses

Every computer on the Internet has a unique 32-bit address. These 32 bits (or 4 bytes) are normally written as illustrated in the second row in Example 13.1, “Writing IP Addresses” (page 219).

Example 13.1: *Writing IP Addresses*

```
IP Address (binary):  11000000 10101000 00000000 00010100
IP Address (decimal):      192.      168.      0.      20
```

In decimal form, the four bytes are written in the decimal number system, separated by periods. The IP address is assigned to a host or a network interface. It can be used only once throughout the world. There are exceptions to this rule, but these are not relevant to the following passages.

The points in IP addresses indicate the hierarchical system. Until the 1990s, IP addresses were strictly categorized in classes. However, this system proved too inflexible and was discontinued. Now, *classless routing* (CIDR, classless interdomain routing) is used.

13.1.2 Netmasks and Routing

Netmasks are used to define the address range of a subnetwork. If two hosts are in the same subnetwork, they can reach each other directly. If they are not in the same subnetwork, they need the address of a gateway that handles all the traffic for the subnetwork. To check if two IP addresses are in the same subnet, simply “AND” both addresses with the netmask. If the result is identical, both IP addresses are in the same local network. If there are differences, the remote IP address, and thus the remote interface, can only be reached over a gateway.

To understand how the netmask works, look at Example 13.2, “Linking IP Addresses to the Netmask” (page 220). The netmask consists of 32 bits that identify how much of an IP address belongs to the network. All those bits that are 1 mark the corresponding bit in the IP address as belonging to the network. All bits that are 0 mark bits inside the subnetwork. This means that the more bits are 1, the smaller the subnetwork is. Because the netmask always consists of several successive 1 bits, it is also possible to just count the number of bits in the netmask. In Example 13.2, “Linking IP Addresses to the Netmask” (page 220) the first net with 24 bits could also be written as 192.168.0.0/24.

Example 13.2: *Linking IP Addresses to the Netmask*

```
IP address (192.168.0.20): 11000000 10101000 00000000 00010100
Netmask   (255.255.255.0): 11111111 11111111 11111111 00000000
-----
Result of the link:      11000000 10101000 00000000 00000000
In the decimal system:   192.      168.      0.      0

IP address (213.95.15.200): 11010101 10111111 00001111 11001000
Netmask   (255.255.255.0): 11111111 11111111 11111111 00000000
-----
Result of the link:      11010101 10111111 00001111 00000000
In the decimal system:   213.      95.      15.      0
```

To give another example: all machines connected with the same ethernet cable are usually located in the same subnetwork and are directly accessible. Even when the subnet is physically divided by switches or bridges, these hosts can still be reached directly.

IP addresses outside the local subnet can only be reached if a gateway is configured for the target network. In the most common case, there is only one gateway that handles all traffic that is external. However, it is also possible to configure several gateways for different subnets.

If a gateway has been configured, all external IP packets are sent to the appropriate gateway. This gateway then attempts to forward the packets in the same manner—from host to host—until it reaches the destination host or the packet's TTL (time to live) expires.

Table 13.2: *Specific Addresses*

Address Type	Description
Base Network Address	This is the netmask AND any address in the network, as shown in Example 13.2, “Linking IP Addresses to the Netmask” (page 220) under Result. This address cannot be assigned to any hosts.
Broadcast Address	This basically says, “Access all hosts in this subnetwork.” To generate this, the netmask is inverted in binary form and linked to the base net-

Address Type	Description
	work address with a logical OR. The above example therefore results in 192.168.0.255. This address cannot be assigned to any hosts.
Local Host	The address 127.0.0.1 is assigned to the “loopback device” on each host. A connection can be set up to your own machine with this address and with all addresses from the complete 127.0.0.0/8 loopback network as defined with IPv4. With IPv6 there is just one loopback address (::1).

Because IP addresses must be unique all over the world, you cannot just select random addresses. There are three address domains to use if you want to set up a private IP-based network. These cannot get any connection from the rest of the Internet, because they cannot be transmitted over the Internet. These address domains are specified in RFC 1597 and listed in Table 13.3, “Private IP Address Domains” (page 221).

Table 13.3: *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

13.2 IPv6—The Next Generation Internet

Due to the emergence of the WWW (World Wide Web), the Internet has experienced explosive growth, with an increasing number of computers communicating

via TCP/IP in the past fifteen years. Since Tim Berners-Lee at CERN (<http://public.web.cern.ch>) invented the WWW in 1990, the number of Internet hosts has grown from a few thousand to about a hundred million.

As mentioned, an IPv4 address consists of only 32 bits. Also, quite a few IP addresses are lost—they cannot be used due to the way in which networks are organized. The number of addresses available in your subnet is two to the power of the number of bits, minus two. A subnetwork has, for example, 2, 6, or 14 addresses available. To connect 128 hosts to the Internet, for example, you need a subnetwork with 256 IP addresses, from which only 254 are usable, because two IP addresses are needed for the structure of the subnetwork itself: the broadcast and the base network address.

Under the current IPv4 protocol, DHCP or NAT (network address translation) are the typical mechanisms used to circumvent the potential address shortage. Combined with the convention to keep private and public address spaces separate, these methods can certainly mitigate the shortage. The problem with them lies in their configuration, which is a chore to set up and a burden to maintain. To set up a host in an IPv4 network, you need a number of address items, such as the host's own IP address, the subnetmask, the gateway address and maybe a name server address. All these items need to be known and cannot be derived from somewhere else.

With IPv6, both the address shortage and the complicated configuration should be a thing of the past. The following sections tell more about the improvements and benefits brought by IPv6 and about the transition from the old protocol to the new one.

13.2.1 Advantages

The most important and most visible improvement brought by the new protocol is the enormous expansion of the available address space. An IPv6 address is made up of 128 bit values instead of the traditional 32 bits. This provides for as many as several quadrillion IP addresses.

However, IPv6 addresses are not only different from their predecessors with regard to their length. They also have a different internal structure that may contain more specific information about the systems and the networks to which they belong. More details about this are found in Section 13.2.2, “Address Types and Structure” (page 224).

The following is a list of some other advantages of the new protocol:

Autoconfiguration

IPv6 makes the network “plug and play” capable, which means that a newly set up system integrates into the (local) network without any manual configuration. The new host uses its automatic configuration mechanism to derive its own address from the information made available by the neighboring routers, relying on a protocol called the *neighbor discovery* (ND) protocol. This method does not require any intervention on the administrator's part and there is no need to maintain a central server for address allocation—an additional advantage over IPv4, where automatic address allocation requires a DHCP server or the usage of ARP and 169.254.0.0/16 addresses.

Nevertheless if a router is connected to a switch, the router should send periodic advertisements with flags telling the hosts of a network how they should interact with each other. For more information, see RFC 2462 and the `radvd.conf(5)` manpage, and RFC 3315.

Mobility

IPv6 makes it possible to assign several addresses to one network interface at the same time. This allows users to access several networks easily, something that could be compared with the international roaming services offered by mobile phone companies: when you take your mobile phone abroad, the phone automatically logs in to a foreign service as soon as it enters the corresponding area, so you can be reached under the same number everywhere and are able to place an outgoing call just like in your home area.

Secure Communication

With IPv4, network security is an add-on function. IPv6 includes IPsec as one of its core features, allowing systems to communicate over a secure tunnel to avoid eavesdropping by outsiders on the Internet.

Backward Compatibility

Realistically, it would be impossible to switch the entire Internet from IPv4 to IPv6 at one time. Therefore, it is crucial that both protocols are able to coexist not only on the Internet, but also on one system. This is ensured by compatible addresses (IPv4 addresses can easily be translated into IPv6 addresses) and through the use of a number of tunnels. See Section 13.2.3, “Coexistence of IPv4 and IPv6” (page 228). 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 pre-defined groups to address all name servers (the *all name servers multicast group*), for example, or all routers (the *all routers multicast group*).

13.2.2 Address Types and Structure

As mentioned, the current IP protocol is lacking in two important aspects: there is an increasing shortage of IP addresses and configuring the network and maintaining the routing tables is becoming a more complex and burdensome task. IPv6 solves the first problem by expanding the address space to 128 bits. The second one is countered by introducing a hierarchical address structure, combined with sophisticated techniques to allocate network addresses, as well as *multihoming* (the ability to assign several addresses to one device, giving access to several networks).

When dealing with IPv6, it is useful to know about three different types of addresses:

Unicast

Addresses of this type are associated with exactly one network interface. Packets with such an address are delivered to only one destination. Accordingly, unicast addresses are used to transfer packets to individual hosts on the local network or the Internet.

Multicast

Addresses of this type relate to a group of network interfaces. Packets with such an address are delivered to all destinations that belong to the group. Multicast addresses are mainly used by certain network services to communicate with certain groups of hosts in a well-directed manner.

Anycast

Addresses of this type are related to a group of interfaces. Packets with such an address are delivered to the member of the group that is closest to the sender, according to the principles of the underlying routing protocol. Anycast addresses are used to make it easier for hosts to find out about servers offering certain services in the given network area. All servers of the same type have the same any-

cast address. Whenever a host requests a service, it receives a reply from the server with the closest location, as determined by the routing protocol. If this server should fail for some reason, the protocol automatically selects the second closest server, then the third one, and so forth.

An IPv6 address is made up of eight four-digit fields, each representing 16 bits, written in hexadecimal notation. They are 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 13.3, “Sample IPv6 Address” (page 225), where all three lines represent the same address.

Example 13.3: *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 13.4, “IPv6 Address Specifying the Prefix Length” (page 225), 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 13.4: *IPv6 Address Specifying the Prefix Length*

```
fe80::10:1000:1a4/64
```

IPv6 knows about several predefined types of prefixes. Some of these are shown in Table 13.4, “Various IPv6 Prefixes” (page 225).

Table 13.4: *Various IPv6 Prefixes*

Prefix (hex)	Definition
00	IPv4 addresses and IPv4 over IPv6 compatibility addresses. These are

Prefix (hex)	Definition
	used to maintain compatibility with IPv4. Their use still requires a router able to translate IPv6 packets into IPv4 packets. Several special addresses, such as the one for the loopback device, have this prefix as well.
2 or 3 as the first digit	Aggregatable global unicast addresses. As is the case with IPv4, an interface can be assigned to form part of a certain subnetwork. Currently, there are the following address spaces: 2001::/16 (production quality address space) and 2002::/16 (6to4 address space).
fe80::/10	Link-local addresses. Addresses with this prefix should not be routed and should therefore only be reachable from within the same subnetwork.
fec0::/10	Site-local addresses. These may be routed, but only within the network of the organization to which they belong. In effect, they are the IPv6 equivalent of the current private network address space, such as 10.x.x.x.
ff	These are multicast addresses.

A unicast address consists of three basic components:

Public Topology

The first part (which also contains one of the prefixes mentioned above) is used to route packets through the public Internet. It includes information about the company or institution that provides the Internet access.

Site Topology

The second part contains routing information about the subnetwork to which to deliver the packet.

Interface ID

The third part identifies the interface to which to deliver the packet. This also allows for the MAC to form part of the address. Given that the MAC is a globally unique, fixed identifier coded into the device by the hardware maker, the configuration procedure is substantially simplified. In fact, the first 64 address bits are consolidated to form the `EUI-64` token, with the last 48 bits taken from the MAC, and the remaining 24 bits containing special information about the token type. This also makes it possible to assign an `EUI-64` token to interfaces that do not have a MAC, such as those based on PPP or ISDN.

On top of this basic structure, IPv6 distinguishes between five different types of unicast addresses:

`::` (unspecified)

This address is used by the host as its source address when the interface is initialized for the first time—when the address cannot yet be determined by other means.

`::1` (loopback)

The address of the loopback device.

IPv4 Compatible Addresses

The IPv6 address is formed by the IPv4 address and a prefix consisting of 96 zero bits. This type of compatibility address is used for tunneling (see Section 13.2.3, “Coexistence of IPv4 and IPv6” (page 228)) to allow IPv4 and IPv6 hosts to communicate with others operating in a pure IPv4 environment.

IPv4 Addresses Mapped to IPv6

This type of address specifies a pure IPv4 address in IPv6 notation.

Local Addresses

There are two address types for local use:

link-local

This type of address can only be used in the local subnetwork. Packets with a source or target address of this type should not be routed to the Internet or other subnetworks. These addresses contain a special prefix (`fe80::/10`) and the interface ID of the network card, with the middle part consisting of

zero bytes. Addresses of this type are used during automatic configuration to communicate with other hosts belonging to the same subnetwork.

site-local

Packets with this type of address may be routed to other subnetworks, but not to the wider Internet—they must remain inside the organization's own network. Such addresses are used for intranets and are an equivalent of the private address space defined by IPv4. They contain a special prefix ($\text{fec0}::/10$), the interface ID, and a 16 bit field specifying the subnetwork ID. Again, the rest is filled with zero bytes.

As a completely new feature introduced with IPv6, each network interface normally gets several IP addresses, with the advantage that several networks can be accessed through the same interface. One of these networks can be configured completely automatically using the MAC and a known prefix with the result that all hosts on the local network can be reached as soon as IPv6 is enabled (using the link-local address). With the MAC forming part of it, any IP address used in the world is unique. The only variable parts of the address are those specifying the *site topology* and the *public topology*, depending on the actual network in which the host is currently operating.

For a host to go back and forth between different networks, it needs at least two addresses. One of them, the *home address*, not only contains the interface ID but also an identifier of the home network to which it normally belongs (and the corresponding prefix). The home address is a static address and, as such, it does not normally change. Still, all packets destined to the mobile host can be delivered to it, regardless of whether it operates in the home network or somewhere outside. This is made possible by the completely new features introduced with IPv6, such as *stateless autoconfiguration* and *neighbor discovery*. In addition to its home address, a mobile host gets one or more additional addresses that belong to the foreign networks where it is roaming. These are called *care-of* addresses. The home network has a facility that forwards any packets destined to the host when it is roaming outside. In an IPv6 environment, this task is performed by the *home agent*, which takes all packets destined to the home address and relays them through a tunnel. On the other hand, those packets destined to the care-of address are directly transferred to the mobile host without any special detours.

13.2.3 Coexistence of IPv4 and IPv6

The migration of all hosts connected to the Internet from IPv4 to IPv6 is a gradual process. Both protocols will coexist for some time to come. The coexistence on one

system is guaranteed where there is a *dual stack* implementation of both protocols. That still leaves the question of how an IPv6 enabled host should communicate with an IPv4 host and how IPv6 packets should be transported by the current networks, which are predominantly IPv4 based. The best solutions offer tunneling and compatibility addresses (see Section 13.2.2, “Address Types and Structure” (page 224)).

IPv6 hosts that are more or less isolated in the (worldwide) IPv4 network can communicate through tunnels: IPv6 packets are encapsulated as IPv4 packets to move them across an IPv4 network. Such a connection between two IPv4 hosts is called a *tunnel*. To achieve this, packets must include the IPv6 destination address (or the corresponding prefix) as well as the IPv4 address of the remote host at the receiving end of the tunnel. A basic tunnel can be configured manually according to an agreement between the hosts' administrators. This is also called *static tunneling*.

However, the configuration and maintenance of static tunnels is often too labor-intensive to use them for daily communication needs. Therefore, IPv6 provides for three different methods of *dynamic tunneling*:

6over4

IPv6 packets are automatically encapsulated as IPv4 packets and sent over an IPv4 network capable of multicasting. IPv6 is tricked into seeing the whole network (Internet) as a huge local area network (LAN). This makes it possible to determine the receiving end of the IPv4 tunnel automatically. However, this method does not scale very well and is also hampered by the fact that IP multicasting is far from widespread on the Internet. Therefore, it only provides a solution for smaller corporate or institutional networks where multicasting can be enabled. The specifications for this method are laid down in RFC 2529.

6to4

With this method, IPv4 addresses are automatically generated from IPv6 addresses, enabling isolated IPv6 hosts to communicate over an IPv4 network. However, a number of problems have been reported regarding the communication between those isolated IPv6 hosts and the Internet. The method is described in RFC 3056.

IPv6 Tunnel Broker

This method relies on special servers that provide dedicated tunnels for IPv6 hosts. It is described in RFC 3053.

13.2.4 Configuring IPv6

To configure IPv6, you normally do not need to make any changes on the individual workstations. IPv6 is enabled by default. You can disable it during installation in the network configuration step described in Section 1.14.2.2, “Network Configuration” (page 30). To disable or enable IPv6 on an installed system, use the YaST *Network Settings* module. On the *Global Options* tab, check or uncheck the *Enable IPv6* option as necessary. To enable or disable IPv6 manually, edit `/etc/modprobe.d/50-ipv6.conf` and restart the system. If you want to enable it temporarily until the next reboot, enter `modprobe -i ipv6` as root. It is basically impossible to unload the `ipv6` module once loaded.

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/quagga* for automatic configuration of both addresses and routing.

Consult the `ifcfg-tunnel(5)` man page to get information about how to set up various types of tunnels using the `/etc/sysconfig/network` files.

13.2.5 For More Information

The above overview does not cover the topic of IPv6 comprehensively. For a more in-depth look at the new protocol, refer to the following online documentation and books:

<http://www.ipv6.org/>

The starting point for everything about IPv6.

<http://www.ipv6day.org>

All information needed to start your own IPv6 network.

<http://www.ipv6-to-standard.org/>

The list of IPv6-enabled products.

<http://www.bieringer.de/linux/IPv6/>

Here, find the Linux IPv6-HOWTO and many links related to the topic.

RFC 2640

The fundamental RFC about IPv6.

IPv6 Essentials

A book describing all the important aspects of the topic is *IPv6 Essentials* by Silvia Hagen (ISBN 0-596-00125-8).

13.3 Name Resolution

DNS assists in assigning an IP address to one or more names and assigning a name to an IP address. In Linux, this conversion is usually carried out by a special type of software known as *bind*. The machine that takes care of this conversion is called a *name server*. The names make up a hierarchical system in which each name component is separated by a period. The name hierarchy is, however, independent of the IP address hierarchy described above.

Consider a complete name, such as `jupiter.example.com`, written in the format `hostname.domain`. A full name, referred to as a *fully qualified domain name* (FQDN), consists of a hostname and a domain name (`example.com`). The latter also includes the *top level domain* or TLD (`com`).

TLD assignment has become quite confusing for historical reasons. Traditionally, three-letter domain names are used in the USA. In the rest of the world, the two-letter ISO national codes are the standard. In addition to that, longer TLDs were introduced in 2000 that represent certain spheres of activity (for example, `.info`, `.name`, `.museum`).

In the early days of the Internet (before 1990), the file `/etc/hosts` was used to store the names of all the machines represented over the Internet. This quickly proved to be impractical in the face of the rapidly growing number of computers connected to the Internet. For this reason, a decentralized database was developed to store the hostnames in a widely distributed manner. This database, similar to the name server, does not have the data pertaining to all hosts in the Internet readily available, but can dispatch requests to other name servers.

The top of the hierarchy is occupied by *root name servers*. These root name servers manage the top level domains and are run by the Network Information Center (NIC). Each root name server knows about the name servers responsible for a given top

level domain. Information about top level domain NICs is available at <http://www.internic.net>.

DNS can do more than just resolve hostnames. The name server also knows which host is receiving e-mails for an entire domain—the *mail exchanger (MX)*.

For your machine to resolve an IP address, it must know about at least one name server and its IP address. Easily specify such a name server with the help of YaST. If you have a modem dial-up connection, you may not need to configure a name server manually at all. The dial-up protocol provides the name server address as the connection is made. The configuration of name server access with openSUSE® is described in Section 13.4.1.4, “Configuring Hostname and DNS” (page 241). Setting up your own name server is described in Chapter 15, *The Domain Name System* (page 279).

The protocol `whois` is closely related to DNS. With this program, quickly find out who is responsible for any given domain.

NOTE: MDNS and .local Domain Names

The `.local` top level domain is treated as link-local domain by the resolver. DNS requests are sent as multicast DNS requests instead of normal DNS requests. If you already use the `.local` domain in your nameserver configuration, you must switch this option off in `/etc/host.conf`. For more information, see the `host.conf` manual page.

If you want to switch off MDNS during installation, use `nomdns=1` as a boot parameter.

For more information on multicast DNS, see <http://www.multicastdns.org>.

13.4 Configuring a Network Connection with YaST

There are many supported networking types on Linux. Most of them use different device names and the configuration files are spread over several locations in the file system. For a detailed overview of the aspects of manual network configuration, see Section 13.6, “Configuring a Network Connection Manually” (page 256).

During installation on a laptop (where NetworkManager is active by default) YaST configures all interfaces that have been detected. If NetworkManager is not active, only the first interface with link up (with a network cable connected) is automatically configured. Additional hardware can be configured any time on the installed system. The following sections describe the network configuration for all types of network connections supported by openSUSE.

13.4.1 Configuring the Network Card with YaST

To configure your wired or wireless network card in YaST, select *Network Devices > Network Settings*. After starting the module, YaST displays the *Network Settings* dialog with four tabs: *Global Options*, *Overview*, *Hostname/DNS* and *Routing*.

The *Global Options* tab allows you to set general networking options such as the use of NetworkManager, IPv6 and general DHCP options. For more information, see Section 13.4.1.1, “Configuring Global Networking Options” (page 234).

The *Overview* tab contains information about installed network interfaces and configurations. Any properly detected network card is listed with its name. You can manually configure new cards, remove or change their configuration in this dialog. If you want to manually configure a card that was not automatically detected, see Section 13.4.1.3, “Configuring an Undetected Network Card” (page 240). If you want to change the configuration of an already configured card, see Section 13.4.1.2, “Changing the Configuration of a Network Card” (page 235).

The *Hostname/DNS* tab allows to set the hostname of the machine and name the servers to be used. For more information, see Section 13.4.1.4, “Configuring Hostname and DNS” (page 241).

The *Routing* tab is used for the configuration of routing. See Section 13.4.1.5, “Configuring Routing” (page 243) for more information.

Figure 13.3: *Configuring Network Settings*



13.4.1.1 Configuring Global Networking Options

The *Global Options* tab of the YaST *Network Settings* module allows you to set important global networking options, such as the use of NetworkManager, IPv6 and DHCP client options. These settings are applicable for all network interfaces.

In the *Network Setup Method* choose the way network connections are managed. If you want a NetworkManager desktop applet to manage connections for all interfaces, choose *User Controlled with NetworkManager*. This option is well suited for switching between multiple wired and wireless networks. If you do not run a desktop environment (GNOME or KDE), or if your computer is a Xen server, virtual system, or provides network services such as DHCP or DNS in your network, use the *Traditional Method with ifup*. If NetworkManager is used, `nm-applet` should be used to configure network options and the *Overview*, *Hostname/DNS* and *Routing* tabs of the *Network Settings* module are disabled. For more information on NetworkManager, see Chapter 25, *Using NetworkManager* (page 435).

In the *IPv6 Protocol Settings* choose whether you want to use the IPv6 protocol. It is possible to use IPv6 together with IPv4. By default, IPv6 is activated. However, in

networks not using IPv6 protocol, response times can be faster with IPv6 protocol disabled. If you want to disable IPv6, uncheck the *Enable IPv6* option. This disables autoloading of the kernel module for IPv6. This will be applied after reboot.

In the *DHCP Client Options* configure options for the DHCP client. The *DHCP Client Identifier* must be different for each DHCP client on a single network. If left empty, it defaults to the hardware address of the network interface. However, if you are running several virtual machines using the same network interface and, therefore, the same hardware address, specify a unique free-form identifier here.

The *Hostname to Send* specifies a string used for the hostname option field when *dhcpcd* sends messages to DHCP server. Some DHCP servers update name server zones (forward and reverse records) according to this hostname (Dynamic DNS). Also, some DHCP servers require the *Hostname to Send* option field to contain a specific string in the DHCP messages from clients. Leave *AUTO* to send the current hostname (that is the one defined in */etc/HOSTNAME*). Leave the option field empty for not sending any hostname. If you do not want to change the default route according to the information from DHCP, uncheck *Change Default Route via DHCP*.

13.4.1.2 Changing the Configuration of a Network Card

To change the configuration of a network card, select a card from the list of the detected cards in *Network Settings > Overview* in YaST and click *Edit*. The *Network Card Setup* dialog appears in which to adjust the card configuration using the *General*, *Address* and *Hardware* tabs. For information about wireless card configuration, see Section 24.5, “Configuration with YaST” (page 421).

Configuring IP Addresses

You can set the IP address of the network card or the way its IP address is determined in the *Address* tab of the *Network Card Setup* dialog. Both IPv4 and IPv6 addresses are supported. The network card can have *No IP Address* (which is useful for bonding devices), a *Statically Assigned IP Address* (IPv4 or IPv6) or a *Dynamic Address* assigned via *DHCP* or *Zeroconf* or both.

If using *Dynamic Address*, select whether to use *DHCP Version 4 Only* (for IPv4), *DHCP Version 6 Only* (for IPv6) or *DHCP Both Version 4 and 6*.

If possible, the first network card with link that is available during the installation is automatically configured to use automatic address setup via DHCP. In case of laptop

computers where NetworkManager is active by default, all network cards are configured.

DHCP should also be used if you are using a DSL line but with no static IP assigned by the ISP (Internet Service Provider). If you decide to use DHCP, configure the details in *DHCP Client Options* in the *Global Options* tab of the *Network Settings* dialog of the YaST network card configuration module. If you have a virtual host setup where different hosts communicate through the same interface, an *DHCP Client Identifier* is necessary to distinguish them.

DHCP is a good choice for client configuration but it is not ideal for server configuration. To set a static IP address, proceed as follows:

- 1 Select a card from the list of detected cards in the *Overview* tab of the YaST network card configuration module and click *Edit*.
- 2 In the *Address* tab, choose *Statically Assigned IP Address*.
- 3 Enter the *IP Address*. Both IPv4 and IPv6 addresses can be used. Enter the network mask in *Subnet Mask*. If the IPv6 address is used, use *Subnet Mask* for prefix length in format / 64.

Optionally, you can enter a fully qualified *Hostname* for this address, which will be written to the `/etc/hosts` configuration file.

- 4 Click *Next*.
- 5 To activate the configuration, click *OK*.

If you use the static address, the name servers and default gateway are not configured automatically. To configure name servers, proceed as described in Section 13.4.1.4, “Configuring Hostname and DNS” (page 241). To configure a gateway, proceed as described in Section 13.4.1.5, “Configuring Routing” (page 243).

Configuring Aliases

One network device can have multiple IP addresses, called aliases.

NOTE: Aliases Are a Compatibility Feature

These so-called aliases resp. labels work with IPv4 only. With IPv6 they will be ignored. Using `iproute2` network interfaces can have one or more addresses.

Using YaST to set an alias for your network card, proceed as follows:

- 1 Select a card from the list of detected cards in the *Overview* tab of the YaST network card configuration module and click *Edit*.
- 2 In the *Address > Additional Addresses* tab, click *Add*.
- 3 Enter *Alias Name*, *IP Address*, and *Netmask*. Do not include the interface name in the alias name.
- 4 Click *OK*.
- 5 Click *Next*.
- 6 To activate the configuration, click *OK*.

Changing the Device Name and Udev Rules

It is possible to change the device name of the network card when it is used. It is also possible to determine whether the network card should be identified by udev via its hardware (MAC) address or via the bus ID. The later option is preferable in large servers to ease hot swapping of cards. To set these options with YaST, proceed as follows:

- 1 Select a card from the list of detected cards in the *Overview* tab of the YaST *Network Settings* module and click *Edit*.
- 2 Go to the *Hardware* tab. The current device name is shown in *Udev Rules*. Click *Change*.
- 3 Select whether udev should identify the card by its *MAC Address* or *Bus ID*. The current MAC address and bus ID of the card are shown in the dialog.
- 4 To change the device name, check the *Change Device Name* option and edit the name.

5 Click *OK* and *Next*.

6 To activate the configuration, click *OK*.

Changing Network Card Kernel Driver

For some network cards, several kernel drivers may be available. If the card is already configured, YaST allows you to select a kernel driver to be used from a list of available suitable drivers. It is also possible to specify options for the kernel driver. To set these options with YaST, proceed as follows:

- 1 Select a card from the list of detected cards in the *Overview* tab of the YaST Network Settings module and click *Edit*.
- 2 Go to the *Hardware* tab.
- 3 Select the kernel driver to be used in *Module Name*. Enter any options for the selected driver in *Options* in the form `option=value` . If more options are used, they should be space-separated.
- 4 Click *OK* and *Next*.
- 5 To activate the configuration, click *OK*.

Activating the Network Device

If you use the traditional method with ifup, you can configure your device to either start during boot, on cable connection, on card detection, manually or never. To change device start-up, proceed as follows:

- 1 In YaST select a card from the list of detected cards in *Network Devices > Network Settings* and click *Edit*.
- 2 In the *General* tab, select the desired entry from *Device Activation*.

Choose *At Boot Time* to start the device during the system boot. With *On Cable Connection*, the interface is watched for any existing physical connection. With *On Hotplug*, the interface is set as soon as available. It is similar to the *At Boot Time* option, and only differs in the fact that no error occurs if the interface is not present at boot time. Choose *Manually* to control the interface manually with

ifup. Choose *Never* to not start the device at all. The *On NFSroot* is similar to *At Boot Time*, but the interface does not shut down with the `rcnetwork stop` command. Use this if you use an nfs or iscsi root file system.

3 Click *Next*.

4 To activate the configuration, click *OK*.

Usually, only the system administrator can activate and deactivate network interfaces. If you want any user to be able to activate this interface via KInternet, select *Enable Device Control for Non-root User via KInternet*.

Setting Up Maximum Transfer Unit Size

You can set a maximum transmission unit (MTU) for the interface. MTU refers to the largest allowed packet size in bytes. A higher MTU brings higher bandwidth efficiency. However, large packets can block up a slow interface for some time, increasing the lag for further packets.

1 In YaST select a card from the list of detected cards in *Network Devices > Network Settings* and click *Edit*.

2 In the *General* tab, select the desired entry from the *Set MTU* list.

3 Click *Next*.

4 To activate the configuration, click *OK*.

Configuring the Firewall

Without having to enter the detailed firewall setup as described in Section “Configuring the Firewall with YaST” (Chapter 13, *Masquerading and Firewalls*, ↑Security Guide), you can determine the basic firewall setup for your device as part of the device setup. Proceed as follows:

1 Open the YaST *Network Devices > Network Settings* module. In the *Overview* tab, select a card from the list of detected cards and click *Edit*.

2 Enter the *General* tab of the *Network Settings* dialog.

- 3 Determine the firewall zone to which your interface should be assigned. The following options are available:

Firewall Disabled

This option is available only if the firewall is disabled and the firewall does not run at all. Only use this option if your machine is part of a greater network that is protected by an outer firewall.

Automatically Assign Zone

This option is available only if the firewall is enabled. The firewall is running and the interface is automatically assigned to a firewall zone. The zone which contains the keyword `any` or the external zone will be used for such an interface.

Internal Zone (Unprotected)

The firewall is running, but does not enforce any rules to protect this interface. Use this option if your machine is part of a greater network that is protected by an outer firewall. It is also useful for the interfaces connected to the internal network, when the machine has more network interfaces.

Demilitarized Zone

A demilitarized zone is an additional line of defense in front of an internal network and the (hostile) Internet. Hosts assigned to this zone can be reached from the internal network and from the Internet, but cannot access the internal network.

External Zone

The firewall is running on this interface and fully protects it against other—presumably hostile—network traffic. This is the default option.

- 4 Click *Next*.
- 5 Activate the configuration by clicking *OK*.

13.4.1.3 Configuring an Undetected Network Card

Your card may not be detected correctly. In this case, the card is not included in the list of detected cards. If you are sure that your system includes a driver for your card, you can configure it manually. You can also configure special network device types, such as bridge, bond, TUN or TAP. To configure an undetected network card (or a special device) proceed as follows:

- 1 In the *Network Devices > Network Settings > Overview* dialog in YaST click *Add*.
- 2 In the *Hardware* dialog, set the *Device Type* of the interface from the available options and *Configuration Name*. If the network card is a PCMCIA or USB device, activate the respective check box and exit this dialog with *Next*. Otherwise, you can define the kernel *Module Name* to be used for the card and its *Options*, if necessary.

In *Ethtool Options*, you can set `ethtool` options used by `ifup` for the interface. See the `ethtool` manual page for available options. If the option string starts with a `-` (for example `-K interface_name rx on`), the second word in the string is replaced with the current interface name. Otherwise (for example `autoneg off speed 10`) `ifup` prepends `-s interface_name`.

- 3 Click *Next*.
- 4 Configure any needed options, such as the IP address, device activation or firewall zone for the interface in the *General*, *Address*, and *Hardware* tabs. For more information about the configuration options, see Section 13.4.1.2, “Changing the Configuration of a Network Card” (page 235).
- 5 If you selected *Wireless* as the device type of the interface, configure the wireless connection in the next dialog.
- 6 Click *Next*.
- 7 To activate the new network configuration, click *OK*.

13.4.1.4 Configuring Hostname and DNS

If you did not change the network configuration during installation and the wired card was already available, a hostname was automatically generated for your computer and DHCP was activated. The same applies to the name service information your host needs to integrate into a network environment. If DHCP is used for network address setup, the list of domain name servers is automatically filled with the appropriate data. If a static setup is preferred, set these values manually.

To change the name of your computer and adjust the name server search list, proceed as follows:

- 1 Go to the *Network Settings > Hostname/DNS* tab in the *Network Devices* module in YaST.

- 2 Enter the *Hostname* and, if needed, the *Domain Name*. The domain is especially important if the machine is a mail server. Note that the hostname is global and applies to all set network interfaces.

If you are using DHCP to get an IP address, the hostname of your computer will be automatically set by the DHCP. You may want to disable this behavior if you connect to different networks, because they may assign different hostnames and changing the hostname at runtime may confuse the graphical desktop. To disable using DHCP to get an IP address uncheck *Change Hostname via DHCP*.

Assign Hostname to Loopback IP associates your hostname with 127.0.0.2 (loopback) IP address in `/etc/hosts`. This is an useful option if you want to have the hostname resolvable at all times, even without active network.

- 3 In *Modify DNS Configuration*, select the way the DNS configuration (name servers, search list, the content of the `/etc/resolv.conf` file) is modified.

If the *Use Default Policy* option is selected, the configuration is handled by the `netconfig` script which merges the data defined statically (with YaST or in the configuration files) with data obtained dynamically (from the DHCP client or NetworkManager). This default policy is sufficient in most cases.

If the *Only Manually* option is selected, `netconfig` is not allowed to modify the `/etc/resolv.conf` file. However, this file can be edited manually.

If the *Custom Policy* option is selected, a *Custom Policy Rule* string defining the merge policy should be specified. The string consists of a comma-separated list of interface names to be considered a valid source of settings. Except for complete interface names, basic wildcards to match multiple interfaces are allowed, as well. For example, `eth* ppp?` will first target all `eth` and then all `ppp0-ppp9` interfaces. There are two special policy values that indicate how to apply the static settings defined in the `/etc/sysconfig/network/config` file:

`STATIC`

The static settings have to be merged together with the dynamic settings.

`STATIC_FALLBACK`

The static settings are used only when no dynamic configuration is available.

For more information, see the `man 8 netconfig`.

- 4 Enter the *Name Servers* and fill in the *Domain Search* list. Name servers must be specified by IP addresses, such as 192.168.1.116, not by hostnames. Names specified in the *Domain Search* tab are domain names used for resolving hostnames without a specified domain. If more than one *Domain Search* is used, separate domains with commas or white space.
- 5 To activate the configuration, click *OK*.

13.4.1.5 Configuring Routing

To make your machine communicate with other machines and other networks, routing information must be given to make network traffic take the correct path. If DHCP is used, this information is automatically provided. If a static setup is used, this data must be added manually.

- 1 In YaST go to *Network Settings > Routing*.
- 2 Enter the IP address of the *Default Gateway* (IPv4 and IPv6 if necessary). The default gateway matches every possible destination, but if any other entry exists that matches the required address, use this instead of the default route.
- 3 More entries can be entered in the *Routing Table*. Enter the *Destination* network IP address, *Gateway* IP address and the *Netmask*. Select the *Device* through which the traffic to the defined network will be routed (the minus sign stands for any device). To omit any of these values, use the minus sign -. To enter a default gateway into the table, use `default` in the *Destination* field.

NOTE

If more default routes are used, it is possible to specify the metric option to determine which route has a higher priority. To specify the metric option, enter `- metric number` in *Options*. The route with the highest metric is used as default. If the network device is disconnected, its route will be removed and the next one will be used. However, the current kernel does not use metric in static routing, only routing daemons like multipathd do.

- 4 If the system is a router, enable the *IP Forwarding* option in the *Network Settings*.
- 5 To activate the configuration, click *OK*.


13.4.2 Modem

In the YaST Control Center, access the modem configuration under *Network Devices* > *Modem*. If your modem was not automatically detected, go to the *Modem Devices* tab and open the dialog for manual configuration by clicking *Add*. Enter the interface to which the modem is connected under *Modem Device*.

TIP: CDMA and GPRS Modems

Configure supported CDMA and GPRS modems with the YaST *Modem* module just as you would configure regular modems.

Figure 13.4: *Modem Configuration*

 **Modem Parameters**
Enter all modem configuration values. [more](#)

Modem Device:

Dial Prefix (if needed)

Dial Mode	Special Settings
<input checked="" type="radio"/> Tone Dialing	<input checked="" type="checkbox"/> Speaker On
<input type="radio"/> Pulse Dialing	<input checked="" type="checkbox"/> Detect Dial Tone

If you are behind a private branch exchange (PBX), you may need to enter a dial prefix. This is often a zero. Consult the instructions that came with the PBX to find out. Also select whether to use tone or pulse dialing, whether the speaker should be on and whether the modem should wait until it detects a dial tone. The last option should not be enabled if the modem is connected to an exchange.

Under *Details*, set the baud rate and the modem initialization strings. Only change these settings if your modem was not detected automatically or if it requires special settings for data transmission to work. This is mainly the case with ISDN terminal adapters. Leave this dialog by clicking *OK*. To delegate control over the modem to the normal user without root permissions, activate *Enable Device Control for Non-root User via KInternet*. In this way, a user without administrator permissions can activate or deactivate an interface. Under *Dial Prefix Regular Expression*, specify a regular expression. The *Dial Prefix* in KInternet, which can be modified by the normal user, must match this regular expression. If this field is left empty, the user cannot set a different *Dial Prefix* without administrator permissions.

In the next dialog, select the ISP. To choose from a predefined list of ISPs operating in your country, select *Country*. Alternatively, click *New* to open a dialog in which to provide the data for your ISP. This includes a name for the dial-up connection and ISP as well as the login and password provided by your ISP. Enable *Always Ask for Password* to be prompted for the password each time you connect.

In the last dialog, specify additional connection options:

Dial on Demand

If you enable *Dial on Demand*, set at least one name server. Use this feature only if your Internet connection is inexpensive, because there are programs that periodically request data from the Internet.

Modify DNS when Connected

This option is enabled by default, with the effect that the name server address is updated each time you connect to the Internet.

Automatically Retrieve DNS

If the provider does not transmit its domain name server after connecting, disable this option and enter the DNS data manually.

Automatically Reconnect

If this options is enabled, the connection is automatically reestablished after failure.

Ignore Prompts

This option disables the detection of any prompts from the dial-up server. If the connection build-up is slow or does not work at all, try this option.

External Firewall Interface

Selecting this option activates the firewall and sets the interface as external. This way, you are protected from outside attacks for the duration of your Internet connection.

Idle Time-Out (seconds)

With this option, specify a period of network inactivity after which the modem disconnects automatically.

IP Details

This opens the address configuration dialog. If your ISP does not assign a dynamic IP address to your host, disable *Dynamic IP Address* then enter your host's local IP address and the remote IP address. Ask your ISP for this information. Leave *Default Route* enabled and close the dialog by selecting *OK*.

Selecting *Next* returns to the original dialog, which displays a summary of the modem configuration. Close this dialog with *OK*.

13.4.3 ISDN

Use this module to configure one or several ISDN cards for your system. If YaST did not detect your ISDN card, click on *Add* in the *ISDN Devices* tab and manually select your card. Multiple interfaces are possible, but several ISPs can be configured for one interface. In the subsequent dialogs, set the ISDN options necessary for the proper functioning of the card.

Figure 13.5: *ISDN Configuration*

ISDN Low-Level Configuration for contr0
With OnBoot, the driver is loaded during system boot. [more](#)

ISDN Card Information

Vendor	Abocom/Magitek
ISDN Card	2BD1

Driver: HiSax driver

ISDN Protocol

☒ Euro-ISDN (EDSSI)
☐ ITR6
☐ Leased Line
☐ NI1

Country: Germany Code: +49

Area Code: Dial Prefix:

☒ Start ISDN Log

Activate device: At Boot Time

Help Cancel Back OK

In the next dialog, shown in Figure 13.5, “ISDN Configuration” (page 247), select the protocol to use. The default is *Euro-ISDN (EDSSI)*, but for older or larger exchanges, select *ITR6*. 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. If you do not want to log all your ISDN traffic, uncheck the *Start ISDN Log* option.

Activate Device defines how the ISDN interface should be started: *At Boot Time* causes the ISDN driver to be initialized each time the system boots. *Manually* requires you to load the ISDN driver as `root` with the command `rcisdn start`. *On Hot-plug*, used for PCMCIA or USB devices, loads the driver after the device is plugged in. When finished with these settings, select *OK*.

In the next dialog, specify the interface type for your ISDN card and add ISPs to an existing interface. Interfaces may be either the `SyncPPP` or the `RawIP` type, but most ISPs operate in the `SyncPPP` mode, which is described below.

Figure 13.6: *ISDN Interface Configuration*

The number to enter for *My Phone Number* depends on your particular setup:

ISDN Card Directly Connected to Phone Outlet

A standard ISDN line provides three phone numbers (called multiple subscriber numbers, or MSNs). If the subscriber asked for more, there may be up to 10. One of these MSNs must be entered here, but without your area code. If you enter the wrong number, your phone operator automatically falls back to the first MSN assigned to your ISDN line.

ISDN Card Connected to a Private Branch Exchange

Again, the configuration may vary depending on the equipment installed:

1. Smaller private branch exchanges (PBX) built for home purposes mostly use the Euro-ISDN (EDSS1) protocol for internal calls. These exchanges have an internal S0 bus and use internal numbers for the equipment connected to them.

Use one of the internal numbers as your MSN. You should be able to use at least one of the exchange's MSNs that have been enabled for direct outward dialing. If this does not work, try a single zero. For further information, consult the documentation delivered with your phone exchange.

2. Larger phone exchanges designed for businesses normally use the ITR6 protocol for internal calls. Their MSN is called EAZ and usually corresponds to the direct-dial number. For the configuration under Linux, it should be sufficient to enter the last digit of the EAZ. As a last resort, try each of the digits from 1 to 9.

For the connection to be terminated just before the next charge unit is due, enable *ChargeHUP*. However, remember that may not work with every ISP. You can also enable channel bundling (multilink PPP) by selecting the corresponding option. Finally, you can enable the firewall for your link by selecting *External Firewall Interface* and *Restart Firewall*. To enable the normal user without administrator permissions to activate or deactivate the interface, select the *Enable Device Control for Non-root User via KInternet*.

Details opens a dialog in which to implement more complex connection schemes which are not relevant for normal home users. Leave the *Details* dialog by selecting *OK*.

In the next dialog, configure IP address settings. If you have not been given a static IP by your provider, select *Dynamic IP Address*. Otherwise, use the fields provided to enter your host's local IP address and the remote IP address according to the specifications of your ISP. If the interface should be the default route to the Internet, select *Default Route*. Each host can only have one interface configured as the default route. Leave this dialog by selecting *Next*.

The following dialog allows you to set your country and select an ISP. The ISPs included in the list are call-by-call providers only. If your ISP is not in the list, select *New*. This opens the *Provider Parameters* dialog in which to enter all the details for your ISP. When entering the phone number, do not include any blanks or commas among the digits. Finally, enter your login and the password as provided by the ISP. When finished, select *Next*.

To use *Dial on Demand* on a stand-alone workstation, specify the name server (DNS server) as well. Most ISPs support dynamic DNS, which means the IP address of a name server is sent by the ISP each time you connect. For a single workstation, however, you still need to provide a placeholder address like 192.168.22.99. If your ISP does not support dynamic DNS, specify the name server IP addresses of the ISP. If desired, specify a time-out for the connection—the period of network inactivity (in seconds) after which the connection should be automatically terminated. Confirm your settings with *Next*. YaST displays a summary of the configured interfaces. To activate these settings, select *OK*.

13.4.4 Cable Modem

In some countries it is quite common to access the Internet through the TV cable network. The TV cable subscriber usually gets a modem that is connected to the TV cable outlet on one side and to a computer network card on the other (using a 10Base-TG twisted pair cable). The cable modem then provides a dedicated Internet connection with a fixed IP address.

Depending on the instructions provided by your ISP, when configuring the network card either select *Dynamic Address* or *Statically Assigned IP Address*. Most providers today use DHCP. A static IP address often comes as part of a special business account.

13.4.5 DSL

To configure your DSL device, select the *DSL* module from the YaST *Network Devices* section. This YaST module consists of several dialogs in which to set the parameters of DSL links based on one of the following protocols:

- PPP over Ethernet (PPPoE)
- PPP over ATM (PPPoATM)
- CAPI for ADSL (Fritz Cards)
- Point-to-Point Tunneling Protocol (PPTP)—Austria

In the *DSL Devices* tab of the *DSL Configuration Overview* dialog, you will find a list of installed DSL devices. To change the configuration of a DSL device, select it in the list and click *Edit*. If you click *Add*, you can manually configure a new DSL device.

The configuration of a DSL connection based on PPPoE or PPTP requires that the corresponding network card be set up in the correct way. If you have not done so yet, first configure the card by selecting *Configure Network Cards* (see Section 13.4.1, “Configuring the Network Card with YaST” (page 233)). In the case of a DSL link, addresses may be assigned automatically but not via DHCP, which is why you should not enable the option *Dynamic Address*. Instead, enter a static dummy address for the interface, such as 192.168.22.1. In *Subnet Mask*, enter 255.255.255.0. If you are configuring a stand-alone workstation, leave *Default Gateway* empty.


TIP

Values in *IP Address* and *Subnet Mask* are only placeholders. They are only needed to initialize the network card and do not represent the DSL link as such.

In the first DSL configuration dialog (see Figure 13.7, “DSL Configuration” (page 252)), select the *PPP Mode* and the *Ethernet Card* to which the DSL modem is connected (in most cases, this is `eth0`). Then use *Activate Device* to specify whether the DSL link should be established during the boot process. Click *Enable Device Control for Non-root User via KInternet* to authorize the normal user without root permissions to activate or deactivate the interface with KInternet.

In the next dialog select your country and choose from a number of ISPs operating in it. The details of any subsequent dialogs of the DSL configuration depend on the options set so far, which is why they are only briefly mentioned in the following paragraphs. For details on the available options, read the detailed help available from the dialogs.

Figure 13.7: DSL Configuration

 **DSL Configuration**
Here, set the most important settings for the DSL connection. [more](#)

DSL Connection Settings

PPP Mode:
PPP over Ethernet

PPP Mode-Dependent Settings

VPI/VCI:

Ethernet Card

82566DM Gigabit Network Connection
Network Card - DHCP address

Server Name or IP Address:

Activate device:
Manually

☒ Enable Device Control for Non-root User Via KInetnet

To use *Dial on Demand* on a stand-alone workstation, also specify the name server (DNS server). Most ISPs support dynamic DNS—the IP address of a name server is sent by the ISP each time you connect. For a single workstation, however, provide a placeholder address like 192.168.22.99. If your ISP does not support dynamic DNS, enter the name server IP address provided by your ISP.

Idle Time-Out (seconds) defines a period of network inactivity after which to terminate the connection automatically. A reasonable time-out value is between 60 and 300 seconds. If *Dial on Demand* is disabled, it may be useful to set the time-out to zero to prevent automatic hang-up.

The configuration of T-DSL is very similar to the DSL setup. Just select *T-Online* as your provider and YaST opens the T-DSL configuration dialog. In this dialog, provide some additional information required for T-DSL—the line ID, the T-Online number, the user code and your password. All of these should be included in the information you received after subscribing to T-DSL.

13.5 NetworkManager

NetworkManager is the ideal solution for laptops and other portable computers. With NetworkManager, you do not need to worry about configuring network interfaces and switching between networks when you are moving.

13.5.1 NetworkManager and ifup

However, NetworkManager is not a suitable solution for all cases, so you can still choose between the traditional method for managing network connections (ifup) and NetworkManager. If you want to manage your network connection with NetworkManager, enable NetworkManager in the YaST Network Settings module as described in Section 25.2, “Enabling or Disabling NetworkManager” (page 436) and configure your network connections with NetworkManager. For a list of use cases and a detailed description how to configure and use NetworkManager, refer to Chapter 25, *Using NetworkManager* (page 435).

Some differences between ifup and NetworkManager include:

`root` Privileges

If you use NetworkManager for network setup, you can easily switch, stop or start your network connection at any time from an icon within your desktop environment. NetworkManager also makes it possible to change and configure wireless card connections without requiring `root` privileges. For this reason, NetworkManager is the ideal solution for a mobile workstation.

Traditional configuration with ifup also provides some ways to switch, stop or start the connection with or without user intervention, like user-managed devices. However, this always requires `root` privileges to change or configure a network device. This is often a problem for mobile computing, where it is not possible to preconfigure all the connection possibilities.

Types of Network Connections

Both traditional configuration and NetworkManager can handle network connections with a wireless network (with WEP, WPA-PSK, and WPA-Enterprise access) and wired networks using DHCP and static configuration. They also support connection through dial-up, DSL and VPN. With NetworkManager you can also connect a mobile broadband (3G) modem, which is not possible with the traditional configuration.

NetworkManager tries to keep your computer connected at all times using the best connection available. If the network cable is accidentally disconnected, it tries to reconnect. It can find the network with the best signal strength from the list of your wireless connections and automatically use it to connect. To get the same functionality with `ifup`, a great deal of configuration effort is required.

13.5.2 NetworkManager Functionality and Configuration Files

The individual network connection settings created with NetworkManager are stored in configuration profiles. The *system* connections configured with either NetworkManager or YaST are saved in `/etc/networkmanager/system-connections/*` or in `/etc/sysconfig/network/ifcfg-*`. Any user-defined connections are stored in GConf for GNOME.

In case no profile is configured, NetworkManager automatically creates one and names it `Auto $INTERFACE-NAME`. That is made in an attempt to work without any configuration for as many cases as (securely) possible. If the automatically created profiles do not suit your needs, use the network connection configuration dialogs provided by KDE or GNOME to modify them as desired. For more information, refer to Section 25.3, “Configuring Network Connections” (page 437).

13.5.3 Controlling and Locking Down NetworkManager Features

On centrally administered machines, certain NetworkManager features can be controlled or disabled with PolicyKit, for example if a user is allowed to modify administrator defined connections or if a user is allowed to define his own network configurations. To view or change the respective NetworkManager policies, start the graphical *Authorizations* tool for PolicyKit. In the tree on the left side, find them below the *network-manager-settings* entry.

The following table gives an overview of the PolicyKit identifiers related to NetworkManager:

Table 13.5: *PolicyKit Identifiers for NetworkManager*

Identifier	Description
org.freedesktop.NetworkManager.enable-disable-network	Enable or disable system networking.
org.freedesktop.NetworkManager.sleep-wake	Put NetworkManager to sleep or wake it up.
org.freedesktop.NetworkManager.enable-disable-wwan	Enable or disable mobile broadband devices.
org.freedesktop.NetworkManager.enable-disable-wimax	Enable or disable WiMAX mobile broadband devices.
org.freedesktop.NetworkManager.network-control	Allow control of network connections.
org.freedesktop.NetworkManager.enable-disable-wifi	Enable or disable WiFi devices.
org.freedesktop.NetworkManager.settings.modify.hosts	Modify persistent system hostname.
org.freedesktop.network-manager-settings.modify.system	Modify network connections for all users.
org.freedesktop.network-manager-settings.modify.own	Modify personal network connections.
org.freedesktop.network-manager-settings.system.wifi.share.open	Connection sharing via an open WiFi network.
org.freedesktop.network-manager-settings.system.wifi.share.protected	Connection sharing via a protected WiFi network.

13.6 Configuring a Network Connection Manually

Manual configuration of the network software should always be the last alternative. Using YaST is recommended. However, this background information about the network configuration can also assist your work with YaST.

When the Kernel detects a network card and creates a corresponding network interface, it assigns the device a name depending on the order of device discovery, or order of the loading of the Kernel modules. The default Kernel device names are only predictable in very simple or tightly controlled hardware environments. Systems which allow adding or removing hardware during runtime or support automatic configuration of devices cannot expect stable network device names assigned by the Kernel across reboots.

However, all system configuration tools rely on persistent interface names. This problem is solved by udev. The udev persistent net generator (`/lib/udev/rules.d/75-persistent-net-generator.rules`) generates a rule matching the hardware (using its hardware address by default) and assigns a persistently unique interface for the hardware. The udev database of network interfaces is stored in the file `/etc/udev/rules.d/70-persistent-net.rules`. Every line in the file describes one network interface and specifies its persistent name. System administrators can change the assigned names by editing the `NAME=""` entries. The persistent rules can also be modified using YaST.

Table 13.6, “Manual Network Configuration Scripts” (page 256) summarizes the most important scripts involved in the network configuration.

Table 13.6: *Manual Network Configuration Scripts*

Command	Function
<code>ifup, ifdown, ifstatus</code>	The <code>if</code> scripts start or stop network interfaces, or return the status of the specified interface. For more information, see the <code>ifup</code> manual page.
<code>rcnetwork</code>	The <code>rcnetwork</code> script can be used to start, stop or restart all network in-

Command	Function
	<p>interfaces (or just a specified one). Use <code>rcnetwork stop</code> to stop, <code>rcnetwork start</code> to start and <code>rcnetwork restart</code> to restart network interfaces. If you want to stop, start or restart just one interface, use the command followed by the interface name, for example <code>rcnetwork restart eth0</code>. The <code>rcnetwork status</code> command displays the state of the interfaces, their IP addresses and whether a DHCP client is running. With <code>rcnetwork stop-all-dhcp-clients</code> and <code>rcnetwork restart-all-dhcp-clients</code> you can stop or restart DHCP clients running on network interfaces.</p>

For more information about udev and persistent device names, see Chapter 12, *Dynamic Kernel Device Management with udev* (page 199).

13.6.1 Configuration Files

This section provides an overview of the network configuration files and explains their purpose and the format used.

13.6.1.1 /etc/sysconfig/network/ifcfg-*

These files contain the configurations for network interfaces. They include information such as the start mode and the IP address. Possible parameters are described in the manual page of `ifup`. Additionally, most variables from the `dhcp` and `wireless` files can be used in the `ifcfg-*` files if a general setting should be used for only one interface. However, most of the `/etc/sysconfig/network/config` variables are global and cannot be overridden in `ifcfg`-files. For example `NETWORKMANAGER` or `NETCONFIG_*` variables are global.

For `ifcfg.template`, see Section 13.6.1.2, “`/etc/sysconfig/network/config`, `/etc/sysconfig/network/dhcp`, and `/etc/sysconfig/network/wireless`” (page 258).

13.6.1.2 `/etc/sysconfig/network/config`, `/etc/sysconfig/network/dhcp`, and `/etc/sysconfig/network/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. Some of the variables from `/etc/sysconfig/network/config` can also be used in `ifcfg-*` files, where they are given a higher priority. The `/etc/sysconfig/network/ifcfg.template` file lists variables that can be specified in a per interface scope. However, most of the `/etc/sysconfig/network/config` variables are global and cannot be overridden in `ifcfg`-files. For example, `NETWORKMANAGER` or `NETCONFIG_*` variables are global.

13.6.1.3 `/etc/sysconfig/network/routes` and `/etc/sysconfig/network/ifroute-*`

The static routing of TCP/IP packets is determined here. All the static routes required by the various system tasks can be entered in the `/etc/sysconfig/network/routes` file: routes to a host, routes to a host via a gateway and routes to a network. For each interface that needs individual routing, define an additional configuration file: `/etc/sysconfig/network/ifroute-*`. Replace `*` with the name of the interface. The entries in the routing configuration files look like this:

# Destination	Dummy/Gateway	Netmask	Device
#			
127.0.0.0	0.0.0.0	255.255.255.0	lo
204.127.235.0	0.0.0.0	255.255.255.0	eth0
default	204.127.235.41	0.0.0.0	eth0
207.68.156.51	207.68.145.45	255.255.255.255	eth1
192.168.0.0	207.68.156.51	255.255.0.0	eth1

The route's destination is in the first column. This column may contain the IP address of a network or host or, in the case of *reachable* name servers, the fully qualified network or hostname.

The second column contains the default gateway or a gateway through which a host or network can be accessed. The third column contains the netmask for networks or

hosts behind a gateway. For example, the mask is 255.255.255.255 for a host behind a gateway.

The fourth column is only relevant for networks connected to the local host such as loopback, Ethernet, ISDN, PPP and dummy device. The device name must be entered here.

An (optional) fifth column can be used to specify the type of a route. Columns that are not needed should contain a minus sign – to ensure that the parser correctly interprets the command. For details, refer to the `routes(5)` man page.

The unified format for IPv4 and IPv6 now looks as follows:

```
prefix/length      gateway -           [interface]
```

And the so-called compatibility format looks accordingly:

```
prefix            gateway length      [interface]
```

For IPv4 you still can use the old format with netmask:

```
ipv4-network      gateway ipv4-netmask [interface]
```

The following examples are equivalent:

2001:db8:abba:cafe::/64	2001:db8:abba:cafe::dead	-	eth0
208.77.188.0/24	208.77.188.166	-	eth0
2001:db8:abba:cafe::	2001:db8:abba:cafe::dead	64	eth0
208.77.188.0	208.77.188.166	24	eth0
208.77.188.0	208.77.188.166	255.255.255.0	eth0

13.6.1.4 /etc/resolv.conf

The domain to which the host belongs is specified in this file (keyword search). Also listed is the status of the name server address to access (keyword `nameserver`). Multiple domain names can be specified in the file. When resolving a name that is not fully qualified, an attempt is made to generate one by attaching the individual search entries. Multiple name servers can be specified in multiple lines, each beginning with `nameserver`. Comments are preceded by `#` signs. Example 13.5, “`/etc/resolv.conf`” (page 260) shows what `/etc/resolv.conf` could look like.

However, the `/etc/resolv.conf` should not be edited by hand. Instead, it is generated by the `netconfig` script. To define static DNS configuration without us-

ing YaST, edit the appropriate variables manually in the `/etc/sysconfig/network/config` file:

`NETCONFIG_DNS_STATIC_SEARCHLIST`

list of DNS domain names used for hostname lookup

`NETCONFIG_DNS_STATIC_SERVERS`

list of name server IP addresses to use for hostname lookup

`NETCONFIG_DNS_FORWARDER`

defines the name of the DNS forwarder that has to be configured

To disable DNS configuration using `netconfig`, set `NETCONFIG_DNS_POLICY=''`. For more information about `netconfig`, see `man 8 netconfig`.

Example 13.5: */etc/resolv.conf*

```
# Our domain
search example.com
#
# We use dns.example.com (192.168.1.116) as nameserver
nameserver 192.168.1.116
```

13.6.1.5 /sbin/netconfig

`netconfig` is a modular tool to manage additional network configuration settings. It merges statically defined settings with settings provided by autoconfiguration mechanisms as DHCP or PPP according to a predefined policy. The required changes are applied to the system by calling the `netconfig` modules that are responsible for modifying a configuration file and restarting a service or a similar action.

`netconfig` recognizes three main actions. The `netconfig modify` and `netconfig remove` commands are used by daemons such as DHCP or PPP to provide or remove settings to `netconfig`. Only the `netconfig update` command is available for the user:

`modify`

The `netconfig modify` command modifies the current interface and service specific dynamic settings and updates the network configuration. `Netconfig` reads settings from standard input or from a file specified with the `--lease-file filename` option and internally stores them until a system reboot (or the next `modify` or `remove` action). Already existing settings for the same interface and service combination are overwritten. The interface is specified by

the `-i interface_name` parameter. The service is specified by the `-s service_name` parameter.

remove

The `netconfig remove` command removes the dynamic settings provided by a modificatory action for the specified interface and service combination and updates the network configuration. The interface is specified by the `-i interface_name` parameter. The service is specified by the `-s service_name` parameter.

update

The `netconfig update` command updates the network configuration using current settings. This is useful when the policy or the static configuration has changed. Use the `-m module_type` parameter, if you want to update a specified service only (dns, nis, or ntp).

The `netconfig` policy and the static configuration settings are defined either manually or using YaST in the `/etc/sysconfig/network/config` file. The dynamic configuration settings provided by autoconfiguration tools as DHCP or PPP are delivered directly by these tools with the `netconfig modify` and `netconfig remove` actions. NetworkManager also uses `netconfig modify` and `netconfig remove` actions. When NetworkManager is enabled, `netconfig` (in policy mode `auto`) uses only NetworkManager settings, ignoring settings from any other interfaces configured using the traditional `ifup` method. If NetworkManager does not provide any setting, static settings are used as a fallback. A mixed usage of NetworkManager and the traditional `ifup` method is not supported.

For more information about `netconfig`, see `man 8 netconfig`.

13.6.1.6 /etc/hosts

In this file, shown in Example 13.6, “`/etc/hosts`” (page 261), IP addresses are assigned to hostnames. If no name server is implemented, all hosts to which an IP connection will be set up must be listed here. For each host, enter a line consisting of the IP address, the fully qualified hostname, and the hostname into the file. The IP address must be at the beginning of the line and the entries separated by blanks and tabs. Comments are always preceded by the `#` sign.

Example 13.6: `/etc/hosts`

```
127.0.0.1 localhost
192.168.2.100 jupiter.example.com jupiter
```

192.168.2.101 venus.example.com venus

13.6.1.7 /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 13.7, “`/etc/networks`” (page 262).

Example 13.7: */etc/networks*

```
loopback      127.0.0.0
localnet      192.168.0.0
```

13.6.1.8 /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 13.7, “Parameters for `/etc/host.conf`” (page 262) shows the parameters available. A sample `/etc/host.conf` is shown in Example 13.8, “`/etc/host.conf`” (page 263).

Table 13.7: *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> : uses 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 do not exert any

	influence on the network configuration.
<code>trim domainname</code>	The specified domain name is separated from the hostname after hostname resolution (as long as the hostname includes the domain name). This option is useful only if names from the local domain are in the <code>/etc/hosts</code> file, but should still be recognized with the attached domain names.

Example 13.8: */etc/host.conf*

```
# We have named running
order hosts bind
# Allow multiple address
multi on
```

13.6.1.9 /etc/nsswitch.conf

The introduction of the GNU C Library 2.0 was accompanied by the introduction of the *Name Service Switch* (NSS). Refer to the `nsswitch.conf(5)` man page and *The GNU C Library Reference Manual* for details.

The order for queries is defined in the file `/etc/nsswitch.conf`. A sample `nsswitch.conf` is shown in Example 13.9, “`/etc/nsswitch.conf`” (page 263). Comments are preceded by `#` signs. In this example, the entry under the `hosts` database means that a request is sent to `/etc/hosts` (files) via DNS.

Example 13.9: */etc/nsswitch.conf*

```
passwd:      compat
group:       compat

hosts:       files dns
networks:    files dns

services:    db files
protocols:   db files
rpc:         files
```

```
ethers:      files
netmasks:   files
netgroup:    files nis
publickey:   files

bootparams:  files
automount:   files nis
aliases:     files nis
shadow:      compat
```

The “databases” available over NSS are listed in Table 13.8, “Databases Available via /etc/nsswitch.conf” (page 264). The configuration options for NSS databases are listed in Table 13.9, “Configuration Options for NSS “Databases”” (page 265).

Table 13.8: *Databases Available via /etc/nsswitch.conf*

aliases	Mail aliases implemented by send-mail; see man 5 aliases.
ethers	Ethernet addresses.
netmasks	List of network and their subnet masks. Only needed, if you use subnetting.
group	For user groups used by getgrent. See also the man page for group.
hosts	For hostnames and IP addresses, used by gethostbyname and similar functions.
netgroup	Valid host and user lists in the network for the purpose of controlling access permissions; see the netgroup(5) man page.
networks	Network names and addresses, used by getnetent.
publickey	Public and secret keys for Secure_RPC used by NFS and NIS+.

passwd	User passwords, used by <code>getpwent</code> ; see the <code>passwd(5)</code> man page.
protocols	Network protocols, used by <code>getprotoent</code> ; see the <code>protocols(5)</code> man page.
rpc	Remote procedure call names and addresses, used by <code>getrpcbyname</code> and similar functions.
services	Network services, used by <code>getservent</code> .
shadow	Shadow passwords of users, used by <code>getspnam</code> ; see the <code>shadow(5)</code> man page.

Table 13.9: *Configuration Options for NSS “Databases”*

files	directly access files, for example, <code>/etc/aliases</code>
db	access via a database
nis, nisplus	NIS, see also Chapter 3, <i>Using NIS</i> (↑Security Guide)
dns	can only be used as an extension for <code>hosts</code> and <code>networks</code>
compat	can only be used as an extension for <code>passwd</code> , <code>shadow</code> and <code>group</code>

13.6.1.10 /etc/nscd.conf

This file is used to configure `nscd` (name service cache daemon). See the `nscd(8)` and `nscd.conf(5)` man pages. By default, the system entries of `passwd` and

groups are cached by `nscd`. This is important for the performance of directory services, like NIS and LDAP, because otherwise the network connection needs to be used for every access to names or groups. `hosts` is not cached by default, because the mechanism in `nscd` to cache `hosts` makes the local system unable to trust forward and reverse lookup checks. Instead of asking `nscd` to cache names, set up a caching DNS server.

If the caching for `passwd` is activated, it usually takes about fifteen seconds until a newly added local user is recognized. Reduce this waiting time by restarting `nscd` with the command `rcnsd restart`.

13.6.1.11 /etc/HOSTNAME

This contains the fully qualified hostname with the domain name attached. This file is read by several scripts while the machine is booting. It must contain only one line (in which the hostname is set).

13.6.2 Testing the Configuration

Before you write your configuration to the configuration files, you can test it. To set up a test configuration, use the `ip` command. To test the connection, use the `ping` command. Older configuration tools, `ifconfig` and `route`, are also available.

The commands `ip`, `ifconfig` and `route` change the network configuration directly without saving it in the configuration file. Unless you enter your configuration in the correct configuration files, the changed network configuration is lost on reboot.

13.6.2.1 Configuring a Network Interface with `ip`

`ip` is a tool to show and configure network devices, routing, policy routing, and tunnels.

`ip` is a very complex tool. Its common syntax is `ip options object command`. You can work with the following objects:

`link`

This object represents a network device.

`address`

This object represents the IP address of device.

neighbor

This object represents a ARP or NDISC cache entry.

route

This object represents the routing table entry.

rule

This object represents a rule in the routing policy database.

maddress

This object represents a multicast address.

mroute

This object represents a multicast routing cache entry.

tunnel

This object represents a tunnel over IP.

If no command is given, the default command is used (usually `list`).

Change the state of a device with the command `ip link`

`set device_name command`. For example, to deactivate device `eth0`, enter `ip link set eth0 down`. To activate it again, use `ip link set eth0 up`.

After activating a device, you can configure it. To set the IP address, use `ip addr add ip_address + dev device_name`. For example, to set the address of the interface `eth0` to `192.168.12.154/30` with standard broadcast (option `brd`), enter `ip addr add 192.168.12.154/30 brd + dev eth0`.

To have a working connection, you must also configure the default gateway. To set a gateway for your system, enter `ip route add gateway_ip_address`.

To translate one IP address to another, use `nat: ip route add nat_ip_address via other_ip_address`.

To display all devices, use `ip link ls`. To display the running interfaces only, use `ip link ls up`. To print interface statistics for a device, enter `ip -s link ls device_name`. To view addresses of your devices, enter `ip addr`. In the output of the `ip addr`, also find information about MAC addresses of your devices. To show all routes, use `ip route show`.

For more information about using `ip`, enter `ip help` or see the `ip(8) man` page. The `help` option is also available for all `ip` subcommands. If, for example, you need

help for ip addr, enter `ip addr help`. Find the ip manual in `/usr/share/doc/packages/iproute2/ip-cref.pdf`.

13.6.2.2 Testing a Connection with ping

The `ping` command is the standard tool for testing whether a TCP/IP connection works. It uses the ICMP protocol to send a small data packet, `ECHO_REQUEST` datagram, to the destination host, requesting an immediate reply. If this works, `ping` displays a message to that effect, which indicates that the network link is basically functioning.

`ping` does more than only test the function of the connection between two computers: it also provides some basic information about the quality of the connection. In Example 13.10, “Output of the Command `ping`” (page 268), you can see an example of the `ping` output. The second-to-last line contains information about the number of transmitted packets, packet loss, and total time of `ping` running.

As the destination, you can use a hostname or IP address, for example, `ping example.com` or `ping 192.168.3.100`. The program sends packets until you press `Ctrl + C`.

If you only need to check the functionality of the connection, you can limit the number of the packets with the `-c` option. For example to limit `ping` to three packets, enter `ping -c 3 example.com`.

Example 13.10: *Output of the Command `ping`*

```
ping -c 3 example.com
PING example.com (192.168.3.100) 56(84) bytes of data.
64 bytes from example.com (192.168.3.100): icmp_seq=1 ttl=49 time=188 ms
64 bytes from example.com (192.168.3.100): icmp_seq=2 ttl=49 time=184 ms
64 bytes from example.com (192.168.3.100): icmp_seq=3 ttl=49 time=183 ms
--- example.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 183.417/185.447/188.259/2.052 ms
```

The default interval between two packets is one second. To change the interval, `ping` provides the option `-i`. For example, to increase the ping interval to ten seconds, enter `ping -i 10 example.com`.

In a system with multiple network devices, it is sometimes useful to send the ping through a specific interface address. To do so, use the `-I` option with the name of the selected device, for example, `ping -I wlan1 example.com`.

For more options and information about using ping, enter `ping -h` or see the ping (8) man page.

TIP: Pinging IPv6 Addresses

For IPv6 addresses use the `ping6` command. Note, to ping link-local addresses, you must specify the interface with `-I`. The following command works, if the address is reachable via `eth1`:

```
ping6 -I eth1 fe80::117:21ff:feda:a425
```

13.6.2.3 Configuring the Network with ifconfig

`ifconfig` is a network configuration tool.

NOTE: ifconfig and ip

The `ifconfig` tool is obsolete. Use `ip` instead. In contrast to `ip`, you can use `ifconfig` only for interface configuration. It limits interface names to 9 characters.

Without arguments, `ifconfig` displays the status of the currently active interfaces. As you can see in Example 13.11, “Output of the `ifconfig` Command” (page 269), `ifconfig` has very well-arranged and detailed output. The output also contains information about the MAC address of your device (the value of `HWaddr`) in the first line.

Example 13.11: Output of the ifconfig Command

```
eth0      Link encap:Ethernet  HWaddr 00:08:74:98:ED:51
          inet6 addr: fe80::208:74ff:fe98:ed51/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:634735 errors:0 dropped:0 overruns:4 frame:0
          TX packets:154779 errors:0 dropped:0 overruns:0 carrier:1
          collisions:0 txqueuelen:1000
          RX bytes:162531992 (155.0 Mb)  TX bytes:49575995 (47.2 Mb)
          Interrupt:11 Base address:0xec80

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8559 errors:0 dropped:0 overruns:0 frame:0
```

```

TX packets:8559 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:533234 (520.7 Kb)  TX bytes:533234 (520.7 Kb)

wlan1    Link encap:Ethernet  HWaddr 00:0E:2E:52:3B:1D
         inet addr:192.168.2.4  Bcast:192.168.2.255  Mask:255.255.255.0
         inet6 addr: fe80::20e:2eff:fe52:3b1d/64  Scope:Link
         UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:50828 errors:0 dropped:0 overruns:0 frame:0
         TX packets:43770 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:45978185 (43.8 Mb)  TX bytes:7526693 (7.1 MB)

```

For more options and information about using `ifconfig`, enter `ifconfig -h` or see the `ifconfig (8)` man page.

13.6.2.4 Configuring Routing with `route`

`route` is a program for manipulating the IP routing table. You can use it to view your routing configuration and to add or remove routes.

NOTE: `route` and `ip`

The program `route` is obsolete. Use `ip` instead.

`route` is especially useful if you need quick and comprehensible information about your routing configuration to determine problems with routing. To view your current routing configuration, enter `route -n` as root.

Example 13.12: *Output of the `route -n` Command*

```

route -n
Kernel IP routing table
Destination    Gateway         Genmask         Flags   MSS Window  irtt Iface
10.20.0.0      *               255.255.248.0   U        0 0          0 eth0
link-local     *               255.255.0.0     U        0 0          0 eth0
loopback       *               255.0.0.0       U        0 0          0 lo
default        styx.exam.com   0.0.0.0         UG       0 0          0 eth0

```

For more options and information about using `route`, enter `route -h` or see the `route (8)` man page.

13.6.3 Start-Up Scripts

Apart from the configuration files described above, there are also various scripts that load the network programs while the machine is booting. These are started as soon as the system is switched to one of the *multiuser runlevels*. Some of these scripts are described in Table 13.10, “Some Start-Up Scripts for Network Programs” (page 271).

Table 13.10: *Some Start-Up Scripts for Network Programs*

<code>/etc/init.d/network</code>	This script handles the configuration of the network interfaces. If the <code>network</code> service was not started, no network interfaces are implemented.
<code>/etc/init.d/xinetd</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/rpcbind</code>	Starts the <code>rpcbind</code> utility that converts RPC program numbers to universal addresses. It is needed for RPC services, such as an NFS server.
<code>/etc/init.d/nfsserver</code>	Starts the NFS server.
<code>/etc/init.d/postfix</code>	Controls the postfix process.
<code>/etc/init.d/ypserv</code>	Starts the NIS server.
<code>/etc/init.d/ypbind</code>	Starts the NIS client.

13.7 smpppd as Dial-up Assistant

Some home users do not have a dedicated line connecting them to the Internet. Instead, they use dial-up connections. Depending on the dial-up method (ISDN or DSL), the connection is controlled by `ippd` 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 desktop 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` (SUSE Meta PPP Daemon) is involved. It provides a uniform interface for auxiliary programs and acts in two directions. First, it programs the required `pppd` or `ipppd` and controls its dial-up properties. Second, it makes various providers available to the user programs and transmits information about the current status of the connection. As `smpppd` can also be controlled by way of the network, it is suitable for controlling dial-up connections to the Internet from a workstation in a private subnetwork.

13.7.1 Configuring `smpppd`

The connections provided by `smpppd` are automatically configured by YaST. The actual dial-up programs `KInternet` and `cinternet` are also preconfigured. Manual settings are only required to configure additional features of `smpppd` such as remote control.

The configuration file of `smpppd` is `/etc/smpppd.conf`. By default, it does not enable remote control. The most important options of this configuration file are:

`open-inet-socket = yes/no`

To control `smpppd` via the network, set this option to `yes`. `smpppd` listens on port 3185. If this parameter is set to `yes`, the parameters `bind-address`, `host-range` and `password` must be set accordingly.

`bind-address = ip address`

If a host has several IP addresses, use this parameter to determine at which IP address `smpppd` should accept connections. The default is to listen at all addresses.

`host-range = min ipmax ip`

The parameter `host-range` defines a network range. Hosts whose IP addresses are within this range are granted access to `smpppd`. All hosts not within this range are denied access.

`password = password`

By assigning a password, limit the clients to authorized hosts. As this is a plain-text password, you should not overrate the security it provides. If no password is assigned, all clients are permitted to access `smpppd`.

`slp-register = yes/no`

With this parameter, the smpppd service can be announced in the network via SLP.

More information about smpppd is available in the `smpppd(8)` and `smpppd.conf(5)` man pages.

13.7.2 Configuring qinternet for Remote Use

qinternet can be used to control a local or remote smpppd. cinternet is the command-line counterpart to the graphical KInternet. To prepare these utilities for use with a remote smpppd, edit the configuration file `/etc/smpppd-c.conf` manually or using qinternet. This file only uses four options:

`sites = list of sites`

`list of sites` where the front-ends search for smpppd. The front-ends test the options in the order specified here. `local` orders the establishment of a connection to the local smpppd. `gateway` points to an smpppd on the gateway. `config-file` indicates that the connection should be established to the smpppd specified in the `server` and `port` options in `/etc/smpppd-c.conf`. `slp` orders the front-ends to connect to an smpppd found via SLP.

`server = server`

The host on which smpppd runs.

`port = port`

The port on which smpppd runs.

`password = password`

The password selected for smpppd.

SLP Services in the Network

The *service location protocol* (SLP) was developed to simplify the configuration of networked clients within a local network. To configure a network client, including all required services, the administrator traditionally needs detailed knowledge of the servers available in the network. SLP makes the availability of selected services known to all clients in the local network. Applications that support SLP can use the information distributed and be configured automatically.

openSUSE® supports installation using installation sources provided with SLP and contains many system services with integrated support for SLP. YaST and Konqueror both have appropriate front-ends for SLP. You can use SLP to provide networked clients with central functions, such as an installation server, file server, or print server on your system.

IMPORTANT: SLP Support in openSUSE

Services that offer SLP support include cupsd, rsyncd, ypserv, openldap2, ksysguardd, saned, kdm, vnc, login, smpppd, rpasswd, postfix, and sshd (via fish).

14.1 Installation

All packages necessary to use SLP services are installed by default. However, if you want to provide services via SLP, check that the `openslp-server` package is installed. For SLP daemon server configuration install the `yast2-slp-server` package.

14.2 Activating SLP

slpd must run on your system to offer services with SLP. If the machine should only operate as client, and does not offer services, it is not necessary to run slpd. Like most system services in openSUSE, the slpd daemon is controlled by means of a separate `init` script. After the installation, the daemon is inactive by default. To activate it temporarily, run `rcslpd start` as root or `rcslpd stop` to stop it. Perform a restart or status check with `restart` or `status`. If slpd should be always active after booting, enable slpd in YaST *System > System Services (Runlevel)* or run the `insserv slpd` command as root.

14.3 SLP Front-Ends in openSUSE

To find services provided via SLP in your network, use an SLP front-end such as `slptool` (openslp package) or YaST:

slptool

slptool is a command line program that can be used to announce SLP inquiries in the network or announce proprietary services. `slptool --help` lists all available options and functions. For example, to find all time servers that announce themselves in the current network, run the command:

```
slptool findsrvs service:ntp
```

YaST

YaST also provides an SLP browser. However, this browser is not available from the YaST Control Center. To start it, run `yast2 slp` as root user. Click on a *Service Type* on the lefthand side to get more information about a service.

14.4 Installation over SLP

If you have an installation server with openSUSE installation media within your network, this can be registered and offered with SLP. For details, see Section 2.2, “Setting Up the Server Holding the Installation Sources” (page 44). If SLP installation is selected, `linuxrc` starts an SLP inquiry after the system has booted from the selected boot medium and displays the sources found.

14.5 Providing Services via SLP

Many applications in openSUSE have integrated SLP support through the use of the `libslp` library. If a service has not been compiled with SLP support, use one of the following methods to make it available via SLP:

Static Registration with `/etc/slp.reg.d`

Create a separate registration file for each new service. This is an example for registering a scanner service:

```
## Register a saned service on this system
## en means english language
## 65535 disables the timeout, so the service registration does
## not need refreshes
service:scanner.sane://$HOSTNAME:6566,en,65535
watch-port-tcp=6566
description=SANE scanner daemon
```

The most important line in this file is the *service URL*, which begins with `service:.` This contains the service type (`scanner.sane`) and the address under which the service is available on the server. `$HOSTNAME` is automatically replaced with the full hostname. The name of the TCP port on which the relevant service can be found follows, separated by a colon. Then enter the language in which the service should appear and the duration of registration in seconds. These should be separated from the service URL by commas. Set the value for the duration of registration between 0 and 65535. 0 prevents registration. 65535 removes all restrictions.

The registration file also contains the two variables `watch-port-tcp` and `description`. `watch-port-tcp` links the SLP service announcement to whether the relevant service is active by having `slpd` check the status of the service. The second variable contains a more precise description of the service that is displayed in suitable browsers.

Static Registration with `/etc/slp.reg`

The only difference between this method and the procedure with `/etc/slp.reg.d` is that all services are grouped within a central file.

Dynamic Registration with `slptool`

If a service needs to be registered dynamically without the need of configuration files, use the `slptool` command line utility. The same utility can also be used to deregister an existing service offering without restarting `slpd`.

14.6 For More Information

RFC 2608, 2609, 2610

RFC 2608 generally deals with the definition of SLP. RFC 2609 deals with the syntax of the service URLs used in greater detail and RFC 2610 deals with DHCP via SLP.

<http://www.openslp.org>

The home page of the OpenSLP project.

`/usr/share/doc/packages/openslp`

This directory contains the documentation for SLP coming with the `openslp-server` package, including a `README.SuSE` containing the openSUSE details, the RFCs, and two introductory HTML documents. Programmers who want to use the SLP functions find more information in the *Programmers Guide* that is included in the `openslp-devel` package.

The Domain Name System

DNS (domain name system) is needed to resolve the domain names and hostnames into IP addresses. In this way, the IP address 192.168.2.100 is assigned to the hostname `jupiter`, for example. Before setting up your own name server, read the general information about DNS in Section 13.3, “Name Resolution” (page 231). The following configuration examples refer to BIND.

15.1 DNS Terminology

Zone

The domain namespace is divided into regions called zones. For instance, if you have `example.com`, you have the `example` section (or zone) of the `com` domain.

DNS server

The DNS server is a server that maintains the name and IP information for a domain. You can have a primary DNS server for master zone, a secondary server for slave zone, or a slave server without any zones for caching.

Master zone DNS server

The master zone includes all hosts from your network and a DNS server master zone stores up-to-date records for all the hosts in your domain.

Slave zone DNS server

A slave zone is a copy of the master zone. The slave zone DNS server obtains its zone data with zone transfer operations from its master server. The slave

zone DNS server responds authoritatively for the zone as long as it has valid (not expired) zone data. If the slave cannot obtain a new copy of the zone data, it stops responding for the zone.

Forwarder

Forwarders are DNS servers to which your DNS server should send queries it cannot answer. To enable different configuration sources in one configuration, `netconfig` is used (see also `man 8 netconfig`).

Record

The record is information about name and IP address. Supported records and their syntax are described in BIND documentation. Some special records are:

NS record

An NS record tells name servers which machines are in charge of a given domain zone.

MX record

The MX (mail exchange) records describe the machines to contact for directing mail across the Internet.

SOA record

SOA (Start of Authority) record is the first record in a zone file. The SOA record is used when using DNS to synchronize data between multiple computers.

15.2 Installation

To install a DNS server, start YaST and select *Software > Software Management*. Choose *View > Patterns* and select *DHCP and DNS Server*. Confirm the installation of the dependent packages to finish the installation process.

15.3 Configuration with YaST

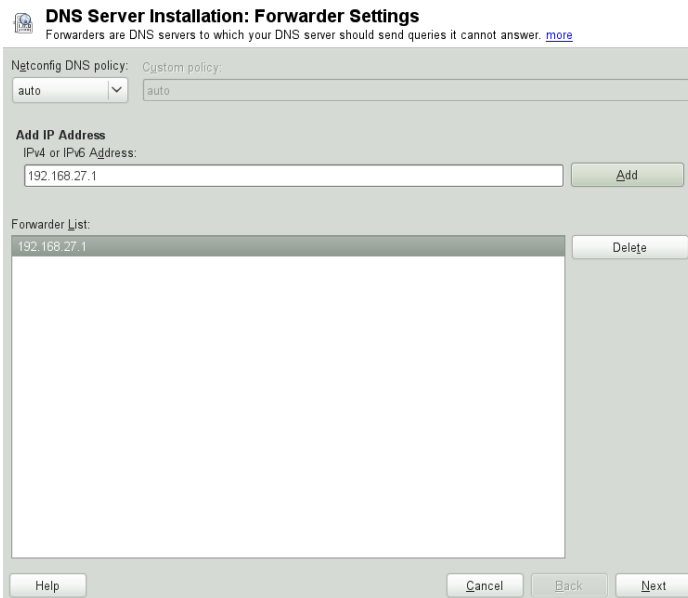
Use the YaST DNS module to configure a DNS server for the local network. When starting the module for the first time, a wizard starts, prompting you to make a few decisions concerning administration of the server. Completing this initial setup produces a basic server configuration. Use the expert mode to deal with more advanced configuration tasks, such as setting up ACLs, logging, TSIG keys, and other options.

15.3.1 Wizard Configuration

The wizard consists of three steps or dialogs. At the appropriate places in the dialogs, you are given the opportunity to enter the expert configuration mode.

- 1 When starting the module for the first time, the *Forwarder Settings* dialog, shown in Figure 15.1, “DNS Server Installation: Forwarder Settings” (page 281), opens. The *Netconfig DNS Policy* decides which devices should provide forwarders or whether you want to supply your own *Forwarder List*. For more information about *netconfig*, see `man 8 netconfig`.

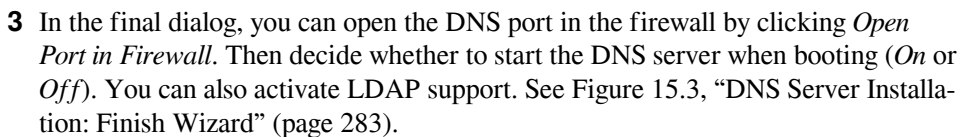
Figure 15.1: *DNS Server Installation: Forwarder Settings*



Forwarders are DNS servers to which your DNS server sends queries it cannot answer itself. Enter their IP address and click *Add*.

- 2 The *DNS Zones* dialog consists of several parts and is responsible for the management of zone files, described in Section 15.6, “Zone Files” (page 296). For a new zone, provide a name for it in *Name*. To add a reverse zone, the name must end in `.in-addr.arpa`. Finally, select the *Type* (master, slave, or forward). See

Figure 15.2: *DNS Server Installation: DNS Zones*



[illegible]

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:

Under *Start-Up*, define whether the DNS server should be started when the booting the system or manually. To start the DNS server immediately, click *Start DNS Server Now*. To stop the DNS server, click *Stop DNS Server Now*. To save the current settings, select *Save Settings and Reload 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*.

The Domain Name System 283

15.3.2.2 Forwarders

If your local DNS server cannot answer a request, it tries to forward the request to a *Forwarder*, if configured so. This forwarder may be added manually to the *Forwarder List*. If the forwarder is not static like in dial-up connections, *netconfig* handles the configuration. For more information about *netconfig*, see `man 8 netconfig`.

15.3.2.3 Basic Options

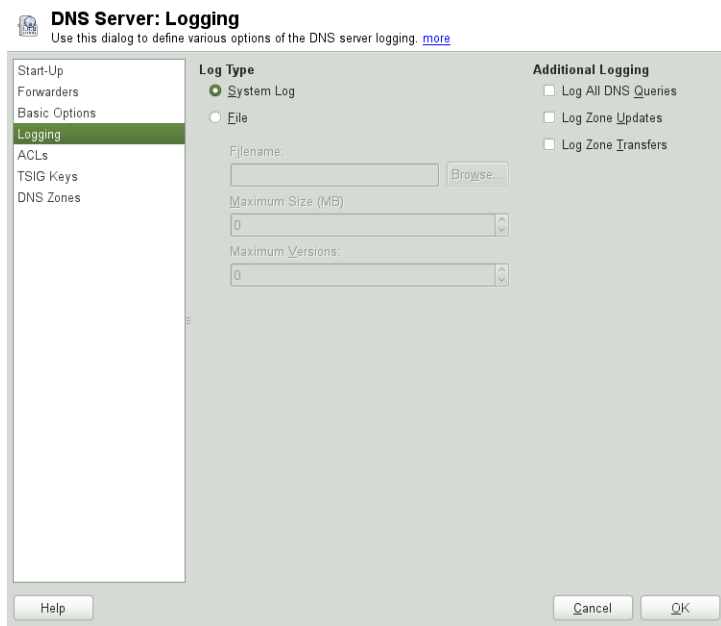
In this section, set basic server options. From the *Option* menu, select the desired item then specify the value in the corresponding entry field. Include the new entry by selecting *Add*.

15.3.2.4 Logging

To set what the DNS server should log and how, select *Logging*. Under *Log Type*, specify where the DNS server should write the log data. Use the systemwide log file `/var/log/messages` by selecting *System Log* or specify a different file by selecting *File*. In the latter case, additionally specify a name, the maximum file size in megabytes and the number of logfile versions to store.

Further options are available under *Additional Logging*. Enabling *Log All DNS Queries* causes *every* query to be logged, in which case the log file could grow extremely large. For this reason, it is not a good idea to enable this option for other than debugging purposes. To log the data traffic during zone updates between DHCP and DNS server, enable *Log Zone Updates*. To log the data traffic during a zone transfer from master to slave, enable *Log Zone Transfer*. See Figure 15.4, “DNS Server: Logging” (page 285).

Figure 15.4: *DNS Server: Logging*



15.3.2.5 ACLs

Use this dialog to define ACLs (access control lists) to enforce access restrictions. After providing a distinct name under *Name*, specify an IP address (with or without net-mask) under *Value* in the following fashion:

```
{ 192.168.1/24; }
```

The syntax of the configuration file requires that the address ends with a semicolon and is put into curly braces.

15.3.2.6 TSIG Keys

The main purpose of TSIGs (transaction signatures) is to secure communications between DHCP and DNS servers. They are described in Section 15.8, “Secure Transactions” (page 300).

To generate a TSIG key, enter a distinctive name in the field labeled *Key ID* and specify the file where the key should be stored (*Filename*). Confirm your choices with *Generate*.

To use a previously created key, leave the *Key ID* field blank and select the file where it is stored under *Filename*. After that, confirm with *Add*.

15.3.2.7 DNS Zones (Adding a Slave Zone)

To add a slave zone, select *DNS Zones*, choose the zone type *Slave*, write the name of the new zone, and click *Add*.

In the *Zone Editor* sub-dialog under *Master DNS Server IP*, specify the master from which the slave should pull its data. To limit access to the server, select one of the ACLs from the list.

15.3.2.8 DNS Zones (Adding a Master Zone)

To add a master zone, select *DNS Zones*, choose the zone type *Master*, write the name of the new zone, and click *Add*. When adding a master zone, a reverse zone is also needed. For example, when adding the zone `example.com` that points to hosts in a subnet `192.168.1.0/24`, you should also add a reverse zone for the IP-address range covered. By definition, this should be named `1.168.192.in-addr.arpa`.

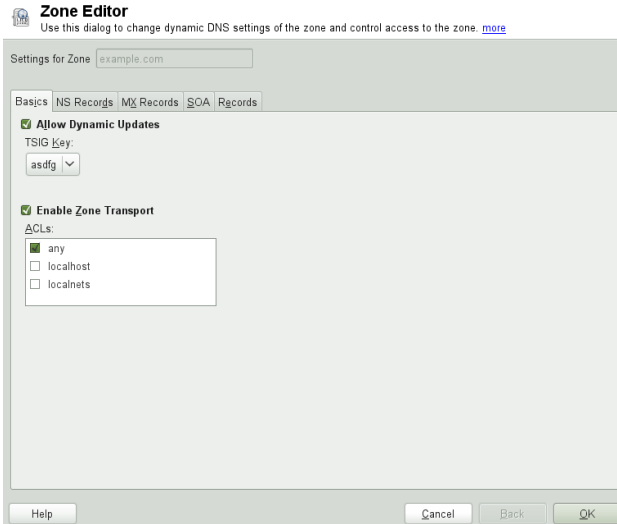
15.3.2.9 DNS Zones (Editing a Master Zone)

To edit a master zone, select *DNS Zones*, select the master zone from the table, and click *Edit*. The dialog consists of several pages: *Basics* (the one opened first), *NS Records*, *MX Records*, *SOA*, and *Records*.

The basic dialog, shown in Figure 15.5, “DNS Server: Zone Editor (Basics)” (page 287), lets you define settings for dynamic DNS and access options for zone transfers to clients and slave name servers. To permit the dynamic updating 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.

In the *Basics* dialog, select whether to enable zone transfers. Use the listed ACLs to define who can download zones.

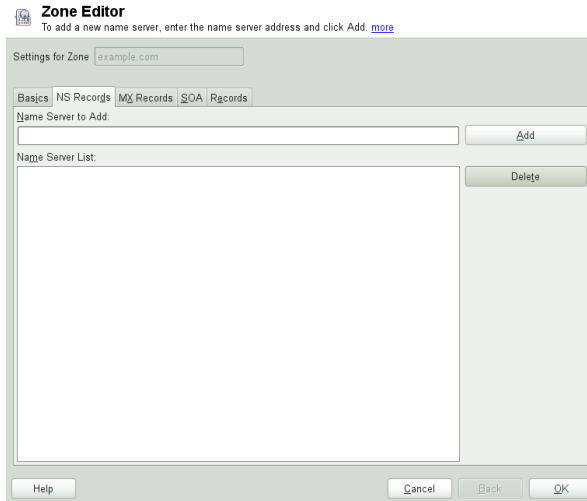
Figure 15.5: *DNS Server: Zone Editor (Basics)*



Zone Editor (NS Records)

The *NS Records* 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 15.6, “DNS Server: Zone Editor (NS Records)” (page 288).

Figure 15.6: *DNS Server: Zone Editor (NS Records)*



Zone Editor (MX Records)

To add a mail server for the current zone to the existing list, enter the corresponding address and priority value. After doing so, confirm by selecting *Add*. See Figure 15.7, “DNS Server: Zone Editor (MX Records)” (page 289).

Figure 15.7: *DNS Server: Zone Editor (MX Records)*

The screenshot shows the 'Zone Editor' window for 'example.com'. It has tabs for 'Basics', 'NS Records', 'MX Records', 'SOA', and 'Records'. The 'MX Records' tab is active. Under 'Mail Server to Add', there is a text field for 'Address' and a spinner for 'Priority' (set to 0), with an 'Add' button. Below this is a 'Mail Relay List' section with a table header 'Mail Server' and 'Priority'. The table is empty, showing 'No entries.' A 'Delete' button is to the right of the table. At the bottom are 'Help', 'Cancel', 'Back', and 'OK' buttons.

Zone Editor
To add a new mail server, enter the Address and Priority and click Add. [more](#)

Settings for Zone:

Basics | NS Records | **MX Records** | SOA | Records

Mail Server to Add

Address:

Priority:

Add

Mail Relay List

Mail Server	Priority
No entries.	

Delete

Help Cancel Back OK

Zone Editor (SOA)

This page allows you to create SOA (start of authority) records. For an explanation of the individual options, refer to Example 15.6, “The /var/lib/named/example.com.zone File” (page 296).

Figure 15.8: *DNS Server: Zone Editor (SOA)*

Zone Editor
Set the entries of the SOA record. [more](#)

Settings for Zone

Basics | NS Records | MX Records | **SOA** | Records

Serial:

TTL: Unit:

Refresh: Unit:

Retry: Unit:

Expiration: Unit:

Minimum: Unit:

Help Cancel Back OK

Zone Editor (Records)

This dialog manages name resolution. In *Record Key*, enter the hostname then select its type. *A-Record* represents the main entry. The value for this should be an IP address. *CNAME* is an alias. Use the types *NS* and *MX* for detailed or partial records that expand on the information provided in the *NS Records* and *MX Records* tabs. These three types resolve to an existing *A* record. *PTR* is for reverse zones. It is the opposite of an *A* record, for example:

```
hostname.example.com. IN A 192.168.0.1
1.0.168.192.in-addr.arpa IN PTR hostname.example.com.
```

NOTE: Editing the Reverse Zone

After adding a forward zone, go back to the main menu and select the reverse zone for editing. There in the tab *Basics* activate the checkbox *Automatically Generate Records From* and select your forward zone. That way, all changes to the forward zone are automatically updated in the reverse zone.

15.4 Starting the BIND Name Server

On a openSUSE® system, the name server BIND (*Berkeley Internet Name Domain*) comes preconfigured so it can be started right after installation without any problems. If you already have a functioning Internet connection and have entered 127.0.0.1 as the name server address for localhost in `/etc/resolv.conf`, you normally already have a working name resolution without needing to know the DNS of the provider. BIND carries out name resolution via the root name server, a notably slower process. Normally, the DNS of the provider should be entered with its IP address in the configuration file `/etc/named.conf` under `forwarders` to ensure effective and secure name resolution. If this works so far, the name server runs as a pure *caching-only* name server. Only when you configure its own zones it becomes a proper DNS. Find a simple example documented in `/usr/share/doc/packages/bind/config`.

TIP: Automatic Adaptation of the Name Server Information

Depending on the type of Internet connection or the network connection, the name server information can automatically be adapted to the current conditions. To do this, set the `NETCONFIG_DNS_POLICY` variable in the `/etc/sysconfig/network/config` file to `auto`.

However, do not set up an official domain until one is assigned to you by the responsible institution. Even if you have your own domain and it is managed by the provider, you are better off not using it, because BIND would otherwise not forward requests for this domain. The Web server at the provider, for example, would not be accessible for this domain.

To start the name server, enter the command `rndc start` as root. If “done” appears to the right in green then 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 unexpectedly, you can usually find the cause in the log file `/var/log/messages`.

To use the name server of the provider (or one already running on your network) as the forwarder, enter the corresponding IP address or addresses in the `options` section under `forwarders`. The addresses included in Example 15.1, “Forwarding Op-

tions in `named.conf`” (page 292) are just examples. Adjust these entries to your own setup.

Example 15.1: *Forwarding Options in `named.conf`*

```
options {
    directory "/var/lib/named";
    forwarders { 10.11.12.13; 10.11.12.14; };
    listen-on { 127.0.0.1; 192.168.1.116; };
    allow-query { 127/8; 192.168/16 };
    notify no;
};
```

The `options` entry is followed by entries for the zone, `localhost`, and `0.0.127.in-addr.arpa`. The `type hint` entry under “.” should always be present. The corresponding files do not need to be modified and should work as they are. Also make sure that each entry is closed with a “;” and that the curly braces are in the correct places. After changing the configuration file `/etc/named.conf` or the zone files, tell BIND to reread them with `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`.

15.5 The `/etc/named.conf` Configuration File

All the settings for the BIND name server itself are stored in the `/etc/named.conf` file. However, the zone data for the domains to handle (consisting of the hostnames, IP addresses, and so on) are stored in separate files in the `/var/lib/named` directory. The details of this are described later.

`/etc/named.conf` is roughly divided into two areas. One is the `options` section for general settings and the other consists of `zone` entries for the individual domains. A logging section and `acl` (access control list) entries are optional. Comment lines begin with a `#` sign or `//`. A minimal `/etc/named.conf` is shown in Example 15.2, “A Basic `/etc/named.conf`” (page 292).

Example 15.2: *A Basic `/etc/named.conf`*

```
options {
    directory "/var/lib/named";
    forwarders { 10.0.0.1; };
    notify no;
```

```

};

zone "localhost" in {
    type master;
    file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" in {
    type master;
    file "127.0.0.zone";
};

zone "." in {
    type hint;
    file "root.hint";
};

```

15.5.1 Important Configuration Options

`directory "filename";`

Specifies the directory in which BIND can find the files containing the zone data. Usually, this is `/var/lib/named`.

`forwarders { ip-address; };`

Specifies the name servers (mostly of the provider) to which DNS requests should be forwarded if they cannot be resolved directly. Replace *ip-address* with an IP address like `192.168.1.116`.

`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 wild card addresses.

`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 `192.168.2.0/24`. The `/24` at the end is an abbreviated expression for the netmask (in this case `255.255.255.0`).

`allow-transfer ! *;;`

Controls which hosts can request zone transfers. In the example, such requests are completely denied with `! *`. Without this entry, zone transfers can be requested from anywhere without restrictions.

`statistics-interval 0;`

In the absence of this entry, BIND generates several lines of statistical information per hour in `/var/log/messages`. Set it to 0 to suppress these statistics completely or set an interval in minutes.

`cleaning-interval 720;`

This option defines at which time intervals BIND clears its cache. This triggers an entry in `/var/log/messages` each time it occurs. The time specification is in minutes. The default is 60 minutes.

`interface-interval 0;`

BIND regularly searches the network interfaces for new or nonexistent interfaces. If this value is set to 0, this is not done and BIND only listens at the interfaces 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.

For a list of available options, read the manual page `man 5 named.conf`.

15.5.2 Logging

What, how, and where logging takes place can be extensively configured in BIND. Normally, the default settings should be sufficient. Example 15.3, “Entry to Disable Logging” (page 295), shows the simplest form of such an entry and completely suppresses any logging.

Example 15.3: *Entry to Disable Logging*

```
logging {  
    category default { null; };  
};
```

15.5.3 Zone Entries

Example 15.4: *Zone Entry for example.com*

```
zone "example.com" in {  
    type master;  
    file "example.com.zone";  
    notify no;  
};
```

After `zone`, specify the name of the domain to administer (`example.com`) followed by `in` and a block of relevant options enclosed in curly braces, as shown in Example 15.4, “Zone Entry for example.com” (page 295). 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 15.5, “Zone Entry for example.net” (page 295).

Example 15.5: *Zone Entry for example.net*

```
zone "example.net" in {  
    type slave;  
    file "slave/example.net.zone";  
    masters { 10.0.0.1; };  
};
```

The zone options:

`type master;`

By specifying `master`, tell BIND that the zone is handled by the local name server. This assumes that a zone file has been created in the correct format.

`type slave;`

This zone is transferred from another name server. It must be used together with `masters`.

type hint;

The zone `.` of the `hint` type is used to set the root name servers. This zone definition can be left as is.

file `example.com.zone` or file “`slave/example.net.zone`”;

This entry specifies the file where zone data for the domain is located. This file is not required for a slave, because this data is pulled 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.

15.6 Zone Files

Two types of zone files are needed. One assigns IP addresses to hostnames and the other does the reverse: it supplies a hostname for an IP address.

TIP: Using the Dot (Period, Fullstop) in Zone Files

The “.” has an important meaning in the zone files. If hostnames are given without a final “.”, the zone is appended. Complete hostnames specified with a full domain name must end with a “.” to avoid having the domain added to it again. A missing or wrongly placed “.” is probably the most frequent cause of name server configuration errors.

The first case to consider is the zone file `example.com.zone`, responsible for the domain `example.com`, shown in Example 15.6, “The `/var/lib/named/example.com.zone` File” (page 296).

Example 15.6: *The `/var/lib/named/example.com.zone` File*

```
1. $TTL 2D
2. example.com. IN SOA      dns root.example.com. (
3.                  2003072441 ; serial
4.                  1D        ; refresh
```

```

5.                2H                ; retry
6.                1W                ; expiry
7.                2D )              ; minimum
8.
9.                IN NS             dns
10.               IN MX             10 mail
11.
12. gate          IN A              192.168.5.1
13.               IN A              10.0.0.1
14. dns           IN A              192.168.1.116
15. mail          IN A              192.168.3.108
16. jupiter       IN A              192.168.2.100
17. venus         IN A              192.168.2.101
18. saturn        IN A              192.168.2.102
19. mercury       IN A              192.168.2.103
20. ntp           IN CNAME          dns
21. dns6          IN A6 0           2002:c0a8:174::

```

Line 1:

\$TTL defines the default time to live that should apply to all the entries in this file. In this example, entries are valid for a period of two days (2 D).

Line 2:

This is where the SOA (start of authority) control record begins:

- The name of the domain to administer is `example.com` in the first position. This ends with `"."`, because otherwise the zone would be appended a second time. Alternatively, `@` can be entered here, in which case the zone would be extracted from the corresponding entry in `/etc/named.conf`.
- After `IN SOA` is the name of the name server in charge as master for this zone. The name is expanded from `dns` to `dns.example.com`, because it does not end with a `"."`.
- An e-mail address of the person in charge of this name server follows. Because the `@` sign already has a special meaning, `"."` is entered here instead. For `root@example.com` the entry must read `root.example.com..` The `"."` must be included at the end to prevent the zone from being added.
- The `(` includes all lines up to `)` into the SOA record.

Line 3:

The `serial` number is an arbitrary number that is increased each time this file is changed. It is needed to inform the secondary name servers (slave servers)

of changes. For this, a 10 digit number of the date and run number, written as YYYYMMDDNN, has become the customary format.

Line 4:

The `refresh rate` specifies the time interval at which the secondary name servers verify the `zone serial number`. In this case, one day.

Line 5:

The `retry rate` specifies the time interval at which a secondary name server, in case of error, attempts to contact the primary server again. Here, two hours.

Line 6:

The `expiration time` specifies the time frame after which a secondary name server discards the cached data if it has not regained contact to the primary server. Here, a week.

Line 7:

The last entry in the SOA record specifies the `negative caching TTL`—the time for which results of unresolved DNS queries from other servers may be cached.

Line 9:

The `IN NS` specifies the name server responsible for this domain. `dns` is extended to `dns.example.com` because it does not end with a `"."`. There can be several lines like this—one for the primary and one for each secondary name server. If `notify` is not set to `no` in `/etc/named.conf`, all the name servers listed here are informed of the changes made to the zone data.

Line 10:

The `MX` record specifies the mail server that accepts, processes, and forwards e-mails for the domain `example.com`. In this example, this is the host `mail.example.com`. The number in front of the hostname is the preference value. If there are multiple `MX` entries, the mail server with the smallest value is taken first and, if mail delivery to this server fails, an attempt is made with the next higher value.

Lines 12–19:

These are the actual address records where one or more IP addresses are assigned to hostnames. The names are listed here without a `"."` because they do not include their domain, so `example.com` is added to all of them. Two IP addresses are assigned to the host `gate`, as it has two network cards. Wherever the host ad-

dress is a traditional one (IPv4), the record is marked with A. If the address is an IPv6 address, the entry is marked with AAAA.

NOTE: IPv6 Syntax

The IPv6 record has a slightly different syntax than IPv4. Because of the fragmentation possibility, it is necessary to provide information about missed bits before the address. To just fill up the IPv6 address with the needed number of “0”, add two colons at the correct place in the address.

```
pluto      AAAA 2345:00C1:CA11::1234:5678:9ABC:DEF0
pluto      AAAA 2345:00D2:DA11::1234:5678:9ABC:DEF0
```

Line 20:

The alias `ntp` can be used to address `dns` (CNAME means *canonical name*).

The pseudodomain `in-addr.arpa` is used for the reverse lookup of IP addresses into hostnames. It is appended to the network part of the address in reverse notation. So `192.168` is resolved into `168.192.in-addr.arpa`. See Example 15.7, “Reverse Lookup” (page 299).

Example 15.7: Reverse Lookup

```
1. $TTL 2D
2. 168.192.in-addr.arpa.  IN SOA dns.example.com. root.example.com. (
3.                        2003072441      ; serial
4.                        1D              ; refresh
5.                        2H              ; retry
6.                        1W              ; expiry
7.                        2D )            ; minimum
8.
9.                        IN NS          dns.example.com.
10.
11. 1.5                    IN PTR        gate.example.com.
12. 100.3                 IN PTR        www.example.com.
13. 253.2                 IN PTR        cups.example.com.
```

Line 1:

`$TTL` defines the standard TTL that applies to all entries here.

Line 2:

The configuration file should activate reverse lookup for the network `192.168`. Given that the zone is called `168.192.in-addr.arpa`, it should not be added to the hostnames. Therefore, all hostnames are entered in their complete

form—with their domain and with a " ." at the end. The remaining entries correspond to those described for the previous `example.com` example.

Lines 3–7:

See the previous example for `example.com`.

Line 9:

Again this line specifies the name server responsible for this zone. This time, however, the name is entered in its complete form with the domain and a " ." at the end.

Lines 11–13:

These are the pointer records hinting at the IP addresses on the respective hosts. Only the last part of the IP address is entered at the beginning of the line, without the " ." at the end. Appending the zone to this (without the `.in-addr.arpa`) results in the complete IP address in reverse order.

Normally, zone transfers between different versions of BIND should be possible without any problems.

15.7 Dynamic Update of Zone Data

The term *dynamic update* refers to operations by which entries in the zone files of a master server are added, changed, or deleted. This mechanism is described in RFC 2136. Dynamic update is configured individually for each zone entry by adding an optional `allow-update` or `update-policy` rule. Zones to update dynamically should not be edited by hand.

Transmit the entries to update to the server with the command `nsupdate`. For the exact syntax of this command, check the manual page for `nsupdate` (`man 8 nsupdate`). For security reasons, any such update should be performed using TSIG keys as described in Section 15.8, “Secure Transactions” (page 300).

15.8 Secure Transactions

Secure transactions can be made with the help of transaction signatures (TSIGs) based on shared secret keys (also called TSIG keys). This section describes how to generate and use such keys.

Secure transactions are needed for communication between different servers and for the dynamic update of zone data. Making the access control dependent on keys is much more secure than merely relying on IP addresses.

Generate a TSIG key with the following command (for details, see `man dnssec-keygen`):

```
dnssec-keygen -a hmac-md5 -b 128 -n HOST host1-host2
```

This creates two files with names similar to these:

```
Khost1-host2.+157+34265.private Khost1-host2.+157+34265.key
```

The key itself (a string like `ejIkuCyyGJwwuN3xAteKgg==`) is found in both files. To use it for transactions, the second file (`Khost1-host2.+157+34265.key`) must be transferred to the remote host, preferably in a secure way (using `scp`, for example). On the remote server, the key must be included in the `/etc/named.conf` file to enable a secure communication between `host1` and `host2`:

```
key host1-host2 {  
    algorithm hmac-md5;  
    secret "ejIkuCyyGJwwuN3xAteKgg=";  
};
```

WARNING: File Permissions of `/etc/named.conf`

Make sure that the permissions of `/etc/named.conf` are properly restricted. The default for this file is `0640`, with the owner being `root` and the group `named`. As an alternative, move the keys to an extra file with specially limited permissions, which is then included from `/etc/named.conf`. To include an external file, use:

```
include "filename"
```

Replace `filename` with an absolute path to your file with keys.

To enable the server `host1` to use the key for `host2` (which has the address `10.1.2.3` in this example), the server's `/etc/named.conf` must include the following rule:

```
server 10.1.2.3 {  
    keys { host1-host2. ;};  
};
```

Analogous entries must be included in the configuration files of `host2`.

Add TSIG keys for any ACLs (access control lists, not to be confused with file system ACLs) that are defined for IP addresses and address ranges to enable transaction security. The corresponding entry could look like this:

```
allow-update { key host1-host2. ;};
```

This topic is discussed in more detail in the *BIND Administrator Reference Manual* under `update-policy`.

15.9 DNS Security

DNSSEC, or DNS security, is described in RFC 2535. The tools available for DNSSEC are discussed in the BIND Manual.

A zone considered secure must have one or several zone keys associated with it. These are generated with `dnssec-keygen`, just like the host keys. The DSA encryption algorithm is currently used to generate these keys. The public keys generated should be included in the corresponding zone file with an `$INCLUDE` rule.

With the command `dnssec-signzone`, you can create sets of generated keys (`keyset-files`), transfer them to the parent zone in a secure manner, and sign them. This generates the files to include for each zone in `/etc/named.conf`.

15.10 For More Information

For additional information, refer to the *BIND Administrator Reference Manual* from package `bind-doc`, which is installed under `/usr/share/doc/packages/bind/`. Consider additionally consulting the RFCs referenced by the manual and the manual pages included with BIND. `/usr/share/doc/packages/bind/README.SuSE` contains up-to-date information about BIND in openSUSE.

DHCP

The purpose of the *Dynamic Host Configuration Protocol* (DHCP) is to assign network settings centrally (from a server) rather than configuring them locally on each and every workstation. A host configured to use DHCP does not have control over its own static address. It is enabled to configure itself completely and automatically according to directions from the server. If you use the NetworkManager on the client side, you do not need to configure the client at all. This is useful if you have changing environments and only one interface active at a time. Never use NetworkManager on a machine that runs a DHCP server.

One way to configure a DHCP server is to identify each client using the hardware address of its network card (which should be fixed in most cases), then supply that client with identical settings each time it connects to the server. DHCP can also be configured to assign addresses to each relevant client dynamically from an address pool set up for this purpose. In the latter case, the DHCP server tries to assign the same address to the client each time it receives a request, even over extended periods. This works only if the network does not have more clients than addresses.

DHCP makes life easier for system administrators. Any changes, even bigger ones, related to addresses and the network configuration in general can be implemented centrally by editing the server's configuration file. This is much more convenient than reconfiguring numerous workstations. It is also much easier to integrate machines, particularly new machines, into the network, because they can be given an IP address from the pool. Retrieving the appropriate network settings from a DHCP server is especially useful in case of laptops regularly used in different networks.

In this chapter, the DHCP server will run in the same subnet as the workstations, 192.168.2.0/24 with 192.168.2.1 as gateway. It has the fixed IP address 192.168.2.254 and serves two address ranges, 192.168.2.10 to 192.168.2.20 and 192.168.2.100 192.168.2.200.

A DHCP server supplies not only the IP address and the netmask, but also the host-name, domain name, gateway, and name server addresses for the client to use. In addition to that, DHCP allows a number of other parameters to be configured in a centralized way, for example, a time server from which clients may poll the current time or even a print server.

16.1 Configuring a DHCP Server with YaST

To install a DHCP server, start YaST and select *Software > Software Management*. Choose *Filter > Patterns* and select *DHCP and DNS Server*. Confirm the installation of the dependent packages to finish the installation process.

IMPORTANT: LDAP Support

The YaST DHCP module can be set up to store the server configuration locally (on the host that runs the DHCP server) or to have its configuration data managed by an LDAP server. If you want to use LDAP, set up your LDAP environment before configuring the DHCP server.

For more information about LDAP, see Chapter 4, *LDAP—A Directory Service* (↑Security Guide).

The YaST DHCP module (`yast2-dhcp-server`) allows you to set up your own DHCP server for the local network. The module can run in wizard mode or expert configuration mode.

16.1.1 Initial Configuration (Wizard)

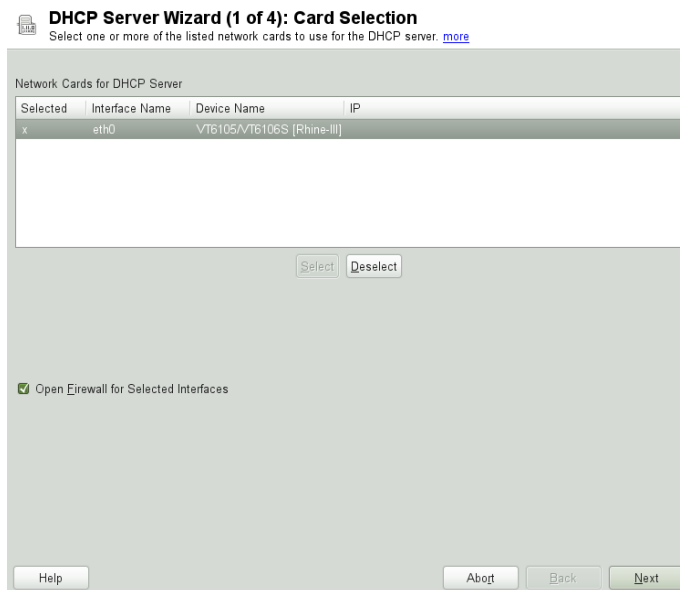
When the module is started for the first time, a wizard starts, prompting you to make a few basic decisions concerning server administration. Completing this initial setup

produces a very basic server configuration that should function in its essential aspects. The expert mode can be used to deal with more advanced configuration tasks.

Card Selection

In the first step, YaST looks for the network interfaces available on your system and displays them in a list. From the list, select the interface to which the DHCP server should listen and click *Select*. After this, select *Open Firewall for Selected Interfaces* to open the firewall for this interface, and click *Next*. See Figure 16.1, “DHCP Server: Card Selection” (page 305).


Figure 16.1: *DHCP Server: Card Selection*



Global Settings

Use the check box to determine whether your DHCP settings should be automatically stored by an LDAP server. In the entry fields, provide the network specifics for all clients the DHCP server should manage. These specifics are the domain name, address of a time server, addresses of the primary and secondary name server, addresses of a print and a WINS server (for a mixed network with both Windows and Linux clients), gateway address, and lease time. See Figure 16.2, “DHCP Server: Global Settings” (page 306).

Figure 16.2: *DHCP Server: Global Settings*

 **DHCP Server Wizard (2 of 4): Global Settings**
To store the DHCP configuration in LDAP, enable LDAP Support. [more](#)

☐ LDAP Support

DHCP Server Name (optional)

Domain Name:

Primary Name Server IP:

Secondary Name Server IP:

Default Gateway (Router)

NTP Time Server:

Print Server:


WINS Server:

Default Lease Time: Units:

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 16.3, “DHCP Server: Dynamic DHCP” (page 307).

Figure 16.3: *DHCP Server: Dynamic DHCP*

 **DHCP Server Wizard (3 of 4): Dynamic DHCP**
Here you can view the information about the current subnet, such as its address, netmask and minimum and maximum IP addresses available for the clients. [more](#)

Subnet Information	
Current Network:	Current Netmask:
192.168.2.0	255.255.255.0
Minimum IP Address:	Maximum IP Address:
192.168.2.1	192.168.2.254

IP Address Range	
First IP Address:	Last IP Address:
192.168.2.100	192.168.2.139

☐ Allow Dynamic BOOTP

Lease Time	
Default:	Maximum:
4	2
Units: Hours	Units: Days

[Synchronize DNS Server...](#)

[Help](#) [About](#) [Back](#) [Next](#)

Finishing the Configuration and Setting the Start Mode

After the third part of the configuration wizard, a last dialog is shown in which you can define how the DHCP server should be started. Here, specify whether to start the DHCP server automatically when the system is booted or manually when needed (for example, for testing purposes). Click *Finish* to complete the configuration of the server. See Figure 16.4, “DHCP Server: Start-Up” (page 307).

Figure 16.4: *DHCP Server: Start-Up*

 **DHCP Server Wizard (4 of 4): Start-Up**
To start the service every time your computer is booted, set When Booting. [more](#)

Service Start

☐ When Booting

☒ Manually

[DHCP Server Expert Configuration...](#)

[Help](#) [About](#) [Back](#) [Finish](#)

16.2 DHCP Software Packages

Both the DHCP server and the DHCP clients are available for openSUSE. The DHCP server available is `dhcpcd` (published by the Internet Systems Consortium). On the client side, choose between two different DHCP client programs: `dhcpc-client` (also from ISC) and the DHCP client daemon in the `dhcpcd` package.

openSUSE installs `dhcpcd` by default. The program is very easy to handle and is launched automatically on each system boot to watch for a DHCP server. It does not need a configuration file to do its job and works out of the box in most standard setups. For more complex situations, use the ISC `dhcpc-client`, which is controlled by means of the configuration file `/etc/dhclient.conf`.

16.3 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/dhcpd.conf`. By changing the parameters and values in this file, a system administrator can influence the program's behavior in numerous ways. Look at the basic sample `/etc/dhcpd.conf` file in Example 16.1, “The Configuration File `/etc/dhcpd.conf`” (page 308).

Example 16.1: *The Configuration File `/etc/dhcpd.conf`*

```
default-lease-time 600;          # 10 minutes
max-lease-time 7200;             # 2  hours

option domain-name "example.com";
option domain-name-servers 192.168.1.116;
option broadcast-address 192.168.2.255;
option routers 192.168.2.1;
option subnet-mask 255.255.255.0;

subnet 192.168.2.0 netmask 255.255.255.0
{
    range 192.168.2.10 192.168.2.20;
    range 192.168.2.100 192.168.2.200;
}
```

This simple configuration file should be sufficient to get the DHCP server to assign IP addresses in the network. Make sure that a semicolon is inserted at the end of each line, because otherwise `dhcpcd` is not started.

The sample file can be divided into three sections. The first one defines how many seconds an IP address is leased to a requesting client by default (`default-lease-`

time) before it should apply for renewal. This section also includes a statement of the maximum period for which a machine may keep an IP address assigned by the DHCP server without applying for renewal (`max-lease-time`).

In the second part, some basic network parameters are defined on a global level:

- The line `option domain-name` defines the default domain of your network.
- With the entry `option domain-name-servers`, specify up to three values for the DNS servers used to resolve IP addresses into hostnames and vice versa. Ideally, configure a name server on your machine or somewhere else in your network before setting up DHCP. That name server should also define a hostname for each dynamic address and vice versa. To learn how to configure your own name server, read Chapter 15, *The Domain Name System* (page 279).
- The line `option broadcast-address` defines the broadcast address the requesting client should use.
- With `option routers`, set where the server should send data packets that cannot be delivered to a host on the local network (according to the source and target host address and the subnet mask provided). In most cases, especially in smaller networks, this router is identical to the Internet gateway.
- With `option subnet-mask`, specify the netmask assigned to clients.

The last section of the file defines a network, including a subnet mask. To finish, specify the address range that the DHCP daemon should use to assign IP addresses to interested clients. In Example 16.1, “The Configuration File / `etc/dhcpd.conf`” (page 308), clients may be given any address between `192.168.2.10` and `192.168.2.20` as well as `192.168.2.100` and `192.168.2.200`.

After editing these few lines, you should be able to activate the DHCP daemon with the command `rcdhcpd start`. It will be ready for use immediately. Use the command `rcdhcpd check-syntax` to perform a brief syntax check. If you encounter any unexpected problems with your configuration (the server aborts with an error or does not return `done` on start), you should be able to find out what has gone wrong by looking for information either in the main system log `/var/log/messages` or on console 10 (Ctrl + Alt + F10).

On a default openSUSE system, the DHCP daemon is started in a chroot environment for security reasons. The configuration files must be copied to the chroot environment

so the daemon can find them. Normally, there is no need to worry about this because the command `redhcpd start` automatically copies the files.

16.3.1 Clients with Fixed IP Addresses

DHCP can also be used to assign a predefined, static address to a specific client. Addresses assigned explicitly always take priority over dynamic addresses from the pool. A static address never expires in the way a dynamic address would, for example, if there were not enough addresses available and the server needed to redistribute them among clients.

To identify a client configured with a static address, `dhcpd` uses the hardware address (which is a globally unique, fixed numerical code consisting of six octet pairs) for the identification of all network devices (for example, `00:30:6E:08:EC:80`). If the respective lines, like the ones in Example 16.2, “Additions to the Configuration File” (page 310), are added to the configuration file of Example 16.1, “The Configuration File `/etc/dhcpd.conf`” (page 308), the DHCP daemon always assigns the same set of data to the corresponding client.

Example 16.2: *Additions to the Configuration File*

```
host jupiter {  
    hardware ethernet 00:30:6E:08:EC:80;  
    fixed-address 192.168.2.100;  
}
```

The name of the respective client (host *hostname*, here `jupiter`) is entered in the first line and the MAC address in the second line. On Linux hosts, find the MAC address with the command `ip link show` followed by the network device (for example, `eth0`). The output should contain something like

```
link/ether 00:30:6E:08:EC:80
```

In the preceding example, a client with a network card having the MAC address `00:30:6E:08:EC:80` is assigned the IP address `192.168.2.100` and the hostname `jupiter` automatically. The type of hardware to enter is `ethernet` in nearly all cases, although `token-ring`, which is often found on IBM systems, is also supported.

16.3.2 The openSUSE Version

To improve security, the openSUSE version of the ISC's DHCP server comes with the non-root/chroot patch by Ari Edelkind applied. This enables `dhcpd` to run with

the user ID `nobody` and run in a chroot environment (`/var/lib/dhcp`). To make this possible, the configuration file `dhcpd.conf` must be located in `/var/lib/dhcp/etc`. The init script automatically copies the file to this directory when starting.

Control the server's behavior regarding this feature by means of entries in the file `/etc/sysconfig/dhcpd`. To run `dhcpd` without the chroot environment, set the variable `DHCPD_RUN_CHROOTED` in `/etc/sysconfig/dhcpd` to “no”.

To enable `dhcpd` to resolve hostnames even from within the chroot environment, some other configuration files must be copied as well:

- `/etc/localtime`
- `/etc/host.conf`
- `/etc/hosts`
- `/etc/resolv.conf`

These files are copied to `/var/lib/dhcp/etc/` when starting the init script. Take these copies into account for any changes that they require if they are dynamically modified by scripts like `/etc/ppp/ip-up`. However, there should be no need to worry about this if the configuration file only specifies IP addresses (instead of hostnames).

If your configuration includes additional files that should be copied into the chroot environment, set these under the variable `DHCPD_CONF_INCLUDE_FILES` in the file `/etc/sysconfig/dhcpd`. To ensure that the DHCP logging facility keeps working even after a restart of the `syslog-ng` daemon, there is an additional entry `SYSLOGD_ADDITIONAL_SOCKET_DHCP` in the file `/etc/sysconfig/syslog`.

16.4 For More Information

More information about DHCP is available at the Web site of the *Internet Systems Consortium* (<http://www.isc.org/products/DHCP/>). Information is also available in the `dhcpd`, `dhcpd.conf`, `dhcpd.leases`, and `dhcp-options` man pages.

Time Synchronization with NTP

The NTP (network time protocol) mechanism is a protocol for synchronizing the system time over the network. First, a machine can obtain the time from a server that is a reliable time source. Second, a machine can itself act as a time source for other computers in the network. The goal is twofold—maintaining the absolute time and synchronizing the system time of all machines within a network.

Maintaining an exact system time is important in many situations. The built-in hardware (BIOS) clock does often not meet the requirements of applications such as databases or clusters. Manual correction of the system time would lead to severe problems because, for example, a backward leap can cause malfunction of critical applications. Within a network, it is usually necessary to synchronize the system time of all machines, but manual time adjustment is a bad approach. NTP provides a mechanism to solve these problems. The NTP service continuously adjusts the system time with the help of reliable time servers in the network. It further enables the management of local reference clocks, such as radio-controlled clocks.

17.1 Configuring an NTP Client with YaST

The NTP daemon (`ntpd`) coming with the `ntp` package is preset to use the local computer clock as a time reference. Using the (BIOS) clock, however, only serves as a fallback for cases where no time source of better precision is available. YaST facilitates the configuration of an NTP client.

17.1.1 Basic Configuration

The YaST NTP client configuration (*Network Services > NTP Configuration*) consists of tabs. Set the start mode of `ntpd` and the server to query on the *General Settings* tab.

Figure 17.1: *Advanced NTP Configuration: General Settings*

The screenshot shows the 'Advanced NTP Configuration' window with the 'General Settings' tab selected. The window title is 'Advanced NTP Configuration' with a subtitle 'Select whether to start the NTP daemon now and on every system boot. [more](#)'. The 'Start NTP Daemon' section has three radio buttons: 'Only Manually', 'Synchronize without Daemon', and 'Now and On Boot' (which is selected). Below this is a 'Runtime Configuration Policy' section with a dropdown set to 'Auto' and a text field. The 'Interval of the Synchronization in Minutes' is set to 30. A table lists synchronization sources: 'Undisciplined Local Clock (LOCAL)' and 'Server 1.opensuse.pool.ntp.org' (which is highlighted). At the bottom are buttons for 'Add', 'Edit', 'Delete', 'Display Log...', 'Help', 'Cancel', and 'OK'.

Advanced NTP Configuration
Select whether to start the NTP daemon now and on every system boot. [more](#)

General Settings Security Settings

Start NTP Daemon

☐ Only Manually

☐ Synchronize without Daemon

☒ Now and On Boot

Runtime Configuration Policy: Custom Policy:

Auto

Interval of the Synchronization in Minutes:

30

Synchronization Type	Address
Undisciplined Local Clock (LOCAL)	
Server	1.opensuse.pool.ntp.org

Add Edit Delete Display Log...

Help Cancel OK

Only Manually

Select *Only Manually*, if you want to configure everything on your own.

Synchronize without Daemon

On laptops and other machines that suspend automatically, select *Synchronize without Daemon*. Using this mode, YaST does not start `ntpd` on the local machine for synchronizing. Instead YaST creates a crontab entry (`/etc/cron.d/novell.ntp-synchronize`) that checks the time with the time server as specified in the *Interval of the Synchronization in Minutes* field. For more information about cron, see Section 11.1.2, “The cron Package” (page 188).

Now and On Boot

Select *Now and On Boot* to start `ntpd` automatically when the system is booted. Either of `0.opensuse.pool.ntp.org`, `1.opensuse.pool.ntp.org`, `2.opensuse.pool.ntp.org`, or `3.opensuse.pool.ntp.org` is pre-selected.

17.1.2 Changing Basic Configuration

The servers and other time sources for the client to query are listed in the lower part of the *General Settings* tab. Modify this list as needed with *Add*, *Edit*, and *Delete*. *Display Log* provides the possibility to view the log files of your client.

Click *Add* to add a new source of time information. In the following dialog, select the type of source with which the time synchronization should be made. The following options are available:

Figure 17.2: *YaST: NTP Server*



Server

In the pull-down *Select* list (see Figure 17.2, “YaST: NTP Server” (page 315), determine whether to set up time synchronization using a time server from your

local network (*Local NTP Server*) or an Internet-based time server that takes care of your time zone (*Public NTP Server*). For a local time server, click *Lookup* to start an SLP query for available time servers in your network. Select the most suitable time server from the list of search results and exit the dialog with *OK*. For a public time server, select your country (time zone) and a suitable server from the list under *Public NTP Server* then exit the dialog with *OK*. In the main dialog, test the availability of the selected server with *Test*. *Options* allows you to specify additional options for `ntpd`.

Using *Access Control Options*, you can restrict the actions that the remote computer can perform with the daemon running on your computer. This field is enabled only after checking *Restrict NTP Service to Configured Servers Only* on the *Security Settings* tab (see Figure 17.3, “Advanced NTP Configuration: Security Settings” (page 317)). The options correspond to the `restrict` clauses in `/etc/ntp.conf`. For example, `nomodify notrap noquery` disallows the server to modify NTP settings of your computer and to use the trap facility (a remote event logging feature) of your NTP daemon. Using these restrictions is recommended for servers out of your control (for example, on the Internet).

Refer to `/usr/share/doc/packages/ntp-doc` (part of the `ntp-doc` package) for detailed information.

Peer

A peer is a machine to which a symmetric relationship is established: it acts both as a time server and as a client. To use a peer in the same network instead of a server, enter the address of the system. The rest of the dialog is identical to the *Server* dialog.

Radio Clock

To use a radio clock in your system for the time synchronization, enter the clock type, unit number, device name, and other options in this dialog. Click *Driver Calibration* to fine-tune the driver. Detailed information about the operation of a local radio clock is available in `/usr/share/doc/packages/ntp-doc/refclock.html`.

Outgoing Broadcast

Time information and queries can also be transmitted by broadcast in the network. In this dialog, enter the address to which such broadcasts should be sent. Do not activate broadcasting unless you have a reliable time source like a radio controlled clock.

Incoming Broadcast

If you want your client to receive its information via broadcast, enter the address from which the respective packets should be accepted in this fields.

Figure 17.3: *Advanced NTP Configuration: Security Settings*



In the *Security Settings* tab (see Figure 17.3, “Advanced NTP Configuration: Security Settings” (page 317)), determine whether `ntpd` should be started in a chroot jail. By default, *Run NTP Daemon in Chroot Jail* is activated. This increases the security in the event of an attack over `ntpd`, as it prevents the attacker from compromising the entire system.

Restrict NTP Service to Configured Servers Only increases the security of your system by disallowing remote computers to view and modify NTP settings of your computer and to use the trap facility for remote event logging. Once enabled, these restrictions apply to all remote computers, unless you override the access control options for individual computers in the list of time sources in the *General Settings* tab. For all other remote computers, only querying for local time is allowed.

Enable *Open Port in Firewall* if `SuSEfirewall2` is active (which it is by default). If you leave the port closed, it is not possible to establish a connection to the time server.

17.2 Manually Configuring ntp in the Network

The easiest way to use a time server in the network is to set server parameters. For example, if a time server called `ntp.example.com` is reachable from the network, add its name to the file `/etc/ntp.conf` by adding the following line:

```
server ntp.example.com
```

To add more time servers, insert additional lines with the keyword `server`. After initializing `ntpd` with the command `rcntp start`, it takes about one hour until the time is stabilized and the drift file for correcting the local computer clock is created. With the drift file, the systematic error of the hardware clock can be computed as soon as the computer is powered on. The correction is used immediately, resulting in a higher stability of the system time.

There are two possible ways to use the NTP mechanism as a client: First, the client can query the time from a known server in regular intervals. With many clients, this approach can cause a high load on the server. Second, the client can wait for NTP broadcasts sent out by broadcast time servers in the network. This approach has the disadvantage that the quality of the server is unknown and a server sending out wrong information can cause severe problems.

If the time is obtained via broadcast, you do not need the server name. In this case, enter the line `broadcastclient` in the configuration file `/etc/ntp.conf`. To use one or more known time servers exclusively, enter their names in the line starting with `servers`.

17.3 Dynamic Time Synchronization at Runtime

If the system boots without network connection, `ntpd` starts up, but it cannot resolve DNS names of the time servers set in the configuration file. This can happen if you use Network Manager with an encrypted WLAN.

If you want `ntpd` to resolve DNS names at runtime, you must set the `dynamic` option. Then, when the network is established some time after booting, `ntpd` looks up the names again and can reach the time servers to get the time.

Manually edit `/etc/ntp.conf` and add `dynamic` to one or more server entries:

```
server ntp.example.com dynamic
```

Or use YaST and proceed as follows:

- 1 In YaST click *Network Services > NTP Configuration*.
- 2 Select the server you want to configure. Then click *Edit*.
- 3 Activate the *Options* field and add `dynamic`. Separate it with a space, if there are already other options entered.
- 4 Click *Ok* to close the edit dialog. Repeat the previous step to change all servers as wanted.
- 5 Finally click *Ok* to save the settings.

17.4 Setting Up a Local Reference Clock

The software package `ntp` contains drivers for connecting local reference clocks. A list of supported clocks is available in the `ntp-doc` package in the file `/usr/share/doc/packages/ntp-doc/refclock.html`. Every driver is associated with a number. In `ntp`, the actual configuration takes place by means of pseudo IP addresses. The clocks are entered in the file `/etc/ntp.conf` as though they existed in the network. For this purpose, they are assigned special IP addresses in the form `127.127.t.u`. Here, *t* stands for the type of the clock and determines which driver is used and *u* for the unit, which determines the interface used.

Normally, the individual drivers have special parameters that describe configuration details. The file `/usr/share/doc/packages/ntp-doc/drivers/driverNN.html` (where *NN* is the number of the driver) provides information about the particular type of clock. For example, the “type 8” clock (radio clock over serial interface) requires an additional mode that specifies the clock more precisely. The Conrad DCF77 receiver module, for example, has mode 5. To use this clock as a preferred reference, specify the keyword `prefer`. The complete `server` line for a Conrad DCF77 receiver module would be:

```
server 127.127.8.0 mode 5 prefer
```

Other clocks follow the same pattern. Following the installation of the `ntp-doc` package, the documentation for `ntp` is available in the directory `/usr/share/doc/packages/ntp-doc`. The file `/usr/share/doc/packages/ntp-doc/refclock.html` provides links to the driver pages describing the driver parameters.

Sharing File Systems with NFS

Distributing and sharing file systems over a network is a common task in corporate environments. The well-proven network file system (*NFS*) works together with *NIS*, the yellow pages protocol. For a more secure protocol that works together with *LDAP* and may also use Kerberos, check *NFSv4*.

NFS with NIS makes a network transparent to the user. With NFS, it is possible to distribute arbitrary file systems over the network. With an appropriate setup, users always find themselves in the same environment regardless of the terminal they currently use.

IMPORTANT: Need for DNS

In principle, all exports can be made using IP addresses only. To avoid timeouts, you need a working DNS system. DNS is necessary at least for logging purposes, because the mountd daemon does reverse lookups.

18.1 Terminology

The following are terms used in the YaST module.

Exports

A directory *exported* by an NFS server, which clients can integrate it into their system.

NFS Client

The NFS client is a system that uses NFS services from an NFS server over the Network File System protocol. The TCP/IP protocol is already integrated into the Linux kernel; there is no need to install any additional software.

NFS Server

The NFS server provides NFS services to clients. A running server depends on the following daemons: `nfsd` (worker), `idmapd` (user and group name mappings to IDs and vice versa), `statd` (file locking), and `mountd` (mount requests).

18.2 Installing NFS Server

The NFS server software is not part of the default installation. If you configure an NFS Server as described in Section 18.3, “Configuring NFS Server” (page 322) you will automatically be prompted to install the needed packages. Alternatively, install the package `nfs-kernel-server` with YaST or zypper.

Like NIS, NFS is a client/server system. However, a machine can be both—it can supply file systems over the network (export) and mount file systems from other hosts (import).

18.3 Configuring NFS Server

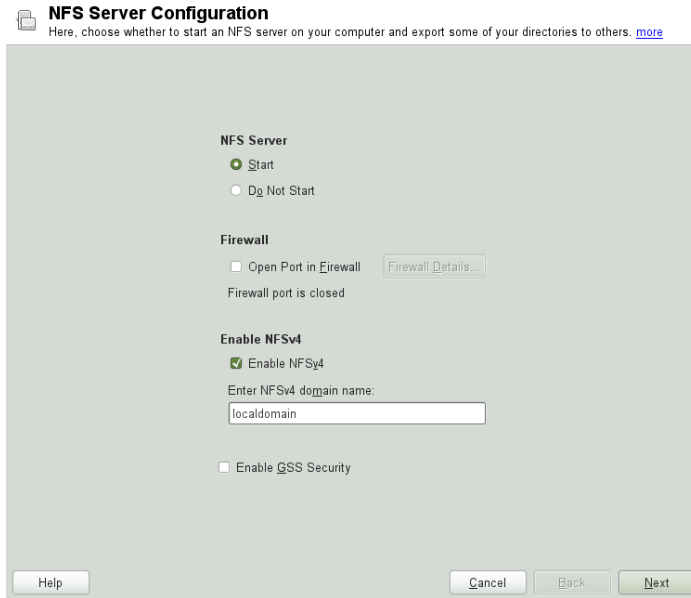
Configuring an NFS server can be done either through YaST or manually. For authentication, NFS can also be combined with Kerberos.

18.3.1 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. The server can also provide applications to all members of a group without installing the applications locally on each and every host. To set up such a server, proceed as follows:

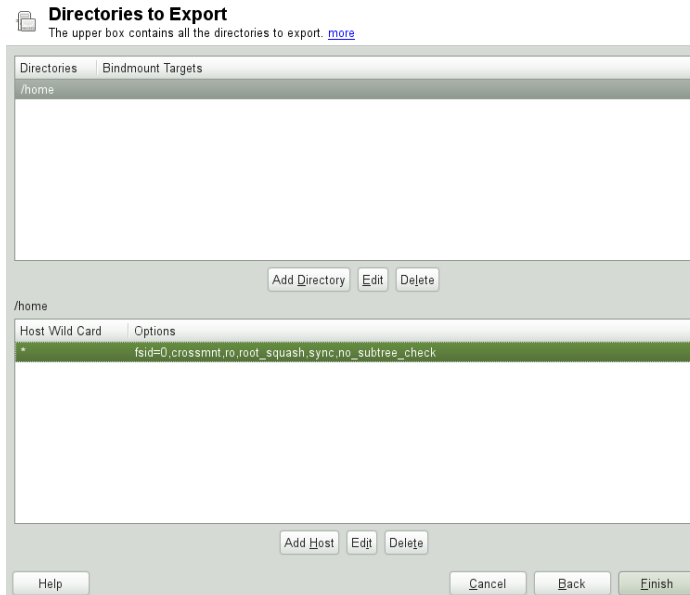
- 1 Start YaST and select *Network Services > NFS Server*; see Figure 18.1, “NFS Server Configuration Tool” (page 323).

Figure 18.1: *NFS Server Configuration Tool*



- 2 Activate the *Start* radio button and enter the *NFSv4 Domain Name*.
- 3 Click *Enable GSS Security* if you need secure access to the server. A prerequisite for this is to have Kerberos installed on your domain and to have both the server and the clients kerberized. Click *Next*.
- 4 Enter the directories to export in the upper text field. Below, enter the hosts that should have access to them. This dialog is shown in Figure 18.2, “Configuring an NFS Server with YaST” (page 324).

Figure 18.2: *Configuring an NFS Server with YaST*



The figure shows the scenario where NFSv4 is enabled in the previous dialog. Bindmount Targets is shown in the right pane. For more details, click *Help*. In the lower half of the dialog, there are four options that can be set for each host: single host, netgroups, wildcards, and IP networks. For a more thorough explanation of these options, refer to the `exports` man page.

- 5 Click *Finish* to complete the configuration.

IMPORTANT: Automatic Firewall Configuration

If a firewall is active on your system (SuSEfirewall2), YaST adapts its configuration for the NFS server by enabling the `nfs` service when *Open Ports in Firewall* is selected.

18.3.1.1 Exporting for NFSv4 Clients

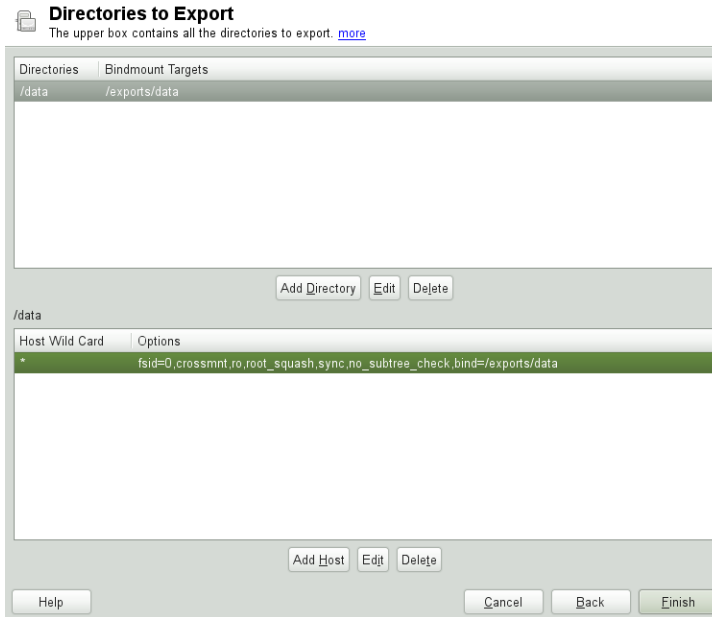
Activate *Enable NFSv4* to support NFSv4 clients. Clients with NFSv3 can still access the server's exported directories if they are exported appropriately. This is explained in detail in Section 18.3.1.3, “Coexisting v3 and v4 Exports” (page 328).

After activating NFSv4, enter an appropriate domain name. Make sure the name is the same as the one in the `/etc/ldapd.conf` file of any NFSv4 client that accesses this particular server. This parameter is for the `ldapd` service that is required for NFSv4 support (on both server and client). Leave it as `localdomain` (the default) if you do not have special requirements. For more information, see the links in Section 18.5, “For More Information” (page 334).

Click *Next*. The dialog that follows has two sections. The upper half consists of two columns named *Directories* and *Bind Mount Targets*. *Directories* is a directly editable column that lists the directories to export.

For a fixed set of clients, there are two types of directories that can be exported—directories that act as pseudo root file systems and those that are bound to some subdirectory of the pseudo file system. This pseudo file system acts as a base point under which all file systems exported for the same client set take their place. For a client or set of clients, only one directory on the server can be configured as pseudo root for export. For this client, export multiple directories by binding them to some existing subdirectory in the pseudo root.

Figure 18.3: *Exporting Directories with NFSv4*



In the lower half of the dialog, enter the client (wild card) and export options for a particular directory. After adding a directory in the upper half, another dialog for entering the client information and options pops up automatically. After that, to add a new client or a set of clients, click *Add Host*.

In the small dialog that opens, enter the host wild card. There are four possible types of host wild cards that can be set for each host: a single host (name or IP address), netgroups, wild cards (such as `*` indicating all machines can access the server), and IP networks. Then, in *Options*, include `fsid=0` in the comma-separated list of options to configure the directory as pseudo root. If this directory needs to be bound to another directory under an already configured pseudo root, make sure that a target bind path is given in the option list with `bind=/target/path`.

For example, suppose that the directory `/exports` is chosen as the pseudo root directory for all the clients that can access the server. Then add this in the upper half and make sure that the options entered for this directory include `fsid=0`. If there is another directory, `/data`, that also needs to be NFSv4 exported, add this directory to the upper half. While entering options for this, make sure that `bind=/exports/data` is in the list and that `/exports/data` is an already existing subdi-

rectory of `/exports`. Any change in the option `bind=/target/path`, whether addition, deletion, or change in value, is reflected in *Bindmount Targets*. This column is not a directly editable column, but instead summarizes directories and their nature. After all information is provided, click *Finish* to complete the configuration. The service will become available immediately.

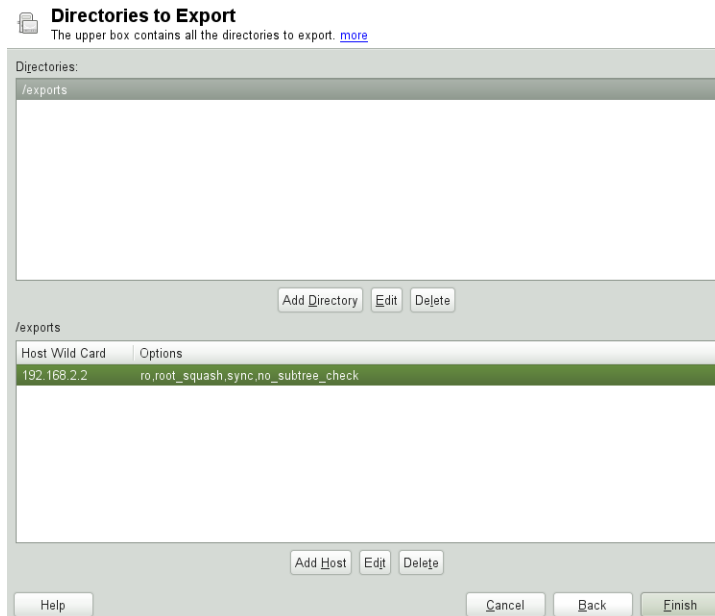
18.3.1.2 NFSv3 and NFSv2 Exports

Make sure that *Enable NFSv4* is not checked in the initial dialog before clicking *Next*.

The next dialog has two parts. In the upper text field, enter the directories to export. Below, enter the hosts that should have access to them. There are four types of host wild cards that can be set for each host: a single host (name or IP address), netgroups, wild cards (such as `*` indicating all machines can access the server), and IP networks.

This dialog is shown in Figure 18.4, “Exporting Directories with NFSv2 and v3” (page 327). Find a more thorough explanation of these options in `man exports`. Click *Finish* to complete the configuration.

Figure 18.4: *Exporting Directories with NFSv2 and v3*



18.3.1.3 Coexisting v3 and v4 Exports

NFSv3 and NFSv4 exports can coexist on a server. After enabling the support for NFSv4 in the initial configuration dialog, those exports for which `fsid=0` and `bind=/target/path` are not included in the option list are considered v3 exports. Consider the example in Figure 18.2, “Configuring an NFS Server with YaST” (page 324). If you add another directory, such as `/data2`, using *Add Directory* then in the corresponding options list do not mention either `fsid=0` or `bind=/target/path`, this export acts as a v3 export.

IMPORTANT

Automatic Firewall Configuration

If `SuSEfirewall2` is active on your system, YaST adapts its configuration for the NFS server by enabling the `nfs` service when *Open Ports in Firewall* is selected.

18.3.2 Exporting File Systems Manually

The configuration files for the NFS export service are `/etc/exports` and `/etc/sysconfig/nfs`. In addition to these files, `/etc/idmapd.conf` is needed for the NFSv4 server configuration. To start or restart the services, run the command `rcnfsserver restart`. This also starts the `rpc.idmapd` if NFSv4 is configured in `/etc/sysconfig/nfs`. The NFS server depends on a running RPC portmapper. Therefore, also start or restart the portmapper service with `rcrpcbind restart`.

18.3.2.1 Exporting File Systems with NFSv4

NFSv4 is the latest version of NFS protocol available on openSUSE. Configuring directories for export with NFSv4 differs slightly from previous NFS versions.

`/etc/exports`

The `/etc/exports` file contains a list of entries. Each entry indicates a directory that is shared and how it is shared. A typical entry in `/etc/exports` consists of:

```
/shared/directory    host(option_list)
```

For example:

```
/export    192.168.1.2(rw,fsid=0,sync,crossmnt)
/export/data    192.168.1.2(rw,bind=/data,sync)
```

Here the IP address `192.168.1.2` is used to identify the allowed client. You can also use the name of the host, a wild card indicating a set of hosts (`*.abc.com`, `*`, etc.), or `netgroups` (`@my-hosts`).

The directory which specifies `fsid=0` is special. It is the root of the filesystem that is exported, sometimes referred to as the pseudo root filesystem. This directory must also have the `crossmnt` for correct operation with NFSv4. All other directories exported via NFSv4 must be mounted below this point. If you want to export a directory that is not within the exported root, it needs to be bound into the exported tree. This can be done using the `bind=` syntax.

In the example above, `/data` is not within `/export`, so we export `/export/data`, and specify that the `/data` directory should be bound to that name. The directory `/export/data` must exist and should normally be empty.

When clients mount from this server, they just mount `servername:/` rather than `servername:/export`. It is not necessary to mount `servername:/data`, because it will automatically appear beneath wherever `servername:/` was mounted.

/etc/sysconfig/nfs

The `/etc/sysconfig/nfs` file contains a few parameters that determine NFSv4 server daemon behavior. It is important to set the parameter `NFS4_SUPPORT` to `yes`. `NFS4_SUPPORT` determines whether the NFS server supports NFSv4 exports and clients.

/etc/idmapd.conf

Every user on a Linux machine has a name and ID. `idmapd` does the name-to-ID mapping for NFSv4 requests to the server and replies to the client. It must be running on both server and client for NFSv4, because NFSv4 uses only names for its communication.

Make sure that there is a uniform way in which usernames and IDs (`uid`) are assigned to users across machines that might probably be sharing file systems using NFS. This

can be achieved by using NIS, LDAP, or any uniform domain authentication mechanism in your domain.

The parameter `Domain` must be set the same for both, client and server in the `/etc/idmapd.conf` file. If you are not sure, leave the domain as `localdomain` in the server and client files. A sample configuration file looks like the following:

```
[General]

Verbosity = 0
Pipefs-Directory = /var/lib/nfs/rpc_pipefs
Domain = localdomain

[Mapping]

Nobody-User = nobody
Nobody-Group = nobody
```

For further reference, read the man page of `idmapd` and `idmapd.conf`; `man idmapd`, `man idmapd.conf`.

Starting and Stopping Services

After changing `/etc/exports` or `/etc/sysconfig/nfs`, start or restart the NFS server service with `rcnfsserver restart`. After changing `/etc/idmapd.conf`, reload the configuration file with the command `killall -HUP rpc.idmapd`.

If the NFS service needs to start at boot time, run the command `chkconfig nfs-server on`.

18.3.2.2 Exporting File Systems with NFSv2 and NFSv3

This section is specific to NFSv3 and NFSv2 exports. Refer to Section 18.3.1.1, “Exporting for NFSv4 Clients” (page 325) for exporting with NFSv4.

Exporting file systems with NFS involves two configuration files: `/etc/exports` and `/etc/sysconfig/nfs`. A typical `/etc/exports` file entry is in the format:

```
/shared/directory    host(list_of_options)
```

For example:

```
/export 192.168.1.2(rw, sync)
```

Here, the directory `/export` is shared with the host `192.168.1.2` with the option list `rw, sync`. This IP address can be replaced with a client name or set of clients using a wild card (such as `*.abc.com`) or even `netgroups`.

For a detailed explanation of all options and their meaning, refer to the man page of `exports` (`man exports`).

After changing `/etc/exports` or `/etc/sysconfig/nfs`, start or restart the NFS server using the command `rcnfsserver restart`.

18.3.3 NFS with Kerberos

To use Kerberos authentication for NFS, GSS security must be enabled. Select *Enable GSS Security* in the initial YaST NFS Server dialog. You must have a working Kerberos server to use this feature. YaST does not set up the server but just uses the provided functionality. If you want to use Kerberos authentication in addition to the YaST configuration, complete at least the following steps before running the NFS configuration:

- 1 Make sure that both the server and the client are in the same Kerberos domain. They must access the same KDC (Key Distribution Center) server and share their `krb5.keytab` file (the default location on any machine is `/etc/krb5.keytab`). For more information about Kerberos, see Chapter 6, *Network Authentication with Kerberos* (↑Security Guide).
- 2 Start the `gssd` service on the client with `rcgssd start`.
- 3 Start the `svcgssd` service on the server with `rcsvcgssd start`.

For more information about configuring kerberized NFS, refer to the links in Section 18.5, “For More Information” (page 334).

18.4 Configuring Clients

To configure your host as an NFS client, you do not need to install additional software. All needed packages are installed by default.

18.4.1 Importing File Systems with YaST

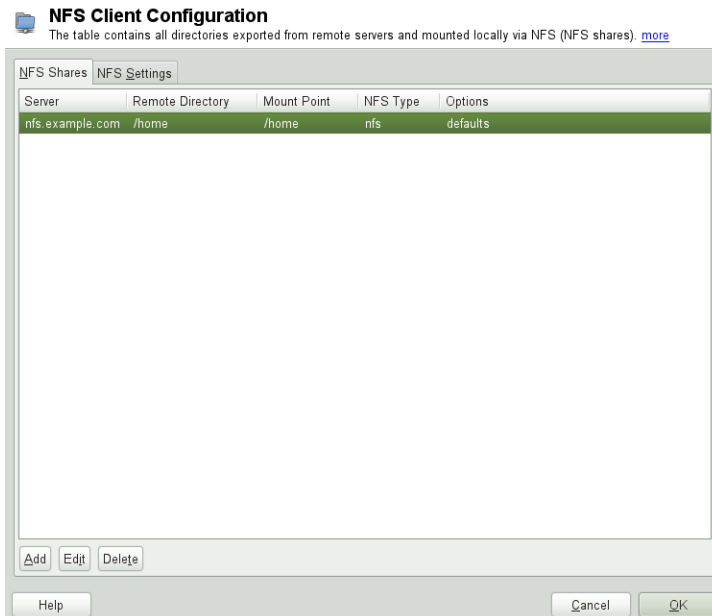
Authorized users can mount NFS directories from an NFS server into the local file tree using the YaST NFS client module. Click on *Add* and enter the hostname of the NFS server, the directory to import, and the mount point at which to mount this directory locally. The changes will take effect after clicking *Finish* in the first dialog.

In the *NFS Settings* tab, enable *Open Port in Firewall* to allow access to the service from remote computers. The firewall status is displayed next to the check box. When using NFSv4, make sure that the checkbox *Enable NFSv4* is selected and that the *NFSv4 Domain Name* contains the same value as used by the NFSv4 server. The default domain is `localdomain`.

Click *OK* to save your changes. See Figure 18.5, “NFS Client Configuration with YaST” (page 332).

The configuration is written to `/etc/fstab` and the specified file systems are mounted. When you start the YaST configuration client at a later time, it also reads the existing configuration from this file.

Figure 18.5: *NFS Client Configuration with YaST*



18.4.2 Importing File Systems Manually

The prerequisite for importing file systems manually from an NFS server is a running RPC port mapper. Start it by entering `rpcbind start` as root. Then remote file systems can be mounted in the file system like local partitions using `mount`:

```
mount host:remote-path local-path
```

To import user directories from the `nfs.example.com` machine, for example, use:

```
mount nfs.example.com:/home /home
```

18.4.2.1 Using the Automount Service

The `autofs` daemon can be used to mount remote file systems automatically. Add the following entry in the your `/etc/auto.master` file:

```
/nfsmounts /etc/auto.nfs
```

Now the `/nfsmounts` directory acts as the root for all the NFS mounts on the client if the `auto.nfs` file is filled appropriately. The name `auto.nfs` is chosen for the sake of convenience—you can choose any name. In `auto.nfs` add entries for all the NFS mounts as follows:

```
localdata -fstype=nfs server1:/data  
nfs4mount -fstype=nfs4 server2:/
```

Activate the settings with `rcautofs start` as root. In this example, `/nfsmounts/localdata`, the `/data` directory of `server1`, is mounted with NFS and `/nfsmounts/nfs4mount` from `server2` is mounted with NFSv4.

If the `/etc/auto.master` file is edited while the service `autofs` is running, the automounter must be restarted for the changes to take effect with `rcautofs restart`.

18.4.2.2 Manually Editing `/etc/fstab`

A typical NFSv3 mount entry in `/etc/fstab` looks like this:

```
nfs.example.com:/data /local/path nfs rw,noauto 0 0
```

NFSv4 mounts may also be added to the `/etc/fstab` file. For these mounts, use `nfs4` instead of `nfs` in the third column and make sure that the remote file system is

given as `/` after the `nfs.example.com:` in the first column. A sample line for an NFSv4 mount in `/etc/fstab` looks like this:

```
nfs.example.com:/ /local/pathv4 nfs4 rw,noauto 0 0
```

The `noauto` option prevents the file system from being mounted automatically at start up. If you want to mount the respective file system manually, it is possible to shorten the mount command specifying the mount point only:

```
mount /local/path
```

Note, that if you do not enter the `noauto` option, the initialization scripts of the system will handle the mount of those file systems at start up.

18.5 For More Information

In addition to the man pages of `exports`, `nfs`, and `mount`, information about configuring an NFS server and client is available in `/usr/share/doc/packages/nfsidmap/README`. For further documentation online refer to the following Web sites:

- Find the detailed technical documentation online at SourceForge [<http://nfs.sourceforge.net/>].
- For instructions for setting up kerberized NFS, refer to NFS Version 4 Open Source Reference Implementation [<http://www.citi.umich.edu/projects/nfsv4/linux/krb5-setup.html>].
- If you have questions on NFSv4, refer to the Linux NFSv4 FAQ [<http://www.citi.umich.edu/projects/nfsv4/linux/faq/>].

Samba

Using Samba, a Unix machine can be configured as a file and print server for Mac OS X, Windows, and OS/2 machines. Samba has developed into a fully-fledged and rather complex product. Configure Samba with YaST, SWAT (a Web interface), or by editing the configuration file manually.

19.1 Terminology

The following are some terms used in Samba documentation and in the YaST module.

SMB protocol

Samba uses the SMB (server message block) protocol that is based on the NetBIOS services. Microsoft released the protocol so other software manufacturers could establish connections to a Microsoft domain network. With Samba, the SMB protocol works on top of the TCP/IP protocol, so the TCP/IP protocol must be installed on all clients.

CIFS protocol

CIFS (common Internet file system) protocol is another protocol supported by Samba. CIFS defines a standard remote file system access protocol for use over the network, enabling groups of users to work together and share documents across the network.

NetBIOS

NetBIOS is a software interface (API) designed for communication between machines providing a name service. It enables machines connected to the network

to reserve names for themselves. After reservation, these machines can be addressed by name. There is no central process that checks names. Any machine on the network can reserve as many names as it wants as long as the names are not already in use. The NetBIOS interface can be implemented for different network architectures. An implementation that works relatively closely with network hardware is called NetBEUI, but this is often referred to as NetBIOS. Network protocols implemented with NetBIOS are IPX from Novell (NetBIOS via TCP/IP) and TCP/IP.

The NetBIOS names sent via TCP/IP have nothing in common with the names used in `/etc/hosts` or those defined by DNS. NetBIOS uses its own, completely independent naming convention. However, it is recommended to use names that correspond to DNS hostnames to make administration easier or use DNS natively. This is the default used by Samba.

Samba server

Samba server provides SMB/CIFS services and NetBIOS over IP naming services to clients. For Linux, there are three daemons for Samba server: `smbd` for SMB/CIFS services, `nmbd` for naming services, and `winbind` for authentication.

Samba client

The Samba client is a system that uses Samba services from a Samba server over the SMB protocol. All common operating systems, such as Mac OS X, Windows, and OS/2, support the SMB protocol. The TCP/IP protocol must be installed on all computers. Samba provides a client for the different UNIX flavors. For Linux, there is a kernel module for SMB that allows the integration of SMB resources on the Linux system level. You do not need to run any daemon for the Samba client.

Shares

SMB servers provide resources to the clients by means of shares. Shares are printers and directories with their subdirectories on the server. It is exported by means of a name and can be accessed by its name. The share name can be set to any name—it does not have to be the name of the export directory. A printer is also assigned a name. Clients can access the printer by its name.

DC

A domain controller (DC) is a server that handles accounts in domain. For data replication, additional domain controllers are available in one domain.

19.2 Installing a Samba Server

To install a Samba server, start YaST and select *Software > Software Management*. Choose *Filter > Patterns* and select *File Server*. Confirm the installation of the required packages to finish the installation process.

19.3 Starting and Stopping Samba

You can start or stop the Samba server automatically (during boot) or manually. Starting and stopping policy is a part of the YaST Samba server configuration described in Section 19.4.1, “Configuring a Samba Server with YaST” (page 337).

To stop or start running Samba services with YaST, use *System > System Services (Runlevel)* and check winbind, smb, and nmb. From a command line, stop services required for Samba with `rcsmb stop && rcnmb stop` and start them with `rcnmb start && rcsmb start`; rcsmb cares about winbind if needed.

19.4 Configuring a Samba Server

A Samba server in openSUSE® can be configured in two different ways: with YaST or manually. Manual configuration offers a higher level of detail, but lacks the convenience of the YaST GUI.

19.4.1 Configuring a Samba Server with YaST

To configure a Samba server, start YaST and select *Network Services > Samba Server*.

19.4.1.1 Initial Samba Configuration

When starting the module for the first time, the *Samba Installation* dialog starts, prompting you to make just a few basic decisions concerning administration of the server. At the end of the configuration it prompts for the Samba administrator password (*Samba Root Password*). For later starts, the *Samba Configuration* dialog appears.

The *Samba Installation* dialog consists of two steps and optional detailed settings:

Workgroup or Domain Name

Select an existing name from *Workgroup or Domain Name* or enter a new one and click *Next*.

Samba Server Type

In the next step, specify whether your server should act as a primary domain controller (PDC), backup domain controller (BDC), or not to act as a domain controller at all. Continue with *Next*.

If you do not want to proceed with a detailed server configuration, confirm with *OK*. Then in the final pop-up box, set the *Samba root Password*.

You can change all settings later in the *Samba Configuration* dialog with the *Start-Up*, *Shares*, *Identity*, *Trusted Domains*, and *LDAP Settings* tabs.

19.4.1.2 Advanced Samba Configuration

During the first start of the Samba server module the *Samba Configuration* dialog appears directly after the two initial steps described in Section 19.4.1.1, “Initial Samba Configuration” (page 337). Use it to adjust your Samba server configuration.

After editing your configuration, click *OK* to save your settings.

Starting the Server

In the *Start Up* tab, configure the start of the Samba server. To start the service every time your system boots, select *During Boot*. To activate manual start, choose *Manually*. More information about starting a Samba server is provided in Section 19.3, “Starting and Stopping Samba” (page 337).

In this tab, you can also open ports in your firewall. To do so, select *Open Port in Firewall*. If you have multiple network interfaces, select the network interface for Samba services by clicking *Firewall Details*, selecting the interfaces, and clicking *OK*.

Shares

In the *Shares* tab, determine the Samba shares to activate. There are some predefined shares, like homes and printers. Use *Toggle Status* to switch between *Active* and *Inactive*. Click *Add* to add new shares and *Delete* to delete the selected share.

Allow Users to Share Their Directories enables members of the group in *Permitted Group* to share directories they own with other users. For example, `users` for a local scope or `DOMAIN\Users` for a domain scope. The user also must make sure that the file system permissions allow access. With *Maximum Number of Shares*, limit the total amount of shares that may be created. To permit access to user shares without authentication, enable *Allow Guest Access*.

Identity

In the *Identity* tab, you can determine the domain with which the host is associated (*Base Settings*) and whether to use an alternative hostname in the network (*Net-BIOS Hostname*). It is also possible to use Microsoft Windows Internet Name Service (WINS) for name resolution. In this case, activate *Use WINS for Hostname Resolution* and decide whether to *Retrieve WINS server via DHCP*. To set expert global settings or set a user authentication source, click *Advanced Settings*.

Trusted Domains

To enable users from other domains to access your domain, make the appropriate settings in the *Trusted Domains* tab. To add a new domain, click *Add*. To remove the selected domain, click *Delete*.

LDAP Settings

In the tab *LDAP Settings*, you can determine the LDAP server to use for authentication. To test the connection to your LDAP server, click *Test Connection*. To set expert LDAP settings or use default values, click *Advanced Settings*.

For more information about LDAP configuration, see Chapter 4, *LDAP—A Directory Service* (↑Security Guide).

19.4.2 Web Administration with SWAT

An alternative tool for Samba server administration is SWAT (Samba Web Administration Tool). It provides a simple Web interface with which to configure the Samba server. To use SWAT, open <http://localhost:901> in a Web browser and log in as user `root`. If you do not have a special Samba root account, use the system `root` account.

NOTE: Activating SWAT

After Samba server installation, SWAT is not activated. To activate it, open *Network Services > Network Services (xinetd)* in YaST, enable the network services configuration, select *swat* from the table, and click *Toggle Status (On or Off)*.

19.4.3 Configuring the Server Manually

If you intend to use Samba as a server, install `samba`. The main configuration file for 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.

19.4.3.1 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 some other machine in the network. If the DNS name is not available, set the server name using `netbiosname=MYNAME`. For more details about this parameter, see the `smb.conf` man page.

```
os level = 20
```

This parameter triggers whether your Samba server tries to become LMB (local master browser) for its workgroup. With the Samba 3 release series, it is seldom necessary to override the default setting (20). Choose a very low value such as 2 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 Network Browsing chapter of the Samba 3 Howto; for more information on the Samba 3 Howto, see Section 19.7, “For More Information” (page 345).

If no other SMB server is present in your network (such as a Windows 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 need to 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.

19.4.3.2 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 19.1: *A CD-ROM Share (deactivated)*

```
[cdrom]
;      comment = Linux CD-ROM
;      path = /media/cdrom
;      locking = No
```

[cdrom] and comment

The `[cdrom]` section entry is the name of the share that can be seen by all SMB clients on the network. An additional `comment` can be added to further describe the share.

```
path = /media/cdrom
path exports the directory /media/cdrom.
```

By means of a very restrictive default configuration, this kind of share is only made available to the users present on this system. If this share should be made available to everybody, add a line `guest ok = yes` to the configuration. This setting gives read permissions to anyone on the network. It is recommended to handle this parameter with great care. This applies even more to the use of this parameter in the `[global]` section.

[homes]

The `[homes]` 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 19.2: *[homes] Share*

```
[homes]
comment = Home Directories
valid users = %S
browseable = No
read only = No
create mask = 0640
directory mask = 0750
```

[homes]

As long as there is no other share using the share name of the user connecting to the SMB server, a share is dynamically generated using the `[homes]` share directives. The resulting name of the share is the username.

```
valid users = %S
```

`%S` is replaced with the concrete name of the share as soon as a connection has been successfully established. For a `[homes]` share, this is always the username. As a consequence, access rights to a user's share are restricted exclusively to that user.

```
browseable = No
```

This setting makes the share invisible in the network environment.

```
read only = No
```

By default, Samba prohibits write access to any exported share by means of the `read only = Yes` parameter. To make a share writable, set the value `read only = No`, which is synonymous with `writable = Yes`.


```
create mask = 0640
```

Systems that are not based on MS Windows NT do not understand the concept of UNIX permissions, so they cannot assign permissions when creating a file. The parameter `create mask` defines the access permissions assigned to newly created files. This only applies to writable shares. In effect, this setting means the owner has read and write permissions and the members of the owner's primary group have read permissions. `valid users = %S` prevents read access even if the group has read permissions. For the group to have read or write access, deactivate the line `valid users = %S`.

19.4.3.3 Security Levels

To improve security, each share access can be protected with a password. SMB offers the following ways of checking 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 variant introduces the concept of the user to SMB. Each user must register with the server with his or her own password. After registration, the server can grant access to individual exported shares dependent on usernames.

Server Level Security (`security = server`)

To its clients, Samba pretends to be working in user level mode. However, it passes all password queries to another user level mode server, which takes care of authentication. This setting requires the additional `password server` parameter.

ADS Level Security (`security = ADS`)

In this mode, Samba will act as a domain member in an Active Directory environment. To operate in this mode, the machine running Samba needs Kerberos installed and configured. You must join the machine using Samba to the ADS realm. This can be done using the *YaST Windows Domain Membership* module.

Domain Level Security (`security = domain`)

This mode will only work correctly if the machine has been joined into a Windows NT Domain. Samba will try to validate username and password by passing it to a Windows NT Primary or Backup Domain Controller. The same way as a

Windows NT Server would do. It expects the encrypted passwords parameter to be set to `yes`.

The selection of share, user, server, or domain 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 3 HOWTO. For multiple servers on one system, pay attention to the options `interfaces` and `bind interfaces only`.

19.5 Configuring Clients

Clients can only access the Samba server via TCP/IP. NetBEUI and NetBIOS via IPX cannot be used with Samba.

19.5.1 Configuring a Samba Client with YaST

Configure a Samba client to access resources (files or printers) on the Samba or Windows server. Enter the NT or Active Directory domain or workgroup in the dialog *Network Services > Windows Domain Membership*. If you activate *Also Use SMB Information for Linux Authentication*, the user authentication runs over the Samba, NT or Kerberos server.

Click *Expert Settings* for advanced configuration options. For example, use the *Mount Server Directories* table to enable mounting server home directory automatically with authentication. This way users will be able to access their home directories when hosted on CIFS. For details, see the `pam_mount` man page.

After completing all settings, confirm the dialog to finish the configuration.

19.6 Samba as Login Server

In networks where predominantly Windows clients are found, it is often preferable that users may only register with a valid account and password. In a Windows-based

network, this task is handled by a primary domain controller (PDC). You can use a Windows NT server configured as PDC, but this task can also be done with a Samba server. The entries that must be made in the `[global]` section of `smb.conf` are shown in Example 19.3, “Global Section in `smb.conf`” (page 345).

Example 19.3: *Global Section in `smb.conf`*

```
[global]
    workgroup = TUX-NET
    domain logons = Yes
    domain master = Yes
```

If encrypted passwords are used for verification purposes 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 domain concept, with the following commands:

```
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.

```
add machine script = /usr/sbin/useradd -g nogroup -c "NT Machine Account" \
-s /bin/false %m$
```

To make sure that Samba can execute this script correctly, choose a Samba user with the required administrator permissions and add it to the `ntadmin` group. Then all users belonging to this Linux group can be assigned `Domain Admin` status with the command:

```
net groupmap add ntgroup="Domain Admins" unixgroup=ntadmin
```

For more information about this topic, see Chapter 12 of the Samba 3 HOWTO, found in `/usr/share/doc/packages/samba/Samba3-HOWTO.pdf`.

19.7 For More Information

Detailed Samba information is available in the digital documentation. Enter `apropos samba` at the command line to display some manual pages or just browse the /

`usr/share/doc/packages/samba` directory if Samba documentation is installed for more online documentation and examples. Find a commented example configuration (`smb.conf.SUSE`) in the `examples` subdirectory.

The Samba 3 HOWTO provided by the Samba team includes a section about troubleshooting. In addition to that, Part V of the document provides a step-by-step guide to checking your configuration. You can find Samba 3 HOWTO in `/usr/share/doc/packages/samba/Samba3-HOWTO.pdf` after installing the package `samba-doc`.

Also read the Samba page in the openSUSE wiki at <http://en.opensuse.org/Samba>.

The Apache HTTP Server

With a share of more than 50%, the Apache HTTP Server (Apache) is the world's most widely-used Web server according to the survey from <http://www.netcraft.com/>. Apache, developed by the Apache Software Foundation (<http://www.apache.org/>), is available for most operating systems. openSUSE® includes Apache version 2.2. In this chapter, learn how to install, configure and set up a Web server; how to use SSL, CGI, and additional modules; and how to troubleshoot Apache.

20.1 Quick Start

With the help of this section, quickly set up and start Apache. You must be `root` to install and configure Apache.

20.1.1 Requirements

Make sure the following requirements are met before trying to set up the Apache Web server:

1. The machine's network is configured properly. For more information about this topic, refer to Chapter 13, *Basic Networking* (page 215).
2. The machine's exact system time is maintained by synchronizing with a time server. This is necessary because parts of the HTTP protocol depend on the correct time.

See Chapter 17, *Time Synchronization with NTP* (page 313) to learn more about this topic.

3. The latest security updates are installed. If in doubt, run a YaST Online Update.
4. The default Web server port (80) is opened in the firewall. For this, configure the SuSEFirewall2 to allow the service *HTTP Server* in the external zone. This can be done using YaST. See Section “Configuring the Firewall with YaST” (Chapter 13, *Masquerading and Firewalls*, ↑Security Guide) for details.

20.1.2 Installation

Apache on openSUSE is not installed by default. To install it with a standard, predefined configuration that runs “out of the box”, proceed as follows:

Procedure 20.1: *Installing Apache with the Default Configuration*

- 1 Start YaST and select *Software > Software Management*.
- 2 Choose *Filter > Patterns* and select *Web and LAMP Server* in *Server Functions*.
- 3 Confirm the installation of the dependent packages to finish the installation process.

The installation includes the multiprocessing module `apache2-prefork` as well as the PHP5 module. Refer to Section 20.4, “Installing, Activating, and Configuring Modules” (page 366) for more information about modules.

20.1.3 Start

You can start Apache automatically at boot time or start it manually.

Procedure 20.2: *Starting Apache Automatically*

- 1 To make sure that Apache is automatically started during boot in runlevels 3 and 5, execute the following command:

```
chkconfig -a apache2
```

- 2 Alternatively, start YaST and select *System > System Services (Runlevel)*.

- 3 Search for *apache2* and *Enable* the service.

The Web server starts immediately.

- 4 Save your changes with *Finish*.

The system is configured to automatically start Apache in runlevels 3 and 5 during boot.

To manually start Apache using the shell, run `rcapache2 start`.

Procedure 20.3: *Checking if Apache is Running*

If you do not receive error messages when starting Apache, this usually indicates that the Web server is running. To test this:

- 1 Start a browser and open <http://localhost/>.

If Apache is up and running, you get a test page stating “It works!”.

- 2 If you do not see this page, refer to Section 20.8, “Troubleshooting” (page 385).

Now that the Web server is running, you can add your own documents, adjust the configuration according to your needs, or add functionality by installing modules.

20.2 Configuring Apache

openSUSE offers two configuration options:

- Configuring Apache Manually (page 353)
- Configuring Apache with YaST (page 357)

Manual configuration offers a higher level of detail, but lacks the convenience of the YaST GUI.

IMPORTANT: Reload or Restart Apache after Configuration Changes

Most configuration changes require a reload (some also a restart) of Apache to take effect. Manually reload Apache with `rcapache2 reload` or use one of the restart options as described in Section 20.3, “Starting and Stopping Apache” (page 364).

If you configure Apache with YaST, this can be taken care of automatically if you set *HTTP Service to Enabled* as described in Section 20.2.3.2, “HTTP Server Configuration” (page 362).

20.2.1 Apache Configuration Files

This section gives an overview of the Apache configuration files. If you use YaST for configuration, you do not need to touch these files—however, the information might be useful for you if you want to switch to manual configuration later on.

Apache configuration files can be found in two different locations:

- `/etc/sysconfig/apache2` (page 350)
- `/etc/apache2/` (page 350)

20.2.1.1 `/etc/sysconfig/apache2`

`/etc/sysconfig/apache2` controls some global settings of Apache, like modules to load, additional configuration files to include, flags with which the server should be started, and flags that should be added to the command line. Every configuration option in this file is extensively documented and therefore not mentioned here. For a general-purpose Web server, the settings in `/etc/sysconfig/apache2` should be sufficient for any configuration needs.

20.2.1.2 `/etc/apache2/`

`/etc/apache2/` hosts all configuration files for Apache. In the following, the purpose of each file is explained. Each file includes several configuration options (also referred to as *directives*). Every configuration option in these files is extensively documented and therefore not mentioned here.

The Apache configuration files are organized as follows:

```
/etc/apache2/  
|  
|- charset.conv  
|- conf.d/  
|  |  
|  |- *.conf
```



```

|
|- default-server.conf
|- errors.conf
|- httpd.conf
|- listen.conf
|- magic
|- mime.types
|- mod_*.conf
|- server-tuning.conf
|- ssl.*
|- ssl-global.conf
|- sysconfig.d
|   |
|   |- global.conf
|   |- include.conf
|   |- loadmodule.conf . .
|
|- uid.conf
|- vhosts.d
|   |- *.conf

```

Apache Configuration Files in /etc/apache2/

`charset.conf`

Specifies which character sets to use for different languages. Do not edit this file.

`conf.d/*.conf`

Configuration files added by other modules. These configuration files can be included into your virtual host configuration where needed. See `vhosts.d/vhost.template` for examples. By doing so, you can provide different module sets for different virtual hosts.

`default-server.conf`

Global configuration for all virtual hosts with reasonable defaults. Instead of changing the values, overwrite them with a virtual host configuration.

`errors.conf`

Defines how Apache responds to errors. To customize these messages for all virtual hosts, edit this file. Otherwise overwrite these directives in your virtual host configurations.

`httpd.conf`

The main Apache server configuration file. Avoid changing this file. It primarily contains include statements and global settings. Overwrite global settings in the pertinent configuration files listed here. Change host-specific settings (such as document root) in your virtual host configuration.

`listen.conf`

Binds Apache to specific IP addresses and ports. Name-based virtual hosting is also configured here. For details, see Section “Name-Based Virtual Hosts” (page 354).

`magic`

Data for the `mime_magic` module that helps Apache automatically determine the MIME type of an unknown file. Do not change this file.

`mime.types`

MIME types known by the system (this actually is a link to `/etc/mime.types`). Do not edit this file. If you need to add MIME types not listed here, add them to `mod_mime-defaults.conf`.

`mod_*.conf`

Configuration files for the modules that are installed by default. Refer to Section 20.4, “Installing, Activating, and Configuring Modules” (page 366) for details. Note that configuration files for optional modules reside in the directory `conf.d`.

`server-tuning.conf`

Contains configuration directives for the different MPMs (see Section 20.4.4, “Multiprocessing Modules” (page 371)) as well as general configuration options that control Apache's performance. Properly test your Web server when making changes here.

`ssl-global.conf` and `ssl.*`

Global SSL configuration and SSL certificate data. Refer to Section 20.6, “Setting Up a Secure Web Server with SSL” (page 377) for details.

`sysconfig.d/*.conf`

Configuration files automatically generated from `/etc/sysconfig/apache2`. Do not change any of these files—edit `/etc/sysconfig/apache2` instead. Do not put other configuration files in this directory.

`uid.conf`

Specifies under which user and group ID Apache runs. Do not change this file.

`vhosts.d/*.conf`

Your virtual host configuration should be located here. The directory contains template files for virtual hosts with and without SSL. Every file in this directory

ending with `.conf` is automatically included in the Apache configuration. Refer to Section 20.2.2.1, “Virtual Host Configuration” (page 353) for details.

20.2.2 Configuring Apache Manually

Configuring Apache manually involves editing plain text configuration files as user `root`.

20.2.2.1 Virtual Host Configuration

The term *virtual host* refers to Apache's ability to serve multiple universal resource identifiers (URIs) from the same physical machine. This means that several domains, such as `www.example.com` and `www.example.net`, are run by a single Web server on one physical machine.

It is common practice to use virtual hosts to save administrative effort (only a single Web server needs to be maintained) and hardware expenses (each domain does not require a dedicated server). Virtual hosts can be name based, IP based, or port based.

To list all existing virtual hosts, use the command `httpd2 -S`. This outputs a list showing the default server and all virtual hosts together with their IP addresses and listening ports. Furthermore, the list also contains an entry for each virtual host showing its location in the configuration files.

Virtual hosts can be configured via YaST as described in Section “Virtual Hosts” (page 360) or by manually editing a configuration file. By default, Apache in openSUSE is prepared for one configuration file per virtual host in `/etc/apache2/vhosts.d/`. All files in this directory with the extension `.conf` are automatically included to the configuration. A basic template for a virtual host is provided in this directory (`vhost.template` or `vhost-ssl.template` for a virtual host with SSL support).

TIP: Always Create a Virtual Host Configuration

It is recommended to always create a virtual host configuration file, even if your Web server only hosts one domain. By doing so, you not only have the domain-specific configuration in one file, but you can always fall back to a working basic configuration by simply moving, deleting, or renaming the con-

figuration file for the virtual host. For the same reason, you should also create separate configuration files for each virtual host.

When using name-based virtual hosts it is recommended to set up a default configuration that will be used when a domain name does not match a virtual host configuration. The default virtual host is the one whose configuration is loaded first. Since the order of the configuration files is determined by filename, start the filename of the default virtual host configuration with an underscore character (`_`) to make sure it is loaded first (for example: `_default_vhost.conf`).

The `<VirtualHost></VirtualHost>` block holds the information that applies to a particular domain. When Apache receives a client request for a defined virtual host, it uses the directives enclosed in this section. Almost all directives can be used in a virtual host context. See <http://httpd.apache.org/docs/2.2/mod/quickreference.html> for further information about Apache's configuration directives.

Name-Based Virtual Hosts

With name-based virtual hosts, more than one Web site is served per IP address. Apache uses the host field in the HTTP header that is sent by the client to connect the request to a matching `ServerName` entry of one of the virtual host declarations. If no matching `ServerName` is found, the first specified virtual host is used as a default.

The directive `NameVirtualHost` tells Apache on which IP address and, optionally, which port it should listen to for requests by clients containing the domain name in the HTTP header. This option is configured in the configuration file `/etc/apache2/listen.conf`.

The first argument can be a fully qualified domain name, but it is recommended to use the IP address. The second argument is the port and is optional. By default, port 80 is used and is configured via the `Listen` directive.

The wild card `*` can be used for both the IP address and the port number to receive requests on all interfaces. IPv6 addresses must be enclosed in square brackets.

Example 20.1: *Variations of Name-Based VirtualHost Entries*

```
# NameVirtualHost IP-address[:Port]
```

```
NameVirtualHost 192.168.3.100:80
NameVirtualHost 192.168.3.100
NameVirtualHost *:80
NameVirtualHost *
NameVirtualHost [2002:c0a8:364::]:80
```

The opening `VirtualHost` tag takes the IP address (or fully qualified domain name) previously declared with the `NameVirtualHost` as an argument in a name-based virtual host configuration. A port number previously declared with the `NameVirtualHost` directive is optional.

The wild card `*` is also allowed as a substitute for the IP address. This syntax is only valid in combination with the wild card usage in `NameVirtualHost *`. When using IPv6 addresses, the address must be included in square brackets.

Example 20.2: Name-Based VirtualHost Directives

```
<VirtualHost 192.168.3.100:80>
...
</VirtualHost>

<VirtualHost 192.168.3.100>
...
</VirtualHost>

<VirtualHost *:80>
...
</VirtualHost>

<VirtualHost *>
...
</VirtualHost>

<VirtualHost [2002:c0a8:364::]>
...
</VirtualHost>
```

IP-Based Virtual Hosts

This alternative virtual host configuration requires the setup of multiple IPs for a machine. One instance of Apache hosts several domains, each of which is assigned a different IP.

The physical server must have one IP address for each IP-based virtual host. If the machine does not have multiple network cards, virtual network interfaces (IP aliasing) can also be used.

The following example shows Apache running on a machine with the IP 192.168.3.100, hosting two domains on the additional IPs 192.168.3.101 and 192.168.3.102. A separate `VirtualHost` block is needed for every virtual server.

Example 20.3: *IP-Based VirtualHost Directives*

```
<VirtualHost 192.168.3.101>
...
</VirtualHost>

<VirtualHost 192.168.3.102>
...
</VirtualHost>
```

Here, `VirtualHost` directives are only specified for interfaces other than 192.168.3.100. When a `Listen` directive is also configured for 192.168.3.100, a separate IP-based virtual host must be created to answer HTTP requests to that interface—otherwise the directives found in the default server configuration (`/etc/apache2/default-server.conf`) are applied.

Basic Virtual Host Configuration

At least the following directives should be present in each virtual host configuration in order to set up a virtual host. See `/etc/apache2/vhosts.d/vhost.template` for more options.

`ServerName`

The fully qualified domain name under which the host should be addressed.

`DocumentRoot`

Path to the directory from which Apache should serve files for this host. For security reasons, access to the entire file system is forbidden by default, so you must explicitly unlock this directory within a `Directory` container.

`ServerAdmin`

E-mail address of the server administrator. This address is, for example, shown on error pages Apache creates.

`ErrorLog`

The error log file for this virtual host. Although it is not necessary to create separate error log files for each virtual host, it is common practice to do so, because

it makes the debugging of errors much easier. `/var/log/apache2/` is the default directory for Apache's log files.

CustomLog

The access log file for this virtual host. Although it is not necessary to create separate access log files for each virtual host, it is common practice to do so, because it allows the separate analysis of access statistics for each host. `/var/log/apache2/` is the default directory for Apache's log files.

As mentioned above, access to the whole file system is forbidden by default for security reasons. Therefore, explicitly unlock the directories in which you have placed the files Apache should serve—for example the `DocumentRoot`:

```
<Directory "/srv/www/www.example.com/htdocs">
    Order allow,deny
    Allow from all
</Directory>
```

The complete configuration file looks like this:

Example 20.4: *Basic VirtualHost Configuration*

```
<VirtualHost 192.168.3.100>
    ServerName www.example.com
    DocumentRoot /srv/www/www.example.com/htdocs
    ServerAdmin webmaster@example.com
    ErrorLog /var/log/apache2/www.example.com_log
    CustomLog /var/log/apache2/www.example.com-access_log common
    <Directory "/srv/www/www.example.com/htdocs">
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

20.2.3 Configuring Apache with YaST

To configure your Web server with YaST, start YaST and select *Network Services* > *HTTP Server*. When starting the module for the first time, the *HTTP Server Wizard* starts, prompting you to make a few basic decisions concerning administration of the server. After having finished the wizard, the *HTTP Server Configuration* dialog starts each time you call the *HTTP Server* module. For more information, see Section 20.2.3.2, “HTTP Server Configuration” (page 362).

20.2.3.1 HTTP Server Wizard

The HTTP Server Wizard consists of five steps. In the last step of the dialog, you are given the opportunity to enter the expert configuration mode to make even more specific settings.

Network Device Selection

Here, specify the network interfaces and ports Apache uses to listen for incoming requests. You can select any combination of existing network interfaces and their respective IP addresses. Ports from all three ranges (well-known ports, registered ports, and dynamic or private ports) that are not reserved by other services can be used. The default setting is to listen on all network interfaces (IP addresses) on port 80.

Check *Open Port In Firewall* to open the ports in the firewall that the Web server listens on. This is necessary to make the Web server available on the network, which can be a LAN, WAN, or the public Internet. Keeping the port closed is only useful in test situations where no external access to the Web server is necessary. If you have multiple network interfaces, click *Firewall Details...* to specify on which interface(s) the port(s) should be opened.

Click *Next* to continue with the configuration.

Modules


The *Modules* configuration option allows for the activation or deactivation of the script languages that the Web server should support. For the activation or deactivation of other modules, refer to Section “Server Modules” (page 363). Click *Next* to advance to the next dialog.

Default Host

This option pertains to the default Web server. As explained in Section 20.2.2.1, “Virtual Host Configuration” (page 353), Apache can serve multiple virtual hosts from a single physical machine. The first declared virtual host in the configuration file is commonly referred to as the *default host*. Each virtual host inherits the default host's configuration.

To edit the host settings (also called *directives*), choose the appropriate entry in the table then click *Edit*. To add new directives, click *Add*. To delete a directive, select it and click *Delete*.

Figure 20.1: *HTTP Server Wizard: Default Host*

 **HTTP Server Wizard (3/5)--Default Host**
To edit the host settings, choose the appropriate entry of the table then click Edit. [more](#)

Option	Value
Document Root	<code>/srv/www/htdocs</code>
Directory	<code>/srv/www/htdocs/...</code>
Alias	<code>/icons/ /usr/share/apache2/icons/</code>
Directory	<code>/usr/share/apache2/icons/...</code>
ScriptAlias	<code>/cgi-bin/ /srv/www/cgi-bin/</code>
Directory	<code>/srv/www/cgi-bin/...</code>
<code>mod_userdir.c</code>	
Include	<code>/etc/apache2/conf.d/* conf</code>
Include	<code>/etc/apache2/conf.d/apache2-manual?conf</code>
Server Name	<code>archimedes</code>
Server Administrator E-Mail	<code>root@archimedes</code>

Here is list of the default settings of the server:

Document Root

Path to the directory from which Apache serves files for this host. `/srv/www/htdocs` is the default location.

Alias

With the help of `Alias` directives, URLs can be mapped to physical file system locations. This means that a certain path even outside the `Document Root` in the file system can be accessed via a URL aliasing that path.

The default openSUSE `Alias /icons` points to `/usr/share/apache2/icons` for the Apache icons displayed in the directory index view.

ScriptAlias

Similar to the `Alias` directive, the `ScriptAlias` directive maps a URL to a file system location. The difference is that `ScriptAlias` designates the target directory as a CGI location, meaning that CGI scripts should be executed in that location.

Directory

With `Directory` settings, you can enclose a group of configuration options that will only apply to the specified directory.

Access and display options for the directories `/srv/www/htdocs`, `/usr/share/apache2/icons` and `/srv/www/cgi-bin` are configured here. It should not be necessary to change the defaults.

Include

With `include`, additional configuration files can be specified. Two `Include` directives are already preconfigured: `/etc/apache2/conf.d/` is the directory containing the configuration files that come with external modules. With this directive, all files in this directory ending in `.conf` are included. With the second directive, `/etc/apache2/conf.d/apache2-manual.conf`, the `apache2-manual` configuration file is included.

Server Name

This specifies the default URL used by clients to contact the Web server. Use a fully qualified domain name (FQDN) to reach the Web server at `http://FQDN/` or its IP address. You cannot choose an arbitrary name here—the server must be “known” under this name.

Server Administrator E-Mail

E-mail address of the server administrator. This address is, for example, shown on error pages Apache creates.

After finishing with the *Default Host* step, click *Next* to continue with the configuration.

Virtual Hosts

In this step, the wizard displays a list of already configured virtual hosts (see Section 20.2.2.1, “Virtual Host Configuration” (page 353)). If you have not made manual changes prior to starting the YaST HTTP wizard, no virtual host is present.

To add a host, click *Add* to open a dialog in which to enter basic information about the host, such as *Server Name*, *Server Contents Root* (`DocumentRoot`), and the *Administrator E-Mail*. *Server Resolution* is used to determine how a host is identified (name based or IP based). Specify the name or IP address with *Change Virtual Host ID*

Clicking *Next* advances to the second part of the virtual host configuration dialog.

In part two of the virtual host configuration you can specify whether or not to enable CGI scripts and which directory to use for these scripts. It is also possible to enable SSL. If you do so, you must specify the path to the certificate as well. See Section 20.6.2, “Configuring Apache with SSL” (page 382) for details on SSL and certificates. With the *Directory Index* option, you can specify which file to display when the client requests a directory (by default, `index.html`). Add one or more filenames (space-separated) if you want to change this. With *Enable Public HTML*, the content of the users public directories (`~user/public_html/`) is made available on the server under `http://www.example.com/~user`.

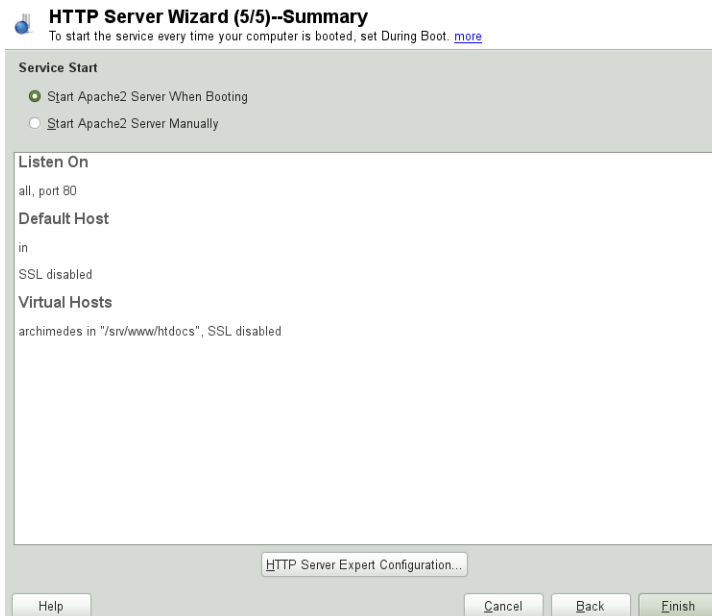
IMPORTANT: Creating Virtual Hosts

It is not possible to add virtual hosts at will. If using name-based virtual hosts, each hostname must be resolved on the network. If using IP-based virtual hosts, you can assign only one host to each IP address available.

Summary

This is the final step of the wizard. Here, determine how and when the Apache server is started: when booting or manually. Also see a short summary of the configuration made so far. If you are satisfied with your settings, click *Finish* to complete configuration. If you want to change something, click *Back* until you have reached the desired dialog. Clicking *HTTP Server Expert Configuration* opens the dialog described in Section 20.2.3.2, “HTTP Server Configuration” (page 362).

Figure 20.2: *HTTP Server Wizard: Summary*



20.2.3.2 HTTP Server Configuration

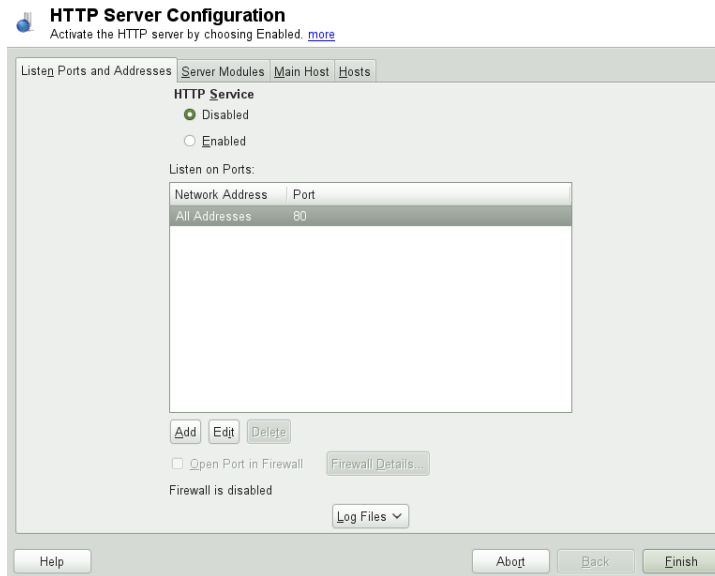
The *HTTP Server Configuration* dialog also lets you make even more adjustments to the configuration than the wizard (which only runs if you configure your Web server for the first time). It consists of four tabs described in the following. No configuration option you change here is effective immediately—you always must confirm your changes with *Finish* to make them effective. Clicking *Abort* leaves the configuration module and discards your changes.

Listen Ports and Addresses

In *HTTP Service*, select whether Apache should be running (*Enabled*) or stopped (*Disabled*). In *Listen on Ports*, *Add*, *Edit*, or *Delete* addresses and ports on which the server should be available. The default is to listen on all interfaces on port 80. You should always check *Open Port In Firewall*, because otherwise the Web server is not reachable from outside. Keeping the port closed is only useful in test situations where no external access to the Web server is necessary. If you have multiple network interfaces, click *Firewall Details...* to specify on which interface(s) the port(s) should be opened.

With *Log Files*, watch either the access log or the error log. This is useful if you want to test your configuration. The log file opens in a separate window from which you can also restart or reload the Web server. For details, see Section 20.3, “Starting and Stopping Apache” (page 364). These commands are effective immediately and their log messages are also displayed immediately.

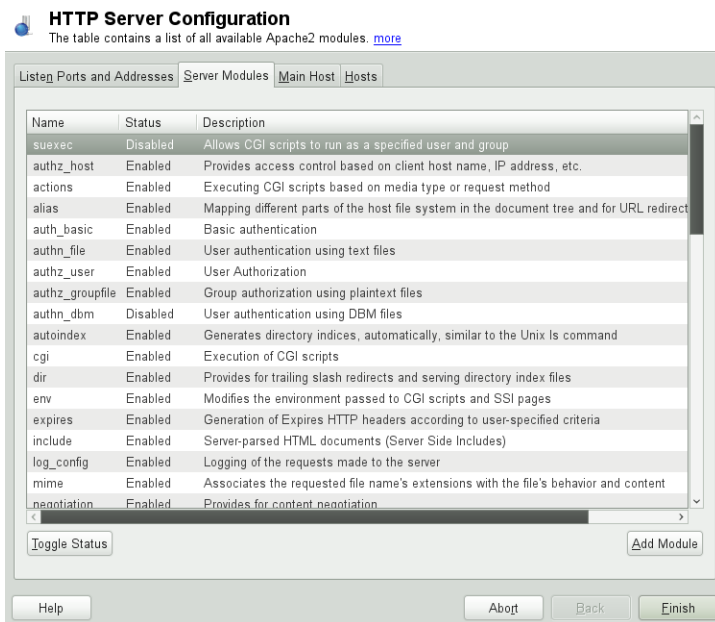
Figure 20.3: *HTTP Server Configuration: Listen Ports and Addresses*



Server Modules

You can change the status (enabled or disabled) of Apache2 modules by clicking *Toggle Status*. Click *Add Module* to add a new module that is already installed but not yet listed. Learn more about modules in Section 20.4, “Installing, Activating, and Configuring Modules” (page 366).

Figure 20.4: *HTTP Server Configuration: Server Modules*



Main Host or Hosts

These dialogs are identical to the ones already described. Refer to Section “Default Host” (page 358) and Section “Virtual Hosts” (page 360).

20.3 Starting and Stopping Apache

If configured with YaST as described in Section 20.2.3, “Configuring Apache with YaST” (page 357), Apache is started at boot time in runlevels 3 and 5 and stopped in runlevels 0, 1, 2, and 6. You can change this behavior using YaST's runlevel editor or the command line tool `chkconfig`.

To start, stop, or manipulate Apache on a running system, use the init script `/usr/sbin/rcapache2`. The `rcapache2` command takes the following parameters:

`status`

Checks if Apache is started.

`start`

Starts Apache if it is not already running.

`startssl`

Starts Apache with SSL support if it is not already running. For more information about SSL support, refer to Section 20.6, “Setting Up a Secure Web Server with SSL” (page 377).

`stop`

Stops Apache by terminating the parent process.

`restart`

Stops and then restarts Apache. Starts the Web server if it was not running before.

`try-restart`

Stops then restarts Apache only if it is already running.

`reload or graceful`

Stops the Web server by advising all forked Apache processes to first finish their requests before shutting down. As each process dies, it is replaced by a newly started one, resulting in a complete “restart” of Apache.

TIP: Restarting Apache in Production Environments

To activate changes in the Apache configuration without causing connection break-offs, use the `rcapache2 reload` command.

`restart-graceful`

Starts a second Web server that immediately serves all incoming requests. The previous instance of the Web server continues to handle all existing requests for a defined period of time configured with `GracefulShutdownTimeout`.

`rcapache2 restart-graceful` is either useful when upgrading to a new version or when having changed configuration options that require a restart. Using this option ensures a minimum server downtime.

`GracefulShutdownTimeout` needs to be set, otherwise `restart-graceful` will result in a regular restart. If set to zero, the server will wait indefinitely until all remaining requests have been fully served.

A graceful restart can fail if the original Apache instance is not able to clear all necessary resources. In this case, the command will result in a graceful stop.

`stop-graceful`

Stops the Web server after a defined period of time configured with `GracefulShutdownTimeout` in order to ensure that existing requests can be finished.

`GracefulShutdownTimeout` needs to be set, otherwise `stop-graceful` will result in a regular restart. If set to zero, the server will wait indefinitely until all remaining requests have been fully served.

`configtest` or `extreme-configtest`

Checks the syntax of the configuration files without affecting a running Web server. Because this check is forced every time the server is started, reloaded, or restarted, it is usually not necessary to run the test explicitly (if a configuration error is found, the Web server is not started, reloaded, or restarted). The `extreme-configtest` options start the Web server as user `nobody` and actually load the configuration, so more errors can be detected. Note that although the configuration is loaded, it is not possible to test the SSL setup because the SSL certificates cannot be read by `nobody`.

`probe`

Probes for the necessity of a reload (checks whether the configuration has changed) and suggests the required arguments for the `rcapache2` command.

`server-status` and `full-server-status`

Dumps a short or full status screen, respectively. Requires either `lynx` or `w3m` installed as well as the module `mod_status` enabled. In addition to that, `status` must be added to `APACHE_SERVER_FLAGS` in the file `/etc/sysconfig/apache2`.

TIP: Additional Flags

If you specify additional flags to the `rcapache2`, these are passed through to the Web server.

20.4 Installing, Activating, and Configuring Modules

The Apache software is built in a modular fashion: all functionality except some core tasks are handled by modules. This has progressed so far that even HTTP is processed by a module (`http_core`).

Apache modules can be compiled into the Apache binary at build time or dynamically loaded at runtime. Refer to Section 20.4.2, “Activation and Deactivation” (page 368) for details of how to load modules dynamically.

Apache modules can be divided into four different categories:

Base Modules

Base modules are compiled into Apache by default. Apache in openSUSE has only `mod_so` (needed to load other modules) and `http_core` compiled in. All others are available as shared objects: rather than being included in the server binary itself, they can be included at runtime.

Extension Modules

In general, modules labeled as extensions are included in the Apache software package, but are usually not compiled into the server statically. In openSUSE, they are available as shared objects that can be loaded into Apache at runtime.

External Modules

Modules labeled external are not included in the official Apache distribution. However, openSUSE provides several of them.

Multiprocessing Modules (MPMs)

MPMs are responsible for accepting and handling requests to the Web server, representing the core of the Web server software.

20.4.1 Module Installation

If you have done a default installation as described in Section 20.1.2, “Installation” (page 348), the following modules are already installed: all base and extension modules, the multiprocessing module Prefork MPM, and the external modules `mod_php5` and `mod_python`.

You can install additional external modules by starting YaST and choosing *Software > Software Management*. Now choose *Filter > Search* and search for *apache*. Among other packages, the results list contains all available external Apache modules.

20.4.2 Activation and Deactivation

Activate or deactivate particular modules either manually or with YaST. In YaST, script language modules (PHP5, Perl, and Python) need to be enabled or disabled with the module configuration described in Section 20.2.3.1, “HTTP Server Wizard” (page 358). All other modules can be enabled or disabled as described in Section “Server Modules” (page 363).

If you prefer to activate or deactivate the modules manually, use the commands `a2enmod mod_foo` or `a2dismod mod_foo`, respectively. `a2enmod -l` outputs a list of all currently active modules.

IMPORTANT: Including Configuration Files for External Modules

If you have activated external modules manually, make sure to load their configuration files in all virtual host configurations. Configuration files for external modules are located under `/etc/apache2/conf.d/` and are not loaded by default. If you need the same modules on each virtual host, you can include `*.conf` from this directory. Otherwise include individual files. See `/etc/apache2/vhost.d/vhost.template` for examples.

20.4.3 Base and Extension Modules

All base and extension modules are described in detail in the Apache documentation. Only a brief description of the most important modules is available here. Refer to <http://httpd.apache.org/docs/2.2/mod/> to learn details about each module.

`mod_actions`

Provides methods to execute a script whenever a certain MIME type (such as `application/pdf`), a file with a specific extension (like `.rpm`), or a certain request method (such as `GET`) is requested. This module is enabled by default.

`mod_alias`

Provides `Alias` and `Redirect` directives with which you can map a URI to a specific directory (`Alias`) or redirect a requested URL to another location. This module is enabled by default.

`mod_auth*`

The authentication modules provide different authentication methods: basic authentication with `mod_auth_basic` or digest authentication with `mod_auth_digest`. Digest authentication in Apache 2.2 is considered experimental.

`mod_auth_basic` and `mod_auth_digest` must be combined with an authentication provider module, `mod_authn_*` (for example, `mod_authn_file` for text file-based authentication) and with an authorization module `mod_authz_*` (for example, `mod_authz_user` for user authorization).

More information about this topic is available in the *Authentication HOWTO* at <http://httpd.apache.org/docs/2.2/howto/auth.html>.

`mod_autoindex`

Autoindex generates directory listings when no index file (for example, `index.html`) is present. The look and feel of these indexes is configurable. This module is enabled by default. However, directory listings are disabled by default via the `Options` directive—overwrite this setting in your virtual host configuration. The default configuration file for this module is located at `/etc/apache2/mod_autoindex-defaults.conf`.

`mod_cgi`

`mod_cgi` is needed to execute CGI scripts. This module is enabled by default.

`mod_deflate`

Using this module, Apache can be configured to compress given file types on the fly before delivering them.

`mod_dir`

`mod_dir` provides the `DirectoryIndex` directive with which you can configure which files are automatically delivered when a directory is requested (`index.html` by default). It also provides an automatic redirect to the correct URL when a directory request does not contain a trailing slash. This module is enabled by default.

`mod_env`

Controls the environment that is passed to CGI scripts or SSI pages. Environment variables can be set or unset or passed from the shell that invoked the `httpd` process. This module is enabled by default.

`mod_expires`

With `mod_expires`, you can control how often proxy and browser caches refresh your documents by sending an `Expires` header. This module is enabled by default.

`mod_include`

`mod_include` lets you use Server Side Includes (SSI), which provide a basic functionality to generate HTML pages dynamically. This module is enabled by default.

`mod_info`

Provides a comprehensive overview of the server configuration under `http://localhost/server-info/`. For security reasons, you should always limit access to this URL. By default only `localhost` is allowed to access this URL. `mod_info` is configured at `/etc/apache2/mod_info.conf`.

`mod_log_config`

With this module, you can configure the look of the Apache log files. This module is enabled by default.

`mod_mime`

The mime module makes certain that a file is delivered with the correct MIME header based on the filename's extension (for example `text/html` for HTML documents). This module is enabled by default.

`mod_negotiation`

Necessary for content negotiation. See <http://httpd.apache.org/docs/2.2/content-negotiation.html> for more information. This module is enabled by default.

`mod_rewrite`

Provides the functionality of `mod_alias`, but offers more features and flexibility. With `mod_rewrite`, you can redirect URLs based on multiple rules, request headers, and more.

`mod_setenvif`

Sets environment variables based on details of the client's request, such as the browser string the client sends, or the client's IP address. This module is enabled by default.

`mod_speling`

`mod_speling` attempts to automatically correct typographical errors in URLs, such as capitalization errors.

`mod_ssl`

Enables encrypted connections between Web server and clients. See Section 20.6, “Setting Up a Secure Web Server with SSL” (page 377) for details. This module is enabled by default.

`mod_status`

Provides information on server activity and performance under `http://localhost/server-status/`. For security reasons, you should always limit access to this URL. By default, only `localhost` is allowed to access this URL. `mod_status` is configured at `/etc/apache2/mod_status.conf`

`mod_suexec`

`mod_suexec` lets you run CGI scripts under a different user and group. This module is enabled by default.

`mod_userdir`

Enables user-specific directories available under `~user/`. The `UserDir` directive must be specified in the configuration. This module is enabled by default.

20.4.4 Multiprocessing Modules

openSUSE provides two different multiprocessing modules (MPMs) for use with Apache:

- Prefork MPM (page 371)
- Section 20.4.4.2, “Worker MPM” (page 372)

20.4.4.1 Prefork MPM

The prefork MPM implements a nonthreaded, preforking Web server. It makes the Web server behave similarly to Apache version 1.x. In this version it isolates each request and handles it by forking a separate child process. Thus problematic requests cannot affect others, avoiding a lockup of the Web server.

While providing stability with this process-based approach, the prefork MPM consumes more system resources than its counterpart, the worker MPM. The prefork MPM is considered the default MPM for Unix-based operating systems.

IMPORTANT: MPMs in This Document

This document assumes Apache is used with the prefork MPM.

20.4.4.2 Worker MPM

The worker MPM provides a multi-threaded Web server. A thread is a “lighter” form of a process. The advantage of a thread over a process is its lower resource consumption. Instead of only forking child processes, the worker MPM serves requests by using threads with server processes. The preforked child processes are multi-threaded. This approach makes Apache perform better by consuming fewer system resources than the prefork MPM.

One major disadvantage is the stability of the worker MPM: if a thread becomes corrupt, all threads of a process can be affected. In the worst case, this may result in a server crash. Especially when using the Common Gateway Interface (CGI) with Apache under heavy load, internal server errors might occur due to threads being unable to communicate with system resources. Another argument against using the worker MPM with Apache is that not all available Apache modules are thread-safe and thus cannot be used in conjunction with the worker MPM.

WARNING: Using PHP Modules with MPMs

Not all available PHP modules are thread-safe. Using the worker MPM with `mod_php` is strongly discouraged.

20.4.5 External Modules

Find a list of all external modules shipped with openSUSE here. Find the module's documentation in the listed directory.

`mod-apparmor`

Adds support to Apache to provide AppArmor confinement to individual CGI scripts handled by modules like `mod_php5` and `mod_perl`.

Package Name: `apache2-mod_apparmor`
More Information: Part “Confining Privileges with AppArmor” (↑Security Guide)

`mod_mono`

Using `mod_mono` allows you to run ASP.NET pages in your server.

Package Name: `apache2-mod_mono`
Configuration File: `/etc/apache2/conf.d/mod_mono.conf`

`mod_perl`

`mod_perl` enables you to run Perl scripts in an embedded interpreter. The persistent interpreter embedded in the server avoids the overhead of starting an external interpreter and the penalty of Perl start-up time.

Package Name: `apache2-mod_perl`
Configuration File: `/etc/apache2/conf.d/mod_perl.conf`
More Information: `/usr/share/doc/packages/apache2-mod_perl`

`mod_php5`

PHP is a server-side, cross-platform HTML embedded scripting language.

Package Name: `apache2-mod_php5`
Configuration File: `/etc/apache2/conf.d/php5.conf`
More Information: `/usr/share/doc/packages/apache2-mod_php5`

`mod_python`

`mod_python` allows embedding Python within the Apache HTTP server for a considerable boost in performance and added flexibility in designing Web-based applications.

Package Name: `apache2-mod_python`
More Information: `/usr/share/doc/packages/apache2-mod_python`

`mod_tidy`

`mod_tidy` validates each outgoing HTML page by means of the TidyLib. In case of a validation error, a page with an error list is delivered. Otherwise the original HTML page is delivered.

Package Name: `apache2-mod_tidy`
Configuration File: `/etc/apache2/mod_tidy.conf`

More Information: `/usr/share/doc/packages/apache2-mod_tidy`

20.4.6 Compilation

Apache can be extended by advanced users by writing custom modules. To develop modules for Apache or compile third-party modules, the package `apache2-devel` is required along with the corresponding development tools. `apache2-devel` also contains the `apxs2` tools, which are necessary for compiling additional modules for Apache.

`apxs2` enables the compilation and installation of modules from source code (including the required changes to the configuration files), which creates *dynamic shared objects* (DSOs) that can be loaded into Apache at runtime.

The `apxs2` binaries are located under `/usr/sbin`:

- `/usr/sbin/apxs2`—suitable for building an extension module that works with any MPM. The installation location is `/usr/lib/apache2`.
- `/usr/sbin/apxs2-prefork`—suitable for prefork MPM modules. The installation location is `/usr/lib/apache2-prefork`.
- `/usr/sbin/apxs2-worker`—suitable for worker MPM modules. The installation location is `/usr/lib/apache2-worker`.

Install and activate a module from source code with the following commands:

```
cd /path/to/module/source; apxs2 -cia  
mod_foo.c
```

where `-c` compiles the module, `-i` installs it, and `-a` activates it. Other options of `apxs2` are described in the `apxs2(1)` man page.

20.5 Getting CGI Scripts to Work

Apache's Common Gateway Interface (CGI) lets you create dynamic content with programs or scripts usually referred to as CGI scripts. CGI scripts can be written in any programming language. Usually, script languages such as Perl or PHP are used.

To enable Apache to deliver content created by CGI scripts, `mod_cgi` needs to be activated. `mod_alias` is also needed. Both modules are enabled by default. Refer to

Section 20.4.2, “Activation and Deactivation” (page 368) for details on activating modules.

WARNING: CGI Security

Allowing the server to execute CGI scripts is a potential security hole. Refer to Section 20.7, “Avoiding Security Problems” (page 384) for additional information.

20.5.1 Apache Configuration

In openSUSE, the execution of CGI scripts is only allowed in the directory `/srv/www/cgi-bin/`. This location is already configured to execute CGI scripts. If you have created a virtual host configuration (see Section 20.2.2.1, “Virtual Host Configuration” (page 353)) and want to place your scripts in a host-specific directory, you must unlock and configure this directory.

Example 20.5: *VirtualHost CGI Configuration*

```
ScriptAlias /cgi-bin/ "/srv/www/www.example.com/cgi-bin/"❶

<Directory "/srv/www/www.example.com/cgi-bin/">
    Options +ExecCGI❷
    AddHandler cgi-script .cgi .pl❸
    Order allow,deny❹
    Allow from all
</Directory>
```

- ❶ Tells Apache to handle all files within this directory as CGI scripts.
- ❷ Enables CGI script execution
- ❸ Tells the server to treat files with the extensions `.pl` and `.cgi` as CGI scripts. Adjust according to your needs.
- ❹ The `Order` and `Allow` directives control the default access state and the order in which `allow` and `deny` directives are evaluated. In this case “allow” statements are evaluated before “deny” statements and universal access is enabled.

20.5.2 Running an Example Script

CGI programming differs from “regular” programming in that the CGI programs and scripts must be preceded by a MIME-Type header such as `Content-type: text/html`. This header is sent to the client, so it understands what kind of content

it receives. Secondly, the script's output must be something the client, usually a Web browser, understands—HTML in most cases or plain text or images, for example.

A simple test script available under `/usr/share/doc/packages/apache2/test-cgi` is part of the Apache package. It outputs the content of some environment variables as plain text. Copy this script to either `/srv/www/cgi-bin/` or the script directory of your virtual host (`/srv/www/www.example.com/cgi-bin/`) and name it `test.cgi`.

Files accessible by the Web server should be owned by the user `root`. For additional information see Section 20.7, “Avoiding Security Problems” (page 384). Because the Web server runs with a different user, the CGI scripts must be world-executable and world-readable. Change into the CGI directory and use the command `chmod 755 test.cgi` to apply the proper permissions.

Now call `http://localhost/cgi-bin/test.cgi` or `http://www.example.com/cgi-bin/test.cgi`. You should see the “CGI/1.0 test script report”.

20.5.3 CGI Troubleshooting

If you do not see the output of the test program but an error message instead, check the following:

CGI Troubleshooting

- Have you reloaded the server after having changed the configuration? Check with `rcapache2 probe`.
- If you have configured your custom CGI directory, is it configured properly? If in doubt, try the script within the default CGI directory `/srv/www/cgi-bin/` and call it with `http://localhost/cgi-bin/test.cgi`.
- Are the file permissions correct? Change into the CGI directory and execute `ls -l test.cgi`. Its output should start with

```
-rwxr-xr-x  1 root root
```
- Make sure that the script does not contain programming errors. If you have not changed `test.cgi`, this should not be the case, but if you are using your own programs, always make sure that they do not contain programming errors.

20.6 Setting Up a Secure Web Server with SSL

Whenever sensitive data, such as credit card information, is transferred between Web server and client, it is desirable to have a secure, encrypted connection with authentication. `mod_ssl` provides strong encryption using the secure sockets layer (SSL) and transport layer security (TLS) protocols for HTTP communication between a client and the Web server. Using SSL/TSL, a private connection between Web server and client is established. Data integrity is ensured and client and server are able to authenticate each other.

For this purpose, the server sends an SSL certificate that holds information proving the server's valid identity before any request to a URL is answered. In turn, this guarantees that the server is the uniquely correct end point for the communication. Additionally, the certificate generates an encrypted connection between client and server that can transport information without the risk of exposing sensitive, plain-text content.

`mod_ssl` does not implement the SSL/TSL protocols itself, but acts as an interface between Apache and an SSL library. In openSUSE, the OpenSSL library is used. OpenSSL is automatically installed with Apache.

The most visible effect of using `mod_ssl` with Apache is that URLs are prefixed with `https://` instead of `http://`.

TIP: Example Certificate

An example certificate for a hypothetical company “Snake Oil” is available when installing the package `apache2-example-certificates`.

20.6.1 Creating an SSL Certificate

In order to use SSL/TSL with the Web server, you need to create an SSL certificate. This certificate is needed for the authorization between Web server and client, so that each party can clearly identify the other party. To ensure the integrity of the certificate, it must be signed by a party every user trusts.

There are three types of certificates you can create: a “dummy” certificate for testing purposes only, a self-signed certificate for a defined circle of users that trust you, and a certificate signed by an independent, publicly-known certificate authority (CA).

Creating a certificate is basically a two step process. First, a private key for the certificate authority is generated then the server certificate is signed with this key.

TIP: For More Information

To learn more about concepts and definitions of SSL/TSL, refer to http://httpd.apache.org/docs/2.2/ssl/ssl_intro.html.

20.6.1.1 Creating a “Dummy” Certificate

Generating a dummy certificate is simple. Just call the script `/usr/bin/gensslcert`. It creates or overwrites the files listed below. Make use of `gensslcert`'s optional switches to fine-tune the certificate. Call `/usr/bin/gensslcert -h` for more information.

- `/etc/apache2/ssl.crt/ca.crt`
- `/etc/apache2/ssl.crt/server.crt`
- `/etc/apache2/ssl.key/server.key`
- `/etc/apache2/ssl.csr/server.csr`
- `/root/.mkcert.cfg`

A copy of `ca.crt` is also placed at `/srv/www/htdocs/CA.crt` for download.

IMPORTANT: For Testing Purposes Only

A dummy certificate should never be used on a production system. Only use it for testing purposes.

20.6.1.2 Creating a Self-Signed Certificate

If you are setting up a secure Web server for an Intranet or for a defined circle of users, it might be sufficient if you sign a certificate with your own certificate authority (CA).

Creating a self-signed certificate is an interactive nine-step process. Change into the directory `/usr/share/doc/packages/apache2` and run the following command: `./mkcert.sh make --no-print-directory /usr/bin/openssl /usr/sbin/ custom`. Do not attempt to run this command from outside this directory. The program provides a series of prompts, some of which require user input.

Procedure 20.4: *Creating a Self-Signed Certificate with mkcert.sh*

1 Decide the signature algorithm used for certificates

Choose RSA (R, the default), because some older browsers have problems with DSA.

2 Generating RSA private key for CA (1024 bit)

No interaction needed.

3 Generating X.509 certificate signing request for CA

Create the CA's distinguished name here. This requires you to answer a few questions, such as country name or organization name. Enter valid data, because everything you enter here later shows up in the certificate. You do not need to answer every question. If one does not apply to you or you want to leave it blank, use “.”. Common name is the name of the CA itself—choose a significant name, such as *My company CA*.

IMPORTANT: Common Name of the CA

The common name of the CA must be different from the server's common name, so do not choose the fully qualified hostname in this step.

4 Generating X.509 certificate for CA signed by itself

Choose certificate version 3 (the default).

5 Generating RSA private key for SERVER (1024 bit)

No interaction needed.

6 Generating X.509 certificate signing request for SERVER

Create the distinguished name for the server key here. Questions are almost identical to the ones already answered for the CA's distinguished name. The data entered here applies to the Web server and does not necessarily need to be identical to the CA's data (for example, if the server is located elsewhere).

IMPORTANT: Selecting a Common Name

The common name you enter here must be the fully qualified hostname of your secure server (for example, `www.example.com`). Otherwise the browser issues a warning that the certificate does not match the server when accessing the Web server.

7 `Generating X.509 certificate signed by own CA`

Choose certificate version 3 (the default).

8 `Encrypting RSA private key of CA with a passphrase for security`

It is strongly recommended to encrypt the private key of the CA with a password, so choose Y and enter a password.

9 `Encrypting RSA private key of SERVER with a passphrase for security`

Encrypting the server key with a password requires you to enter this password every time you start the Web server. This makes it difficult to automatically start the server on boot or to restart the Web server. Therefore, it is common sense to say N to this question. Keep in mind that your key is unprotected when not encrypted with a password and make sure that only authorized persons have access to the key.

IMPORTANT: Encrypting the Server Key

If you choose to encrypt the server key with a password, increase the value for `APACHE_TIMEOUT` in `/etc/sysconfig/apache2`. Otherwise you do not have enough time to enter the passphrase before the attempt to start the server is stopped unsuccessfully.

The script's result page presents a list of certificates and keys it has generated. Contrary to what the script outputs, the files have not been generated in the local directory `conf`, but to the correct locations under `/etc/apache2/`.

The last step is to copy the CA certificate file from `/etc/apache2/ssl.crt/ca.crt` to a location where your users can access it in order to incorporate it into the list of known and trusted CAs in their Web browsers. Otherwise a browser complains that the certificate was issued by an unknown authority. The certificate is valid for one year.

IMPORTANT: Self-Signed Certificates

Only use a self-signed certificate on a Web server that is accessed by people who know and trust you as a certificate authority. It is not recommended to use such a certificate for a public shop, for example.

20.6.1.3 Getting an Officially Signed Certificate

There are a number of official certificate authorities that sign your certificates. The certificate is signed by a trustworthy third party, so can be fully trusted. Publicly operating secure Web servers usually have got an officially signed certificate.

The best-known official CAs are Thawte (<http://www.thawte.com/>) or Verisign (<http://www.verisign.com>). These and other CAs are already compiled into all browsers, so certificates signed by these certificate authorities are automatically accepted by the browser.

When requesting an officially signed certificate, you do not send a certificate to the CA. Instead, issue a Certificate Signing Request (CSR). To create a CSR, call the script `/usr/share/ssl/misc/CA.sh -newreq`.

First the script asks for a password with which the CSR should be encrypted. Then you are asked to enter a distinguished name. This requires you to answer a few questions, such as country name or organization name. Enter valid data—everything you enter here later shows up in the certificate and is checked. You do not need to answer every question. If one does not apply to you or you want to leave it blank, use “.”. Common name is the name of the CA itself—choose a significant name, such as *My company* CA. Last, a challenge password and an alternative company name must be entered.

Find the CSR in the directory from which you called the script. The file is named `newreq.pem`.

20.6.2 Configuring Apache with SSL

The default port for SSL and TLS requests on the Web server side is 443. There is no conflict between a “regular” Apache listening on port 80 and an SSL/TLS-enabled Apache listening on port 443. In fact, HTTP and HTTPS can be run with the same Apache instance. Usually separate virtual hosts are used to dispatch requests to port 80 and port 443 to separate virtual servers.

IMPORTANT: Firewall Configuration

Do not forget to open the firewall for SSL-enabled Apache on port 443. This can be done with YaST as described in Section “Configuring the Firewall with YaST” (Chapter 13, *Masquerading and Firewalls*, ↑Security Guide).

The SSL module is enabled by default in the global server configuration. In case it has been disabled on your host, activate it with the following command: `a2enmod ssl`. To finally enable SSL, the server needs to be started with the flag “SSL”. To do so, call `a2enflag SSL`. If you have chosen to encrypt your server certificate with a password, you should also increase the value for `APACHE_TIMEOUT` in `/etc/sysconfig/apache2`, so you have enough time to enter the passphrase when Apache starts. Restart the server to make these changes active. A reload is not sufficient.

The virtual host configuration directory contains a template `/etc/apache2/vhosts.d/vhost-ssl.template` with SSL-specific directives that are extensively documented. Refer to Section 20.2.2.1, “Virtual Host Configuration” (page 353) for the general virtual host configuration.

To get started, copy the template to `/etc/apache2/vhosts.d/mySSL-host.conf` and edit it. Adjusting the values for the following directives should be sufficient:

- `DocumentRoot`
- `ServerName`
- `ServerAdmin`

- `ErrorLog`
- `TransferLog`

20.6.2.1 Name-Based Virtual Hosts and SSL

By default it is not possible to run multiple SSL-enabled virtual hosts on a server with only one IP address. Name-based virtual hosting requires that Apache knows which server name has been requested. The problem with SSL connections is, that such a request can only be read after the SSL connection has already been established (by using the default virtual host). As a result, users will receive a warning message stating that the certificate does not match the server name.

openSUSE comes with an extension to the SSL protocol called Server Name Indication (SNI) addresses this issue by sending the name of the virtual domain as part of the SSL negotiation. This enables the server to “switch” to the correct virtual domain early and present the browser the correct certificate.

SNI is enabled by default on openSUSE. In order to enable Name-Based Virtual Hosts for SSL, configure the server as described in Section “Name-Based Virtual Hosts” (page 354) (note that you need to use port 443 rather than port 80 with SSL).

IMPORTANT: SNI Browser Support

SNI must also be supported on the client side. Although SNI is supported by most browsers, some browsers for mobile hardware as well as Internet Explorer and Safari on Windows* XP lack SNI support. See http://en.wikipedia.org/wiki/Server_Name_Indication for details.

Configure how to handle non-SNI capable browser with the directive `SSLStrictSNIVHostCheck`. When set to `on` in the server configuration, non-SNI capable browser will be rejected for all virtual hosts. When set to `on` within a `VirtualHost` directive, access to this particular Host will be rejected.

When set to `off` in the server configuration, the server will behave as if not having SNI support. SSL requests will be handled by the *first* Virtual host defined (for port 443).

20.7 Avoiding Security Problems

A Web server exposed to the public Internet requires an ongoing administrative effort. It is inevitable that security issues appear, both related to the software and to accidental misconfiguration. Here are some tips for how to deal with them.

20.7.1 Up-to-Date Software

If there are vulnerabilities found in the Apache software, a security advisory will be issued by SUSE. It contains instructions for fixing the vulnerabilities, which in turn should be applied as soon as possible. The SUSE security announcements are available from the following locations:

- **Web Page** <http://www.novell.com/linux/security/securitysupport.html>
- **Mailing List Archive** <http://lists.opensuse.org/opensuse-security-announce/>
- **RSS Feed** http://www.novell.com/linux/security/suse_security.xml

20.7.2 DocumentRoot Permissions

By default in openSUSE, the `DocumentRoot` directory `/srv/www/htdocs` and the CGI directory `/srv/www/cgi-bin` belong to the user and group `root`. You should not change these permissions. If the directories are writable for all, any user can place files into them. These files might then be executed by Apache with the permissions of `wwwrun`, which may give the user unintended access to file system resources. Use subdirectories of `/srv/www` to place the `DocumentRoot` and CGI directories for your virtual hosts and make sure that directories and files belong to user and group `root`.

20.7.3 File System Access

By default, access to the whole file system is denied in `/etc/apache2/httpd.conf`. You should never overwrite these directives, but specifically enable

access to all directories Apache should be able to read. For details, see Section “Basic Virtual Host Configuration” (page 356). In doing so, ensure that no critical files, such as password or system configuration files, can be read from the outside.

20.7.4 CGI Scripts

Interactive scripts in Perl, PHP, SSI, or any other programming language can essentially run arbitrary commands and therefore present a general security issue. Scripts that will be executed from the server should only be installed from sources the server administrator trusts—allowing users to run their own scripts is generally not a good idea. It is also recommended to do security audits for all scripts.

To make the administration of scripts as easy as possible, it is common practice to limit the execution of CGI scripts to specific directories instead of globally allowing them. The directives `ScriptAlias` and `Option ExecCGI` are used for configuration. The openSUSE default configuration does not allow execution of CGI scripts from everywhere.

All CGI scripts run as the same user, so different scripts can potentially conflict with each other. The module `suEXEC` lets you run CGI scripts under a different user and group.

20.7.5 User Directories

When enabling user directories (with `mod_userdir` or `mod_rewrite`) you should strongly consider not allowing `.htaccess` files, which would allow users to overwrite security settings. At least you should limit the user's engagement by using the directive `AllowOverride`. In openSUSE, `.htaccess` files are enabled by default, but the user is not allowed to overwrite any `Option` directives when using `mod_userdir` (see the `/etc/apache2/mod_userdir.conf` configuration file).

20.8 Troubleshooting

If Apache does not start, the Web page is not accessible, or users cannot connect to the Web server, it is important to find the cause of the problem. Here are some typical places to look for error explanations and important things to check:

Output of `rcapache2`

Instead of starting and stopping the Web server with the binary `/usr/sbin/httpd2`, rather use the `rcapache2` script instead (described in Section 20.3, “Starting and Stopping Apache” (page 364)). It is verbose about errors, and it even provides tips and hints for fixing configuration errors.

Log Files and Verbosity

In case of both fatal and nonfatal errors, check the Apache log files for causes, mainly the error log file located at `/var/log/apache2/error_log` by default. Additionally, you can control the verbosity of the logged messages with the `LogLevel` directive if more detail is needed in the log files.

TIP: A Simple Test

Watch the Apache log messages with the command `tail -F /var/log/apache2/my_error_log`. Then run `rcapache2 restart`. Now, try to connect with a browser and check the output.

Firewall and Ports

A common mistake is to not open the ports for Apache in the firewall configuration of the server. If you configure Apache with YaST, there is a separate option available to take care of this specific issue (see Section 20.2.3, “Configuring Apache with YaST” (page 357)). If you are configuring Apache manually, open firewall ports for HTTP and HTTPS via YaST's firewall module.

If the error cannot be tracked down with the help of any these, check the online Apache bug database at http://httpd.apache.org/bug_report.html. Additionally, the Apache user community can be reached via a mailing list available at <http://httpd.apache.org/userslist.html>. A recommended newsgroup is comp.infosystems.www.servers.unix.

20.9 For More Information

The package `apache2-doc` contains the complete Apache manual in various localizations for local installation and reference. It is not installed by default—the quick-way to install it is to use the command `zypper in apache2-doc`. Once installed, the Apache manual is available at <http://localhost/manual/>. You may also access it on the Web at <http://httpd.apache.org/docs-2.2/>.

SUSE-specific configuration hints are available in the directory `/usr/share/doc/packages/apache2/README.*`.

20.9.1 Apache 2.2

For a list of new features in Apache 2.2, refer to http://httpd.apache.org/docs/2.2/new_features_2_2.html. Information about upgrading from version 2.0 to 2.2 is available at <http://httpd.apache.org/docs-2.2/upgrading.html>.

20.9.2 Apache Modules

More information about external Apache modules that are briefly described in Section 20.4.5, “External Modules” (page 372) is available at the following locations:

`mod-apparmor`

<http://en.opensuse.org/SDB:AppArmor>

`mod_mono`

http://www.mono-project.com/Mod_mono

`mod_perl`

<http://perl.apache.org/>

`mod_php5`

<http://www.php.net/manual/en/install.unix.apache2.php>

`mod_python`

<http://www.modpython.org/>

`mod_tidy`

<http://mod-tidy.sourceforge.net/>

20.9.3 Development

More information about developing Apache modules or about getting involved in the Apache Web server project are available at the following locations:

Apache Developer Information

<http://httpd.apache.org/dev/>

Apache Developer Documentation

<http://httpd.apache.org/docs/2.2/developer/>

Writing Apache Modules with Perl and C

<http://www.modperl.com/>

20.9.4 Miscellaneous Sources

If you experience difficulties specific to Apache in openSUSE, take a look at the openSUSE wiki at <http://old-en.opensuse.org/Apache>. The history of Apache is provided at http://httpd.apache.org/ABOUT_APACHE.html. This page also explains why the server is called Apache.

Setting up an FTP server with YaST

Using the YaST *FTP Server* module, you can configure your machine to function as an FTP (File Transfer Protocol) server. Anonymous and/or authenticated users can connect to your machine and download files using the FTP protocol. Depending on the configuration, they can also upload files to the FTP server. YaST provides a unified configuration interface for various FTP server daemons installed on your system.

You can use the YaST *FTP Server* configuration module to configure two different FTP server daemons:

- `vsftpd` (Very Secure FTP Daemon) and
- `pure-ftpd`

Only installed servers can be configured. Standard openSUSE® media do not contain the `pure-ftpd` package. However, if the `pure-ftpd` package is installed from another repository, it can be configured using the YaST module.

The `vsftpd` and `pure-ftpd` servers have slightly different configuration options, especially in the *Experts Settings* dialog. This chapter describes the settings of the `vsftpd` server, being the default server for openSUSE.

If the YaST FTP Server module is not available in your system, install the `yast2-ftp-server` package.

To configure the FTP server using YaST, follow these steps:

- 1 Open YaST Control Center and choose *Network Services > FTP Server* or run the `yast2 ftp-server` command as `root`.

- 2 If there is not any FTP server installed in your system, you will be asked which server to install when the YaST FTP Server module starts. Choose a server (vsftpd is the standard server for openSUSE) and confirm the dialog. If there are two servers installed, choose the preferred server and click *OK*.
- 3 In the *Start-Up* dialog, configure the options for starting of the FTP server. For more information, see Section 21.1, “Starting the FTP server” (page 390).

In the *General* dialog, configure FTP directories, welcome message, file creation masks and various other parameters. For more information, see Section 21.2, “FTP General Settings” (page 391).

In the *Performance* dialog, set the parameters that affect the load on the FTP server. For more information, see Section 21.3, “FTP Performance Settings” (page 392).

In the *Authentication* dialog, set whether the FTP server should be available for anonymous and/or authenticated users. For more information, see Section 21.4, “Authentication” (page 392).

In the *Expert Settings* dialog, configure the operation mode of the FTP server, SSL connections and firewall settings. For more information, see Section 21.5, “Expert Settings” (page 393).

- 4 Press *Finish* to save the configurations.

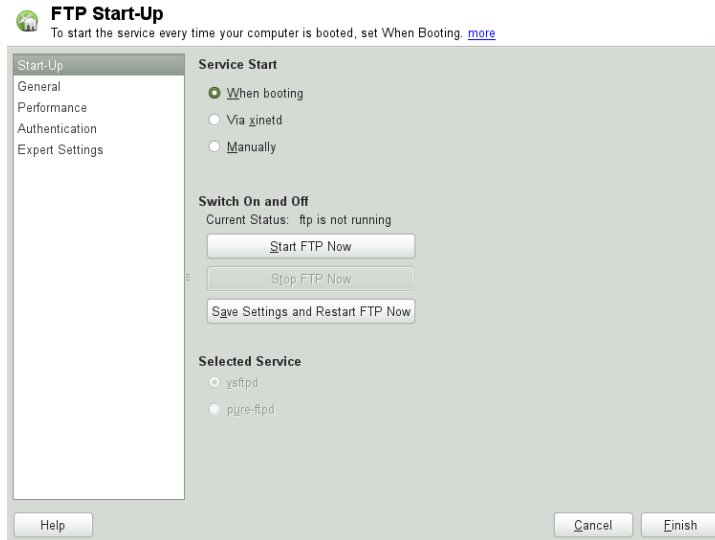
21.1 Starting the FTP server

In the *Service Start* frame of the *FTP Start-Up* dialog set the way the FTP server is started up. You can choose between starting the server automatically during the system boot and starting it manually. If the FTP server should be started only after an FTP connection request, choose *Via xinetd*.

The current status of the FTP server is shown in the *Switch On and Off* frame of the *FTP Start-Up* dialog. Start the FTP server by clicking *Start FTP Now*. To stop the server, click *Stop FTP Now*. After having changed the settings of the server click *Save Settings and Restart FTP Now*. Your configurations will be saved by leaving the configuration module with *Finish*.

The *Selected Service* frame of the *FTP Start-Up* dialog shows which FTP server is used: either vsftpd or pure-ftpd. If both servers are installed, you can switch between them—the current configuration will automatically be converted. The pure-ftpd package is not included in the standard openSUSE media so you have to install it from a different installation source if you want to use it.

Figure 21.1: *FTP Server Configuration — Start-Up*



21.2 FTP General Settings

In the *General Settings* frame of the *FTP General Settings* dialog you can set the *Welcome message* which is shown after connecting to the FTP server.

If you check the *Chroot Everyone* option, all local users will be placed in a chroot jail in their home directory after login. This option has security implications, especially if the users have upload permission or shell access, so be careful enabling this option.

If you check the *Verbose Logging* option, all FTP requests and responses are logged.

You can limit permissions of files created by anonymous and/or authenticated users with umask. Set the file creation mask for anonymous users in *Umask for Anonymous*

and the file creation mask for authenticated users in *Umask for Authenticated Users*. The masks should be entered as octal numbers with a leading zero. For more information about umask, see the umask man page (`man 1p umask`).

In the *FTP Directories* frame set the directories used for anonymous and authorized users. With *Browse*, you can select a directory to be used from the local filesystem. The default FTP directory for anonymous users is `/srv/ftp`. Note that vsftpd does not allow this directory to be writable for all users. The subdirectory `upload` with write permissions for anonymous users is created instead.

NOTE: Write Permissions in FTP Directory

The pure-ftpd server allows the FTP directory for anonymous users to be writable. When switching between servers, make sure you remove the write permissions in the directory that was used with pure-ftpd before switching back to the vsftpd server.

21.3 FTP Performance Settings

In the *Performance* dialog set the parameters which affect the load on the FTP server. *Max Idle Time* is the maximum time (in minutes) the remote client may spend between FTP commands. In case of longer inactivity, the remote client is disconnected. *Max Clients for One IP* determines the maximum number of clients which can be connected from a single IP address. *Max Clients* determines the maximum number of clients which may be connected. Any additional clients will get an error message.

The maximum data transfer rate (in KB/s) is set in *Local Max Rate* for local authenticated users, and in *Anonymous Max Rate* for anonymous clients respectively. The default value for the rate settings is 0, which means unlimited data transfer rate.

21.4 Authentication

In the *Enable/Disable Anonymous and Local Users* frame of the *Authentication* dialog, you are able to set which users are allowed to access your FTP server. You can choose between the following options: granting access to anonymous users only, to authenticated users only (with accounts on the system) or to both types of users.

If you want to allow users to upload files to the FTP server, check *Enable Upload* in the *Uploading* frame of the *Authentication* dialog. Here you are able to allow uploading or creating directories even for anonymous users by checking the respective box.

NOTE: vsftpd—Allowing File Upload for Anonymous Users

If a vsftpd server is used and you want anonymous users to be able to upload files or create directories, a subdirectory with writing permissions for all users has to be created in the anonymous FTP directory.

21.5 Expert Settings

An FTP server can run in active or in passive mode. By default the server runs in passive mode. To switch into active mode, just uncheck *Enable Passive Mode* option in *Expert Settings* dialog. You can also change the range of ports on the server used for the data stream by tweaking the *Min Port for Pas. Mode* and *Max Port for Pas. Mode* options.

If you want encrypted communication between clients and the server, you can *Enable SSL*. Check the versions of the protocol to be supported and specify the DSA certificate to be used for SSL encrypted connections.

If your system is protected by a firewall, check *Open Port in Firewall* to enable a connection to the FTP server.

21.6 For more information

For more information about FTP servers read the manual pages of `vsftpd` and `vsftpd.conf`.

Part V. Mobility

Mobile Computing with Linux

22

Mobile computing is mostly associated with laptops, PDAs and cellular phones (and the data exchange between them). Mobile hardware components, such as external hard disks, flash drives, or digital cameras, can be connected to laptops or desktop systems. A number of software components are involved in mobile computing scenarios and some applications are tailor-made for mobile use.

22.1 Laptops

The hardware of laptops differs from that of a normal desktop system. This is because criteria like exchangeability, space requirements and power consumption must be taken into account. The manufacturers of mobile hardware have developed standard interfaces like PCMCIA (Personal Computer Memory Card International Association), Mini PCI and Mini PCIe that can be used to extend the hardware of laptops. The standards cover memory cards, network interface cards, ISDN (and modem cards) and external hard disks.

TIP: openSUSE and Tablet PCs

openSUSE also supports Tablet PCs. Tablet PCs come with a touchpad/digitizer that allows you to use a digital pen or even fingertips to edit data right on the screen instead of using mouse and keyboard. They are installed and configured much like any other system. For a detailed introduction to the installation and configuration of Tablet PCs, refer to Chapter 26, *Using Tablet PCs* (page 455).

22.1.1 Power Conservation

The inclusion of energy-optimized system components during laptop manufacturing 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. openSUSE® 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 in descending order of contribution towards power conservation:

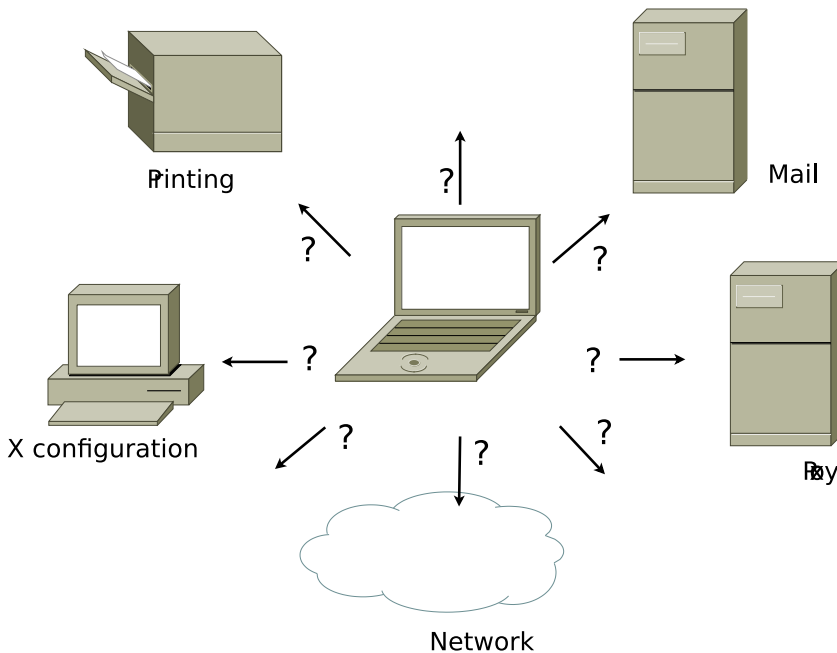
- Throttling the CPU speed.
- Switching off the display illumination during pauses.
- Manually adjusting the display illumination.
- Disconnecting unused, hotplug-enabled accessories (USB CD-ROM, external mouse, unused PCMCIA cards, WLAN, etc.).
- Spinning down the hard disk when idling.

Detailed background information about power management in openSUSE is provided in Chapter 23, *Power Management* (page 407).

22.1.2 Integration in Changing Operating Environments

Your system needs to adapt to changing operating environments when used for mobile computing. Many services depend on the environment and the underlying clients must be reconfigured. openSUSE handles this task for you.

Figure 22.1: *Integrating a Mobile Computer in an Existing Environment*



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

Network

This includes IP address assignment, 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.

X (Graphical Environment)

If your laptop is temporarily connected to a projector or an external monitor, different display configurations must be available.

openSUSE offers several ways of integrating laptops into existing operating environments:

NetworkManager

NetworkManager is especially tailored for mobile networking on laptops. It provides a means to easily and automatically switch between network environments or different types of networks such as mobile broadband (such as GPRS, EDGE, or 3G), wireless LAN, and Ethernet. NetworkManager supports WEP and WPA-PSK encryption in wireless LANs. It also supports dial-up connections (with smpppd). Both desktop environments (GNOME and KDE) include a front-end for NetworkManager. For more information about the desktop applets, see Section 25.4, “Using the KDE NetworkManager Front-End” (page 441) and Section 25.5, “Using GNOME NetworkManager” (page 444).

Table 22.1: *Use Cases for NetworkManager*

My computer...	Use NetworkManager
is a laptop	Yes
is sometimes attached to different networks	Yes
provides network services (such as DNS or DHCP)	No
only uses a static IP address	No

Use the YaST tools to configure networking whenever NetworkManager should not handle network configuration.

TIP: DNS configuration and various types of network connections

If you travel frequently with your laptop and change different types of network connections, NetworkManager works fine when all DNS addresses are assigned correctly assigned with DHCP. If some of your connections use static DNS address(es), add it to the `NETCONFIG_DNS_STATIC_SERVERS` option in `/etc/sysconfig/network/config`.

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 de-

tailed 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 also be used to install a system, minimizing the effort of searching for a suitable installation source. Find detailed information about SLP in Chapter 14, *SLP Services in the Network* (page 275).

22.1.3 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 openSUSE provides for each task.

22.1.3.1 System Monitoring

Two KDE system monitoring tools are provided by openSUSE:

Power Management

Power Management is an application which lets you adjust energy saving related behavior of the KDE desktop. You can typically access it via the *Battery Monitor* tray icon, which changes according to the type of the current power supply. Other way to open its configuration dialog is through the *Kickoff Application Launcher*: *Applications > Configure Desktop > Advanced > Power Management*.

Click the *Battery Monitor* tray icon to access options to configure its behavior. You can choose one of five displayed power profiles which best fits your needs. For example, the *Presentation* scheme disables the screen saver and the power management in general, so that your presentation is not interrupted by system events. Click *More...* to open a more complex configuration screen. Here you can edit individual profiles and set advanced power management options and notifications, such as what to do when the laptop lid has been closed, or when the battery charge is low.

System Monitor

System Monitor (also called *KSysguard*) gathers measurable system parameters into one monitoring environment. It presents the output information in 2 tabs by default. *Process Table* gives detailed information about currently running processes, such as CPU load, memory usage, or process ID number and nice value. The

presentation and filtering of the collected data can be customized — to add a new type of process information, left-click on the process table header and choose which column to hide or add to the view. It is also 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. Find more information about this program in its integrated help function or in the SUSE help pages.

In the GNOME environment use *Power Management Preferences* and *System Monitor*.

22.1.3.2 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 processed data synchronized 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:

Synchronizing E-Mail

Use an IMAP account for storing your e-mails in the office network. Then access the e-mails from the workstation using any disconnected IMAP-enabled e-mail client, like Mozilla Thunderbird Mail, Evolution, or KMail. The e-mail client must be configured so that the same folder is always accessed for `Sent` messages. This ensures that all messages are available along with their status information after the synchronization process has completed. Use an SMTP server 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. One of the most widely used is a command-line tool called `rsync`. For more information, see Chapter 27, *Copying and Sharing Files* (page 465).

22.1.3.3 Wireless Communication

As well as connecting to a home or office network with a cable, a laptop can also use wireless connection to access 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 separate 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 act as intermediaries for access to the Internet. A mobile user can switch among 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. Find details about WLAN in Chapter 24, *Wireless LAN* (page 417).

Bluetooth

Bluetooth has the broadest application spectrum of all wireless technologies. It can be used for communication between computers (laptops) and PDAs or cellular phones, as can IrDA. It can also be used to connect various computers within range. Bluetooth is also used to connect wireless system components, like a keyboard or a 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.

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.

22.1.4 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.

Strong Authentication

Use biometric authentication in addition to standard authentication via login and password. openSUSE supports fingerprint authentication. For more details, see Chapter 7, *Using the Fingerprint Reader* (↑Security Guide).

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 openSUSE is described in Chapter 10, *Encrypting Partitions and Files* (↑Security Guide). Another possibility is to create encrypted home directories when adding the user with YaST.

IMPORTANT: Data Security and Suspend to Disk

Encrypted partitions are not unmounted during a suspend to disk event. Thus, all data on these partitions is available to any party who manages to steal the hardware and issue a resume of the hard disk.

Network Security

Any transfer of data should be secured, no matter how the transfer is done. Find general security issues regarding Linux and networks in Chapter 1, *Security and Confidentiality* (↑Security Guide). Security measures related to wireless networking are provided in Chapter 24, *Wireless LAN* (page 417).

22.2 Mobile Hardware

openSUSE supports the automatic detection of mobile storage devices over FireWire (IEEE 1394) or USB. The term *mobile storage device* applies to any kind of FireWire or USB hard disk, USB flash drive, or digital camera. These devices are automatically detected and configured as soon as they are connected with the system over the corresponding interface. The file managers of both GNOME and KDE offer flexible handling of mobile hardware items. To unmount any of these media safely, use the *Safely Remove* (KDE) or *Unmount* (GNOME) feature of either file manager.

External Hard Disks (USB and FireWire)

As soon as an external hard disk is correctly recognized by the system, its icon appears in the file manager. 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` 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 `f-spot`. For advanced photo processing, use `GIMP`.

22.3 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 Section 22.1.3.3, “Wireless Communication” (page 402). The configuration of these protocols on the cellular phones themselves is described in their manuals.

The support for synchronizing with handheld devices manufactured by Palm, Inc., is already built into `Evolution` and `Kontact`. Initial connection with the device is 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.).

A more sophisticated synchronization solution is available with the program `open-sync` (see the package `libopensync` and the respective plug-ins for the various devices).

22.4 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 hard-

ware 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.opensuse.org/opensuse-mobile-de/>. On this list, users and developers discuss all aspects of mobile computing with openSUSE. Postings in English are answered, but the majority of the archived information is only available in German. Use <http://lists.opensuse.org/opensuse-mobile/> for English postings.

Information about OpenSync is available on <http://en.opensuse.org/OpenSync>.

Power Management

Power management is especially important on laptop computers, but is also useful on other systems. ACPI (Advanced Configuration and Power Interface) is available on all modern computers (laptops, desktops, and servers). Power management technologies require suitable hardware and BIOS routines. Most laptops and many modern desktops and servers meet these requirements. It is also possible to control CPU frequency scaling to save power or decrease noise.

23.1 Power Saving Functions

Power saving functions are not only significant for the mobile use of laptops, but also for desktop systems. The main functions and their use in ACPI are:

Standby

not supported.

Suspend (to memory)

This mode writes the entire system state to the RAM. Subsequently, the entire system except the RAM is put to sleep. In this state, the computer consumes very little power. The advantage of this state is the possibility of resuming work at the same point within a few seconds without having to boot and restart applications. This function corresponds to the ACPI state S3. The support of this state is still under development and therefore largely depends on the hardware.

Hibernation (suspend to disk)

In this operating mode, the entire system state is written to the hard disk and the system is powered off. There must be a swap partition at least as big as the RAM to write all the active data. Reactivation from this state takes about 30 to 90 seconds. The state prior to the suspend is restored. Some manufacturers offer useful hybrid variants of this mode, such as RediSafe in IBM Thinkpads. The corresponding ACPI state is S4. In Linux, suspend to disk is performed by kernel routines that are independent from ACPI.

Battery Monitor

ACPI checks the battery charge status and provides information about it. Additionally, it coordinates actions to perform when a critical charge status is reached.

Automatic Power-Off

Following a shutdown, the computer is powered off. This is especially important when an automatic shutdown is performed shortly before the battery is empty.

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.

23.2 Advanced Configuration and Power Interface (ACPI)

ACPI was designed to enable the operating system to set up and control the individual hardware components. ACPI supersedes both Power Management Plug and Play (PnP) and Advanced Power Management (APM). It delivers information about the battery, AC adapter, temperature, fan and system events, like “close lid” or “battery low.”

The BIOS provides tables containing information about the individual components and hardware access methods. The operating system uses this information for tasks like assigning interrupts or activating and deactivating components. Because the operating system executes commands stored in the BIOS, the functionality depends on the BIOS implementation. The tables ACPI can detect and load are reported in `/var/log/boot.msg`. See Section 23.2.2, “Troubleshooting” (page 409) for more information about troubleshooting ACPI problems.

23.2.1 Controlling the CPU Performance

The CPU can save energy in three ways:

- Frequency and Voltage Scaling
- Throttling the Clock Frequency (T-states)
- Putting the Processor to Sleep (C-states)

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 scaling and throttling are only relevant if the processor is busy, because the most economic C-state is applied anyway when the processor is idle. If the CPU is busy, frequency scaling is the recommended power saving method. Often the processor only works with a partial load. In this case, it can be run with a lower frequency. Usually, dynamic frequency scaling controlled by the kernel on-demand governor is the best approach.

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.

For in-depth information, refer to Chapter 11, *Power Management* (↑System Analysis and Tuning Guide).

23.2.2 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, 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 of other widespread operating systems. Hardware components that have serious errors in the ACPI implementation are

recorded in a blacklist that prevents the Linux kernel from using ACPI for these components.

The first thing to do when problems are encountered is to update the BIOS. If the computer does not boot at all, one of the following boot parameters may be helpful:

`pci=noacpi`

Do not use ACPI for configuring the PCI devices.

`acpi=ht`

Only perform a simple resource configuration. Do not use ACPI for other purposes.

`acpi=off`

Disable ACPI.

WARNING: Problems Booting without ACPI

Some newer machines (especially SMP systems and AMD64 systems) need ACPI for configuring the hardware correctly. On these machines, disabling ACPI can cause problems.

Sometimes, the machine is confused by hardware that is attached over USB or FireWire. If a machine refuses to boot, unplug all unneeded hardware and try again.

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 (*Differentiated System Description Table*)—can be replaced with an improved version. In this case, the faulty DSDT of the BIOS is ignored. The procedure is described in Section 23.4.1, “ACPI Activated with Hardware Support but Functions Do Not Work” (page 413).

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.

23.2.2.1 For More Information

- <http://tldp.org/HOWTO/ACPI-HOWTO/> (detailed ACPI HOWTO, contains DSDT patches)
- <http://www.acpi.info> (Advanced Configuration & Power Interface Specification)
- <http://www.lesswatts.org/projects/acpi/> (the ACPI4Linux project at Sourceforge)
- <http://acpi.sourceforge.net/dsdt/index.php> (DSDT patches by Bruno Ducrot)

23.3 Rest for the Hard Disk

In Linux, the hard disk can be put to sleep entirely if it is not needed or it can be run in a more economic or quieter mode. On modern laptops, you do not need to switch off the hard disks manually, because they automatically enter an economic operating mode whenever they are not needed. However, if you want to maximize power savings, test some of the following methods, using the `hdparm` command.

It can be used to modify various hard disk settings. The option `-y` instantly switches the hard disk to the standby mode. `-Y` puts it to sleep. `hdparm -S x` causes the hard disk to be spun down after a certain period of inactivity. Replace `x` as follows: 0 disables this mechanism, causing the hard disk to run continuously. Values from 1 to 240 are multiplied by 5 seconds. Values from 241 to 251 correspond to 1 to 11 times 30 minutes.

Internal power saving options of the hard disk can be controlled with the option `-B`. Select a value from 0 to 255 for maximum saving to maximum throughput. The result depends on the hard disk used and is difficult to assess. To make a hard disk quieter, use the option `-M`. Select a value from 128 to 254 for quiet to fast.

Often, it is not so easy to put the hard disk to sleep. In Linux, numerous processes write to the hard disk, waking it up repeatedly. Therefore, it is important to understand how Linux handles data that needs to be written to the hard disk. First, all data is buffered in the RAM. This buffer is monitored by the `pdflush` daemon. 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, `pdflush` is set to short intervals to achieve maximum data integrity. It checks the buffer every 5 seconds and writes the data to the hard disk. The following variables are interesting:

`/proc/sys/vm/dirty_writeback_centisecs`

Contains the delay until a `pdflush` thread wakes up (in hundredths of a second).

`/proc/sys/vm/dirty_expire_centisecs`

Defines after which timeframe a dirty page should be written out latest. Default is 3000, which means 30 seconds.

`/proc/sys/vm/dirty_background_ratio`

Maximum percentage of dirty pages until `pdflush` begins to write them. Default is 5%.

`/proc/sys/vm/dirty_ratio`

When the dirty page exceeds this percentage of the total memory, processes are forced to write dirty buffers during their time slice instead of continuing to write.

WARNING: Impairment of the Data Integrity

Changes to the `pdflush` daemon settings endanger the data integrity.

Apart from these processes, journaling file systems, like `ReiserFS`, `Ext3`, `Ext4` and others write their metadata independently from `pdflush`, which also prevents the hard disk from spinning down. To avoid this, a special kernel extension has been developed for mobile devices. To make use of the extension, install the `laptop-mode-tools` package and see `/usr/src/linux/Documentation/laptops/laptop-mode.txt` for details.

Another important factor is the way active programs behave. For example, good editors regularly write hidden backups of the currently modified file to the hard disk, causing the disk to wake up. Features like this can be disabled at the expense of data integrity.

In this connection, the mail daemon `postfix` makes use of the variable `POSTFIX_LAPTOP`. If this variable is set to `yes`, `postfix` accesses the hard disk far less frequently.

23.4 Troubleshooting

All error messages and alerts are logged in the file `/var/log/messages`. The following sections cover the most common problems.

23.4.1 ACPI Activated with Hardware Support but Functions Do Not Work

If you experience problems with ACPI, search the output of `dmesg` for ACPI-specific messages by using the command `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:

Procedure 23.1: *Updating the DSDT Table in the BIOS*

For the procedure below, make sure the following packages are installed: `kernel-source`, `acpica`, and `mkinitrd`.

- 1 Download the DSDT for your system from <http://acpi.sourceforge.net/dsdt/index.php>. Check if the file is decompressed and compiled as shown by the file extension `.aml` (ACPI machine language). If this is the case, continue with step 3.
- 2 If the file extension of the downloaded table is `.asl` (ACPI source language) instead, compile it by executing the following command:

```
iasl -sa file.asl
```
- 3 Copy the (resulting) file `DSDT.aml` to any location (`/etc/DSDT.aml` is recommended).
- 4 Edit `/etc/sysconfig/kernel` and adapt the path to the DSDT file accordingly.

- 5 Start `mkinitrd`. Whenever you install the kernel and use `mkinitrd` to create an `initrd` file, the modified DSDT is integrated and loaded when the system is booted.

23.4.2 CPU Frequency Does Not Work

Refer to the kernel sources to see if your processor is supported. You may need a special kernel module or module option to activate CPU frequency control. If the `kernel-source` package is installed, this information is available in `/usr/src/linux/Documentation/cpu-freq/*`.

23.4.3 Suspend and Standby Do Not Work

ACPI systems may have problems with suspend and standby due to a faulty DSDT implementation (BIOS). If this is the case, update the BIOS.

When the system tries to unload faulty modules, the system is arrested or the suspend event is not triggered. The same can also happen if you do not unload modules or stop services that prevent a successful suspend. In both cases, try to identify the faulty module that prevented the sleep mode. The log file `/var/log/pm-suspend.log` contains detailed information about what is going on and where possible errors are. Modify the `SUSPEND_MODULES` variable in `/usr/lib/pm-utils/defaults` to unload problematic modules prior to a suspend or standby.

Refer to <http://old-en.opensuse.org/Pm-utils> and http://en.opensuse.org/SDB:Suspend_to_RAM to get more detailed information on how to modify the suspend and resume process.

23.5 For More Information

- <http://www.acpi.info> (Advanced Configuration and Power Interface Specification)
- <http://www.lesswatts.org/projects/acpi/> (the ACPI4Linux project at Sourceforge)

- <http://acpi.sourceforge.net/dsdt/index.php> (DSDT patches by Bruno Ducrot)
- http://en.opensuse.org/SDB:Suspend_to_RAM—How to get Suspend to RAM working
- <http://old-en.opensuse.org/Pm-utils>—How to modify the general suspend framework

Wireless LAN

Wireless LANs, or Wireless Local Area Network (WLANs), have become an indispensable aspect of mobile computing. Today, most laptops have built-in WLAN cards. This chapter describes how to set up a WLAN card with YaST, encrypt transmissions, and use tips and tricks. Alternatively, you can configure and manage WLAN access with NetworkManager. For details, refer to Chapter 25, *Using NetworkManager* (page 435).

24.1 WLAN Standards

WLAN cards communicate using the 802.11 standard, prepared by the IEEE organization. Originally, this standard provided for a maximum transmission rate of 2 Mbit/s. Meanwhile, several supplements have been added to increase the data rate. These supplements define details such as the modulation, transmission output, and transmission rates (see Table 24.1, “Overview of Various WLAN Standards” (page 418)). Additionally, many companies implement hardware with proprietary or draft features.

Table 24.1: *Overview of Various WLAN Standards*

Name	Band (GHz)	Maximum Transmission Rate (Mbit/s)	Note
802.11 Legacy	2.4	2	Outdated; virtually no end devices available
802.11a	5	54	Less interference-prone
802.11b	2.4	11	Less common
802.11g	2.4	54	Widespread, backwards-compatible with 11b
802.11n	2.4 and/or 5	300	Common

802.11 Legacy cards are not supported by openSUSE®. Most cards using 802.11a, 802.11b, 802.11g and 802.11n are supported. New cards usually comply with the 802.11n standard, but cards using 802.11g are still available.

24.2 Operating Modes

In wireless networking, various techniques and configurations are used to ensure fast, high-quality, and secure connections. Different operating types suit different setups. It can be difficult to choose the right authentication method. The available encryption methods have different advantages and pitfalls.

Basically, wireless networks can be classified into three network modes:

Managed Mode (Infrastructure Mode), via Access Point

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 through the access point, which may also serve as a connection to an

ethernet. To make sure only authorized stations can connect, various authentication mechanisms (WPA, etc) are used.

Ad-hoc Mode (Peer-to-Peer Network)

Ad-hoc networks do not have an access point. The stations communicate directly with each other, therefore an ad-hoc network is usually faster than a managed network. However, the transmission range and number of participating stations are greatly limited in ad-hoc networks. They also do not support WPA authentication. If you intend to use WPA security, you should not use Ad-Hoc_Mode.

Master Mode

In master mode your network card is used as the access point. It works only if your WLAN card supports this mode. Find out the details of your WLAN card on <http://linux-wless.passys.nl>.

24.3 Authentication

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 (Wired Equivalent Privacy). However, because WEP has proven to be insecure (see Section 24.6.3, “Security” (page 431)), the WLAN industry (joined under the name *Wi-Fi Alliance*) has defined an extension called WPA, which is supposed to eliminate the weaknesses of WEP. The later IEEE 802.11i standard includes WPA and some other authentication and encryption methods. IEEE 802.11i is also referred to as WPA2, because WPA is based on a draft version of 802.11i.

To make sure that only authorized stations can connect, various authentication mechanisms are used in managed networks:

None (Open)

An open system is a system that does not require authentication. Any station can join the network. Nevertheless, WEP encryption can be used, see Section 24.4, “Encryption” (page 421).

Shared Key (according to IEEE 802.11)

In this procedure, the WEP key is used for the authentication. However, this procedure is not recommended, because it makes the WEP key more susceptible to attacks. All an attacker needs to do is to listen long enough to the communication

between the station and the access point. During the authentication process, both sides exchange the same information, once in encrypted form and once in unencrypted form. This makes it possible for the key to be reconstructed with suitable tools. Because this method makes use of the WEP key for the authentication and for the encryption, it does not enhance the security of the network. A station that has the correct WEP key can authenticate, encrypt, and decrypt. A station that does not have the key cannot decrypt received packets. Accordingly, it cannot communicate, regardless of whether it had to authenticate itself.

WPA-PSK (or WPA-Personal, according to IEEE 802.1x)

WPA-PSK (PSK stands for preshared key) works similarly to the Shared Key procedure. All participating stations as well as the access point need the same key. The key is 256 bits in length and is usually entered as a passphrase. This system does not need a complex key management like WPA-EAP and is more suitable for private use. Therefore, WPA-PSK is sometimes referred to as WPA “Home”.

WPA-EAP (or WPA-Enterprise, according to IEEE 802.1x)

Actually, WPA-EAP (Extensible Authentication Protocol) is not an authentication system but a protocol for transporting authentication information. WPA-EAP is used to protect wireless networks in enterprises. In private networks, it is scarcely used. For this reason, WPA-EAP is sometimes referred to as WPA “Enterprise”.

WPA-EAP needs a Radius server to authenticate users. EAP offers three different methods for connecting and authenticating to the server:

- Transport Layer Security (EAP-TLS): TLS authentication relies on the mutual exchange of certificates for both server and client. First, the server presents its certificate to the client where it is evaluated. If the certificate is considered valid, the client in turn presents its certificate to the server. While TLS is secure, it requires a working certification management infrastructure in your network. This infrastructure is rarely found in private networks.
- Tunnelled Transport Layer Security (EAP-TTSL)
- Protected Extensible Authentication Protocol (EAP-PEAP): Both TTLS and PEAP are two-stage protocols. In the first stage, a secure connection is established and in the second the client authentication data is exchanged. They require far less certification management overhead than TLS, if any.

24.4 Encryption

There are various encryption methods to ensure that no unauthorized person can read the data packets that are exchanged in a wireless network or gain access to the network:

WEP (defined in IEEE 802.11)

This standard makes use of the RC4 encryption algorithm, originally with a key length of 40 bits, later also with 104 bits. Often, the length is declared as 64 bits or 128 bits, depending on whether the 24 bits of the initialization vector are included. However, this standard has some weaknesses. Attacks against the keys generated by this system may be successful. Nevertheless, it is better to use WEP than to not encrypt the network at all.

Some vendors have implemented the non-standard “Dynamic WEP”. It works exactly as WEP and shares the same weaknesses, except that the key is periodically changed by a key management service.

TKIP (defined in WPA/IEEE 802.11i)

This key management protocol defined in the WPA standard uses the same encryption algorithm as WEP, but eliminates its weakness. Because a new key is generated for every data packet, attacks against these keys are fruitless. 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.

24.5 Configuration with YaST

IMPORTANT: Security Risks in Wireless Networks

Unencrypted WLAN connections allow third parties to intercept all network data. Be sure to protect your network traffic by using one of the supported authentication and encryption methods.

Use the best possible encryption method your hardware allows. However, to use a certain encryption method, all devices in the network must support

this method, otherwise they cannot communicate with each other. For example, if your router supports both WEP and WPA but the driver for your WLAN card only supports WEP, WEP is the least common denominator you can use. But even a weak encryption with WEP is better than none at all. Refer to Section 24.4, “Encryption” (page 421) and Section 24.6.3, “Security” (page 431) for information.

To configure a wireless LAN with YaST, you need to define the following parameters:

IP Address

Use either a static IP address or let a DHCP server dynamically assign an IP address to the interface.

Operating Mode

Defines how to integrate your machine into a WLAN, depending on the network topology. For background information, refer to Section 24.2, “Operating Modes” (page 418).

Network Name (ESSID)

Unique string identifying a network.

Authentication and Encryption Details

Depending on the authentication and encryption method your network uses, you need to enter one or more keys and/or certificates.

Several input options are available for entering the respective keys: *Passphrase*, *ASCII* (only available for WEP authentication methods), and *Hexadecimal*.

24.5.1 Deactivating NetworkManager

A WLAN card is usually detected during installation. If your machine is a mobile computer, NetworkManager is usually activated by default. If instead you want to configure your WLAN card with YaST, you need to deactivate NetworkManager first:

- 1 Start YaST as user `root`.
- 2 In the YaST Control Center, select *Network Devices > Network Settings* to open the *Network Settings* dialog.

If your network is currently controlled by NetworkManager, you see a warning message that the network settings cannot be edited by YaST.

- 3 To enable editing with YaST, leave the message with *OK* and on the *Global Options* tab, activate *Traditional Method with ifup*.
- 4 For further configuration, proceed with Section 24.5.2, “Configuration for Access Points” (page 423) or Section 24.5.3, “Establishing an Ad-Hoc Network” (page 427).

Otherwise confirm your changes with *OK* to write the network configuration.

24.5.2 Configuration for Access Points

In this section, learn how to configure your WLAN card to connect to an (external) access point or how to use your WLAN card as access point if your WLAN card supports this. For configuration of networks without an access point, refer to Section 24.5.3, “Establishing an Ad-Hoc Network” (page 427).

Procedure 24.1: *Configuring Your WLAN Card for Using an Access Point*

- 1 Start YaST and open the *Network Settings* dialog.
- 2 Switch to the *Overview* tab where all network cards are listed that have been detected by the system. If you need more information about general network configuration, refer to Section 13.4, “Configuring a Network Connection with YaST” (page 232).
- 3 Choose your wireless card from the list and click *Edit* to open the *Network Card Setup* dialog.
- 4 On the *Address* tab, configure whether to use a dynamic or a static IP address for the machine. Usually *Dynamic Address* with *DHCP* is fine.
- 5 Click *Next* to proceed to the *Wireless Network Card Configuration* dialog.
- 6 To use your WLAN card to connect to an access point, set the *Operating Mode* to *Managed*.

If however you want to use your WLAN card as access point, set the *Operating Mode* to *Master*. Note that not all WLAN cards support this mode.

NOTE: Using WPA-PSK or WPA-EAP

If you want to use WPA-PSK or WPA-EAP authentication modes, the operating mode must be set to *Managed*.

- 7 To connect to a certain network, enter the *Network Name (ESSID)*. Alternatively, click *Scan Network* and select a network from the list of available wireless networks.

All stations in a wireless network need the same ESSID for communicating with each other. If no ESSID is specified, your WLAN card automatically associates with the access point that has the best signal strength.

NOTE: WPA Authentication Requires an ESSID

If you select *WPA* authentication, a network name (ESSID) must be set.

- 8 Select an *Authentication Mode* for your network. Which mode is suitable, depends on your WLAN card's driver and the ability of the other devices in the network.
- 9 If you have chosen to set the *Authentication Mode* to *No Encryption*, finish the configuration by clicking *Next*. Confirm the message about this potential security risk and leave the *Overview* tab (showing the newly configured WLAN card) with *OK*.

If you haven chosen any of the other authentication modes, proceed with Procedure 24.2, "Entering the Encryption Details" (page 425).

Figure 24.1: YaST: Configuring the Wireless Network Card

Wireless Network Card Configuration
Here, set the most important settings for wireless networking. [more](#)

Wireless Device Settings

Operating Mode:
Managed

Network Name (ESSID)
testNET Scan Network

Authentication Mode:
WEP - Shared Key

Key Input Type
☒ Passphrase ☐ ASCII ☐ Hexadecimal

Encryption Key:

Expert Settings WEP Keys

Help Abort Back Next

Procedure 24.2: *Entering the Encryption Details*

The following authentication methods require an encryption key: *WEP - Open*, *WEP - Shared Key*, and *WPA-PSK*.

For WEP, usually only key is needed—however, up to 4 different WEP keys can be defined for your station. One of them needs to be set as the default key and is used for encryption. The others are used for decryption. Per default, a key length of 128-bit is used, but you can also choose to set the length to 64-bit.

For higher security, WPA-EAP uses a RADIUS server to authenticate users. For authentication at the server, three different methods are available: TLS, TTLS and PEAP. The credentials and certificates you need for WPA-EAP depend on the authentication method used for the RADIUS server. Ask your system administrator to provide the needed information and credentials. YaST searches for any certificate under `/etc/cert`. Therefore, save the certificates given to you to this location and restrict access to these files to 0600 (owner read and write).

- 1 To enter the key for *WEP - Open* or *WEP - Shared Key*:

1a Set the *Key Input Type* either to *Passphrase*, *ASCII* or *Hexadecimal*.

1b Enter the respective *Encryption Key* (usually only one key is used):

If you have selected *Passphrase*, enter a word or a character string from which a key is generated according to the specified key length (per default, 128-bit) .

ASCII requests an input of 5 characters for a 64-bit key and 13 characters for a 128-bit key.

For *Hexadecimal*, enter 10 characters for a 64-bit key or 26 characters for a 128-bit key in hexadecimal notation.

1c To adjust the key length to a lower bit rate (which might be necessary for older hardware), click *WEP Keys* and set the *Key Length* to 64 bit. The *WEP Keys* dialog also shows the WEP keys that have been entered so far. Unless another key is explicitly set as default, YaST always uses the first key as default key.

1d To enter more keys for WEP or to modify one of the keys, select the respective entry and click *Edit*. Select the *Key Input Type* and enter the key.

1e Confirm your changes with *OK*.

2 To enter a key for *WPA-PSK*:

2a Select the input method *Passphrase* or *Hexadecimal*.

2b Enter the respective *Encryption Key*.

In the *Passphrase* mode, the input must be 8 to 63 characters. In the *Hexadecimal* mode, enter 64 characters.

3 If you have chosen *WPA-EAP* authentication, click *Next* to switch to the *WPA-EAP* dialog, where you enter the credentials and certificates you have been given by your network administrator.

3a Select the *EAP Mode* the RADIUS server uses for authentication. The details you need to enter in the following depend on the selected *EAP Mode*.

3b For TLS, provide *Identity*, *Client Certificate*, *Client Key*, and *Client Key Password*. To increase security, you can also configure a *Server Certificate* used to validate the server's authenticity.

TTLS and PEAP require *Identity* and *Password*, whereas *Server Certificate* and *Anonymous Identity* are optional.

3c To enter the advanced authentication dialog for your WPA-EAP setup, click *Details*.

3d Select the *Authentication Method* for the second stage of EAP-TTLS or EAP-PEAP communication (inner authentication). The choice of methods depends on the authentication method for the RADIUS server you selected in the previous dialog.

3e If the automatically-determined setting does not work for you, choose a specific *PEAP Version* to force the use of a certain PEAP implementation.

4 Confirm your changes with *OK*. The *Overview* tab shows the details of your newly configured WLAN card.

5 Click *OK* to finalize the configuration and to leave the dialog.

24.5.3 Establishing an Ad-Hoc Network

In some cases it is useful to connect two computers equipped with a WLAN card. To establish an ad-hoc network with YaST, do the following:

1 Start YaST and open the *Network Settings* dialog.

2 Switch to the *Overview* tab, choose your wireless card from the list and click *Edit* to open the *Network Card Setup* dialog.

3 Choose *Statically assigned IP Address* and enter the following data:

- *IP Address*: 192.168.1.1. Change this address on the second computer to 192.168.1.2, for example.
- *Subnet Mask*: /24

- *Hostname*: Choose any name you like.

4 Proceed with *Next*.

5 Set the *Operating Mode* to *Ad-hoc*.

6 Choose a *Network Name (ESSID)*. This can be any name, but it has to be used on every computer in the ad-hoc network.

7 Select an *Authentication Mode* for your network. Which mode is suitable, depends on your WLAN card's driver and the ability of the other devices in the network.

8 If you have chosen to set the *Authentication Mode* to *No Encryption*, finish the configuration by clicking *Next*. Confirm the message about this potential security risk and leave the *Overview* tab showing the newly configured WLAN card with *OK*.

If you haven't chosen any of the other authentication modes, proceed with Procedure 24.2, "Entering the Encryption Details" (page 425).

9 If you do not have `smpppd` installed, YaST asks you to do so.

10 Configure the other WLAN cards in the network accordingly, using the same *Network Name (ESSID)*, the same *Authentication Mode* but different IP addresses.

24.5.4 Setting Additional Configuration Parameters

Usually there is no need to change the preconfigured settings when configuring your WLAN card. However, if you need detailed configuration of your WLAN connection, YaST allows you to tweak the following settings:

Channel

The specification of a channel on which the WLAN station should work. This is only needed in *Ad-hoc* and *Master* modes. In *Managed* mode, the card automatically searches the available channels for access points.

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.

Power Management

When you are on the road, power saving technologies can help to maximize the operating time of your battery. Using power management may affect the connection quality and increase the network latency.

To access the advanced options:

- 1** Start YaST and open the *Network Settings* dialog.
- 2** Switch to the *Overview* tab, choose your wireless card from the list and click *Edit* to open the *Network Card Setup* dialog.
- 3** Click *Next* to proceed to the *Wireless Network Card Configuration* dialog.
- 4** Click *Expert Settings*.
- 5** In *Ad-hoc* mode, select one of the offered channels (11 to 14, depending on your country) 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*.
- 6** Select the *Bit Rate* to use.
- 7** Enter the MAC address of the *Access Point* you want to connect to.
- 8** Choose if to *Use Power Management* or not.
- 9** Confirm your changes with *OK* and click *Next* and *OK* to finish the configuration.

24.6 Tips and Tricks for Setting Up a WLAN

The following tools and tips can help to monitor and improve speed and stability as well as security aspects of your WLAN.

24.6.1 Utilities

The package `wireless-tools` contains utilities that allow to set wireless LAN specific parameters and get statistics. See http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html for more information.

`kismet` (package `kismet`) is a network diagnosis tool with which to listen to the WLAN packet traffic. In this way, you can also detect any intrusion attempts in your network. More information is available at <http://www.kismetwireless.net/> and in the manual page.

24.6.2 Stability and Speed

The performance and reliability of a wireless network mainly depend on whether the participating stations receive a clear 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 the Network-Manager applets provided by KDE or GNOME. If you have problems with the signal quality, try to set up the devices somewhere else or adjust the position of the antennas of your access points. Auxiliary antennas that substantially improve the reception are available for a number of PCMCIA WLAN cards. The rate specified by the manufacturer, such as 54 Mbit/s, is a nominal value that represents the theoretical maximum. In practice, the maximum data throughput is no more than half this value.

The `iwspy` command can displays WLAN statistics:

```
iwspy wlan0
wlan0      Statistics collected:
          00:AA:BB:CC:DD:EE : Quality:0   Signal level:0   Noise level:0
          Link/Cell/AP      : Quality:60/94 Signal level:-50 dBm   Noise
          level:-140 dBm (updated)
```


Typical/Reference : Quality:26/94 Signal level:-60 dBm Noise level:-90 dBm

24.6.3 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.

For private use, use WPA-PSK if available. Although Linux supports WPA on most hardware components, some drivers do not offer WPA support. It may also not be available on older access points and routers with WLAN functionality. For such devices, check if WPA can be implemented by means of a firmware update. 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.

Use strong passwords for your authentication method. For example, the Web page <https://www.grc.com/passwords.htm> generates random 64 character passwords.

24.7 Troubleshooting

If your WLAN card is not automatically detected, check whether it is supported by openSUSE. A list of supported WLAN network cards is available under [http://en.opensuse.org/HCL:Network_\(Wireless\)](http://en.opensuse.org/HCL:Network_(Wireless)). If your card is not supported, it may be possible to make it work using the Microsoft Windows drivers with Ndiswrapper. Please refer to <http://en.opensuse.org/SDB:Ndiswrapper> for detailed information.

If your WLAN card fails to respond, check the following prerequisites:

1. Do you know the device name of the WLAN card? Usually it is `wlan0`. Check with the tool `ifconfig`.
2. Have you checked your needed firmware? Refer to `/usr/share/doc/packages/wireless-tools/README.firmware` for more information.
3. Is the ESSID of your router broadcasted and visible (not hidden)?

24.7.1 Check the Network Status

The command `iwconfig` can give you important information about your wireless connection. For example, the following line displays the ESSID, the wireless mode, frequency, if you signal is encrypted, the link quality, and much more:

```
iwconfig wlan0
wlan0 IEEE 802.11abg ESSID:"guest"
      Mode:Managed  Frequency:5.22GHz  Access Point: 00:11:22:33:44:55
      Bit Rate:54 Mb/s   Tx-Power=13 dBm
      Retry min limit:7   RTS thr:off   Fragment thr:off
      Encryption key:off
      Power Management:off
      Link Quality:62/92   Signal level:-48 dBm  Noise level:-127 dBm
      Rx invalid nwid:0   Rx invalid crypt:0   Rx invalid frag:0
      Tx excessive retries:10   Invalid misc:0   Missed beacon:0
```

You can also get the previous information with the `iwlist` command. For example, the following line displays the current bit rate:

```
iwlist wlan0 rate
wlan0      unknown bit-rate information.
          Current Bit Rate=54 Mb/s
```

If you want an overview how many access points are available, it can also be done with the `iwlist` command. It gives you a list of “cells” which looks like this:

```
iwlist wlan0 scanning
wlan0  Scan completed:
      Cell 01 - Address: 00:11:22:33:44:55
          Channel:40
          Frequency:5.2 GHz (Channel 40)
          Quality=67/70   Signal level=-43 dBm
          Encryption key: off
          ESSID:"Guest"
          Bit Rates: 6 Mb/s; 9 Mb/s; 12 Mb/s; 18 Mb/s;
                   24 Mb/s; 36 Mb/s; 48 Mb/s
          Mode: Master
          Extra:tsf=0000111122223333
          Extra: Last beacon: 179ms ago
          IE: Unknown: ...
```

24.7.2 Multiple Network Devices

Modern laptops usually have a network card and a WLAN card. If you configured both devices with DHCP (automatic address assignment), you may encounter problems with the name resolution and the default gateway. This is evident from the

fact that you can ping the router but cannot surf the Internet. The Support Database features an article on this subject at http://old-en.opensuse.org/SDB:Name_Resolution_Does_Not_Work_with_Several_Concurrent_DHCP_Clients.

24.7.3 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`.

24.8 For More Information

More information can be found on the following pages:

http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html

The Internet pages of Jean Tourrilhes, who developed the *Wireless Tools* for Linux, present a wealth of useful information about wireless networks.

tuxmobil.org

Useful hands-on information about mobile computers under Linux.

<http://www.linux-on-laptops.com>

More information about Linux on laptops.

[http://en.opensuse.org/HCL:Network_\(Wireless\)](http://en.opensuse.org/HCL:Network_(Wireless))

Lists supported WLAN network cards.

<http://en.opensuse.org/SDB:Ndiswrapper>

Offers a work-around for running unsupported WLAN cards with the Microsoft Windows using Ndiswrapper.

Using NetworkManager

NetworkManager is the ideal solution for laptops and other portable computers. It supports state-of-the-art encryption types and standards for network connections, including connections to 802.1X protected networks. 802.1X is the “IEEE Standard for Local and Metropolitan Area Networks—Port-Based Network Access Control”. With NetworkManager, you need not worry about configuring network interfaces and switching between wired or wireless networks when you are moving. NetworkManager can automatically connect to known wireless networks or manage several network connections in parallel—the fastest connection is then used as default. Furthermore, you can manually switch between available networks and manage your network connection using an icon in the system tray or in the top bar, respectively.

Instead of only one connection being active, multiple connections may be active at once. This enables you to unplug your laptop from an Ethernet and remain connected via a wireless connection.

25.1 Use Cases for NetworkManager

NetworkManager provides a sophisticated and intuitive user interface, which enables users to easily switch their network environment. However, NetworkManager is not a suitable solution in the following cases:

- Your computer provides network services for other computers in your network, for example, it is a DHCP or DNS server.

- Your computer is a Xen server or your system is a virtual system inside Xen.

25.2 Enabling or Disabling NetworkManager

On laptop computers, NetworkManager is enabled by default. However, it can be at any time enabled or disabled in the YaST Network Settings module.

- 1 Run YaST and go to *Network Devices > Network Settings*.
- 2 The *Network Settings* dialog opens. Go to the *Global Options* tab.
- 3 To configure and manage your network connections with NetworkManager:
 - 3a In the *Network Setup Method* field, select *User Controlled with NetworkManager*.
 - 3b Click *OK* and close YaST.
 - 3c Configure your network connections with NetworkManager as described in Section 25.3, “Configuring Network Connections” (page 437).
- 4 To deactivate NetworkManager and control network the traditional way:
 - 4a In the *Network Setup Method* field, choose *Traditional Method with ifup*.
 - 4b Click *OK*.
 - 4c Set up your network card with YaST using automatic configuration via DHCP or a static IP address. Alternatively, configure your modem with YaST:
 - For dial-up connections, use *Network Devices > Modem*.
 - To configure an internal or USB ISDN modem, select *Network Devices > ISDN*.
 - To configure an internal or USB DSL modem, select *Network Devices > DSL*.

Find a detailed description of the network configuration with YaST in Section 13.4, “Configuring a Network Connection with YaST” (page 232) and Chapter 24, *Wireless LAN* (page 417).

25.3 Configuring Network Connections

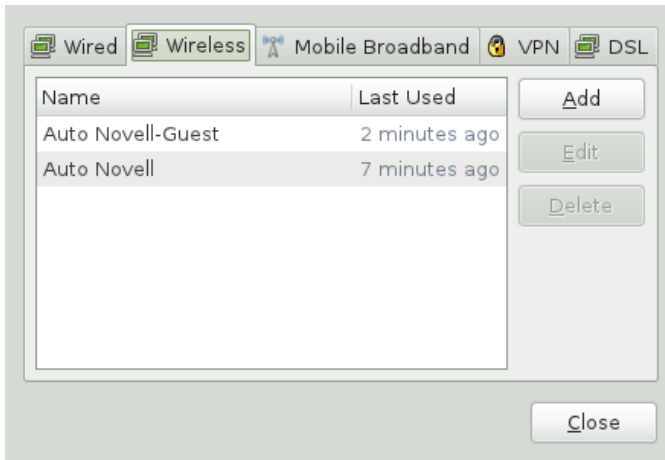
After having enabled NetworkManager in YaST, configure your network connections with the NetworkManager front-ends available in KDE and GNOME. The network configuration dialogs for both front-ends are very similar. They show tabs for all types of network connections, such as wired, wireless, mobile broadband, DSL, and VPN connections. On each tab, you can add, edit or delete connections of that type. If you hover the mouse pointer over the input fields and options, comprehensive tooltips are displayed. In the KDE configuration dialog, the appropriate tabs are only active if the connection type is available on your system (depending on hardware and software).

NOTE: Bluetooth Connections

Currently, Bluetooth connections cannot be configured with NetworkManager.

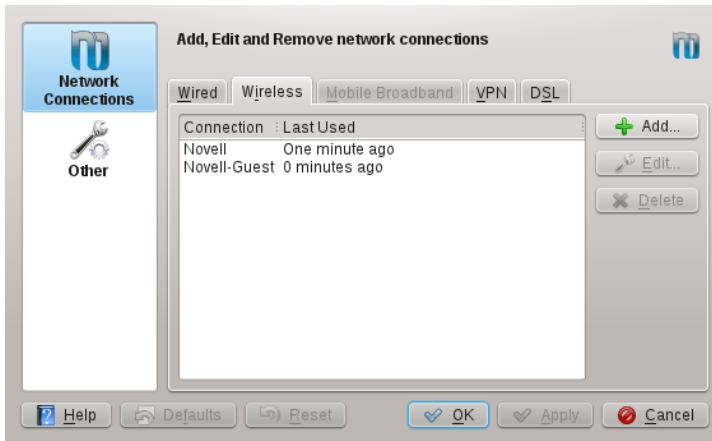
To open the network configuration dialog in GNOME, press **Alt + F2** and enter `nm-connection-editor`.

Figure 25.1: *GNOME Network Connections Dialog*



If you are using KDE, open the main menu and click *System settings*. In the *Network and Connectivity* section select *Network Settings > Network Connections* to open the network configuration dialog.

Figure 25.2: *KDE Network Configuration Dialog*



Alternatively, you can also start the configuration dialogs from the NetworkManager front-end in the system tray or the top bar, respectively. In KDE, left-click the icon and select *Manage Connections*. In GNOME, click the icon and select *Network Settings > Options*.

NOTE: Availability of Options

Depending on your system setup, you may not be allowed to configure connections. In a secured environment, some options might be locked or require `root` permission. Ask your system administrator for details.

Procedure 25.1: *Adding or Editing Connections*

When configuring network connections with NetworkManager, you can also define `system connections` that can be shared by all users. In contrast to `user connections`, system connections are made available right after NetworkManager is started—before any users log in. For more details about both types of connections, refer to Section 25.7.1, “User and System Connections” (page 448).

NOTE: Hidden Networks

To connect to a “hidden” network (a network that does not broadcast its service) you have to know the Service Set Identifier (SSID) or Extended Service Set Identifier (ESSID) of the network. Hidden networks cannot be detected automatically.

- 1** In the network configuration dialog, click the tab for the connection type you want to use.
- 2** Click *Add* to create a new connection or select an existing connection and click *Edit*.
- 3** Enter a *Connection Name* and your connection details.
- 4** You can tie the connection to a certain device, if more than one physical device per connection type is available (for example, your machine is equipped with two ethernet cards or two wireless cards).

If you are using KDE, do so by using the *Restrict to Interface* option. If you are using GNOME, enter the *MAC address* of the device you want to tie the connection to and confirm your settings.

- 5** For NetworkManager to automatically use a certain connection, activate the following option for this connection: *Connect Automatically* (KDE) or *Stay connected when possible* (GNOME).

- 6 To turn a connection into a system connection activate *System Connection* (KDE) or *Available to all users* (GNOME). To create and edit system connections, `root` permission is required.

After having confirmed your changes, the newly configured network connection appears in the list of available networks you get by left-clicking the NetworkManager icon.

Figure 25.3: KDE NetworkManager—Configured and Available Connections

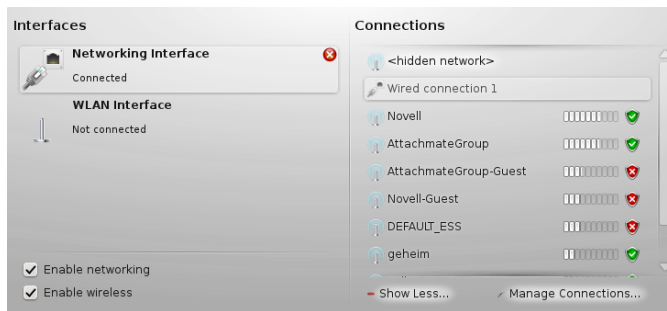
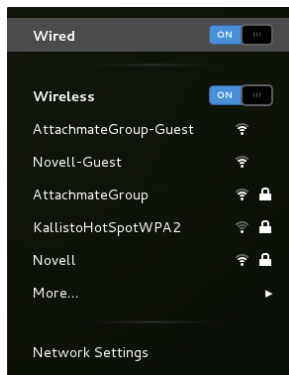


Figure 25.4: GNOME NetworkManager—Configured and Available Connections



25.4 Using the KDE NetworkManager Front-End

The KDE front-end for NetworkManager is the NetworkManager plasmoid. If the network has been set up for NetworkManager control, the plasmoid usually starts automatically with the desktop environment and is shown as an icon in the system tray.

If your system tray does not show any network connection icon, the plasmoid is probably not started. Click the *Panel Tool Box* and choose *Add Widgets*. Double-click the *Network Management* entry and click the *Panel Tool Box* again.

The NetworkManager front-end only shows wireless networks for which you have configured a connection. It hides connections when you are out of range of a wireless network, or when the network cable is disconnected, thus always giving you a clear view of which connections may be used.

25.4.1 Managing Wired Network Connections

If your computer is connected to an existing network with a network cable, use the NetworkManager front-end to manage the network connection(s).

- 1 Left-click the NetworkManager icon to show a list with available *Interfaces* on the left-hand side and *Connections* on the right-hand side. The connections currently being used are shown in bold in the *Connections* list.
- 2 For detailed information and statistics for an interface, click the respective *Networking Interface* entry on the left-hand side of the plasmoid. Switch back to the interface overview by clicking the blue arrow icon.
- 3 To disconnect an active connection, click the red icon next to the respective *Networking Interface* entry.
- 4 If you want to use a different configuration with the wired network, click *Manage Connections* and add another wired connection as described in Procedure 25.1, “Adding or Editing Connections” (page 439). Click the NetworkManager icon and select the newly configured connection to activate it.

- 5 To switch off all network connections, both wired and wireless, click the Network-Manager icon and uncheck *Enable Networking*.

25.4.2 Managing Wireless Network Connections

By default the NetworkManager front-end only lists connections that are already configured. The signal strength of each network is indicated by a series of bars, each one representing 10%. Encrypted wireless networks are marked with a green (WPA) or yellow (WEP) shield, while open networks are marked with a red shield.

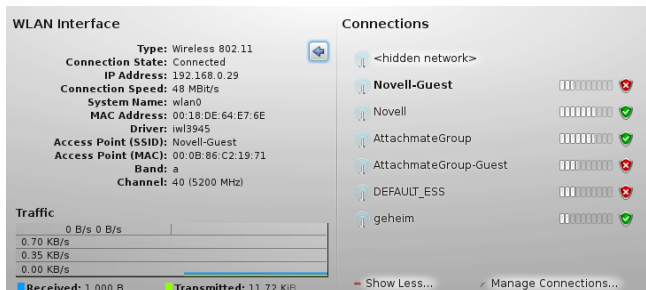
Procedure 25.2: *Connecting to a Wireless Network*

- 1 Click *Show More* to display all visible wireless networks available.
- 2 Left-click on the network of your choice to open the network connection dialog. To connect to a network that does not broadcast its service set identifier (SSID or ESSID), click *hidden network*. Fill in the connection details and press Enter.
- 3 Based on your security settings for NetworkManager (see Section 25.7.2, “Storing Passwords and Credentials” (page 449) for details), you may have to provide your KWallet password.
- 4 NetworkManager automatically connects to the configured network.

Procedure 25.3: *Managing Active Wireless Connections*

- 1 Click a *WLAN Interface* listed in the right half of the plasmoid window to get detailed information and statistics for that interface.

Figure 25.5: *KDE NetworkManager— Connection Details and Statistics*



Switch back to the interface overview by clicking on the blue arrow icon.

- 2 To disconnect an active connection, click the red icon for the *WLAN Interface*.
- 3 To completely disable wireless networking, uncheck *Enable Wireless*. This can be useful if you are on a plane or in any other environment where wireless networking is not allowed.

A wireless network that has been chosen explicitly will remain connected as long as possible. If a network cable is plugged in during that time, any connections that have been set to *Connect Automatically* will be connected, while the wireless connection remains up.

25.4.3 Configuring Your Wireless Card as an Access Point

If your wireless card supports access point mode, you can use NetworkManager for configuration.

NOTE: Availability of Options

Depending on your system setup, you may not be allowed to configure connections. In a secured environment, some options might be locked or require `root` permission. Ask your system administrator for details.

- 1 Start the dialog for configuring network connections as described in Section 25.3, “Configuring Network Connections” (page 437).
- 2 Click *Add > Shared*.
- 3 On the *Wireless* tab provide a *Connection name* for the shared wireless network and a *SSID*.
- 4 Set the encryption on the *Wireless Security* tab.

IMPORTANT: Unprotected Wireless Networks Are a Security Risk

If you set *Security* to *None*, everybody can connect to your network, reuse your connectivity and intercept your network connection. To restrict ac-

cess to your access point and to secure your connection, use encryption. You can choose between various WEP and WPA-based encryptions. If you are not sure which technology is best for you, read Section 24.3, “Authentication” (page 419).

- 5 Confirm your configuration with *OK*.

25.5 Using GNOME NetworkManager

In GNOME, NetworkManager can be controlled with the GNOME NetworkManager icon. If the network is set up for NetworkManager control, the icon is automatically shown in the top bar.

If your top bar does not show any network connection icon, GNOME NetworkManager is probably not started. Press **Alt + F2** and enter `nm-applet` to start it manually.

25.5.1 Managing Wired Network Connections

If your computer is connected to an existing network with a network cable, use the NetworkManager icon to choose the network connection.

- 1 Click the icon to show a menu with available networks. The currently used connection is listed on top of the menu, followed by other active connections.
- 2 Click *Network Settings* to get detailed information on all active interfaces.
- 3 To disconnect an active connection, click the *OFF* button next to *Wired*.
- 4 If you want to use a different configuration with the wired network, open the network configuration dialog and add another wired connection as described in Procedure 25.1, “Adding or Editing Connections” (page 439). Click the NetworkManager icon and select the newly configured connection to activate it.
- 5 To switch off all network connections, both wired and wireless, right-click the icon and uncheck *Enable Networking*.

25.5.2 Managing Wireless Network Connections

GNOME NetworkManager lists a number of available visible wireless networks. To extend the list, click *More Networks*. The signal strength of each network is also shown in the menu. Encrypted wireless networks are marked with a lock icon.

Procedure 25.4: *Connecting to a Wireless Network*

- 1** To connect to a wireless network, click the NetworkManager icon and choose an entry from the list of available wireless networks.
- 2** If the network is encrypted, a dialog opens. It shows the type of encryption the network uses (*Wireless Security*) and holds a number of input fields according to the respective encryption and authentication settings. Enter the appropriate credentials.
- 3** To connect to a network that does not broadcast its service set identifier (SSID or ESSID), and therefore cannot be detected automatically, click the NetworkManager icon and choose *Network Settings > Wireless > Other*.
- 4** In the dialog that opens, enter the connection details and click *Connect*.
- 5** To disable wireless networking, click the icon and click the *OFF* button next to *Wireless*. This can be useful if you are on a plane or in any other environment where wireless networking is not allowed.

A wireless network that has been chosen explicitly will remain connected as long as possible. If a network cable is plugged in during that time, any connections that have been set to *Connect Automatically* will be connected, while the wireless connection remains up.

25.5.3 Configuring Your Wireless Card as an Access Point

If your wireless card supports access point mode, you can use NetworkManager for configuration.

NOTE: Availability of Options

Depending on your system set-up, you may not be allowed to configure connections. In a secured environment, some options might be locked or require `root` permission. Ask your system administrator for details.

- 1 Click the NetworkManager icon and select *Network Settings Wireless*.
- 2 Click *Use as Hotspot*, confirm the pop-up and enter the `root` password to continue.

A *Network Name* (SSID) and a *Security Key* are automatically generated and shown in the *Network* dialog. The network name will be based on the hostname of your computer. Other devices will need this information to connect to the hotspot that you have created.

IMPORTANT: Unprotected Wireless Networks Are a Security Risk

To restrict access to your access point and to secure your connection, use encryption. Depending on the abilities of your network card, you can choose between various WEP and WPA-based encryptions. If you are not sure which technology is best for you, read Section 24.3, “Authentication” (page 419).

- 3 To change the SSID, encryption options (WEP, WPA, etc.) or the security key of the hotspot:
 - 3a Click *Options* next to the *Stop Hotspot* button.
 - 3b Enter the `root` password to continue.
 - 3c Change the SSID on the *Wireless* tab or the encryption details on the *Wireless Security* tab.

NOTE: WEP 40/128-bit Key (Hex or ASCII)

When using `WEP 40/128-bit Key` as encryption method, the *Key* length is restricted to either 5, 10, or 13 characters. Otherwise the *Save* button is inactive.

- 3d** Confirm your changes. After a short delay, the *Network* dialog will show your changes.
- 4** To stop the hotspot and disconnect any users, click *Stop Hotspot* and confirm your choice in the pop-up dialog.

25.6 NetworkManager and VPN

NetworkManager supports several Virtual Private Network (VPN) technologies. For each technology, openSUSE comes with a base package providing the generic support for NetworkManager. In addition to that, you also need to install the respective desktop-specific package for your front-end.

NovellVPN

To use this VPN technology, install

- `NetworkManager-novellvpn` and
- `NetworkManager-novellvpn-kde4` or `NetworkManager-novellvpn-gnome`.

NovellVPN support for KDE is not available yet, but is currently being worked on.

OpenVPN

To use this VPN technology, install

- `NetworkManager-openvpn` and
- `NetworkManager-openvpn-kde4` or `NetworkManager-openvpn-gnome`.

vpnc (Cisco)

To use this VPN technology, install

- `NetworkManager-vpnc` and
- `NetworkManager-vpnc-kde4` or `NetworkManager-vpnc-gnome`.

PPTP (Point-to-Point Tunneling Protocol)

To use this VPN technology, install

- `NetworkManager-pptp` and
- `NetworkManager-pptp-kde4` or `NetworkManager-pptp-gnome`.

After you have installed the packages, configure your VPN connection as described in Section 25.3, “Configuring Network Connections” (page 437).

25.7 NetworkManager and Security

NetworkManager distinguishes two types of wireless connections, trusted and untrusted. A trusted connection is any network that you explicitly selected in the past. All others are untrusted. Trusted connections are identified by the name and MAC address of the access point. Using the MAC address ensures that you cannot use a different access point with the name of your trusted connection.

NetworkManager periodically scans for available wireless networks. If multiple trusted networks are found, the most recently used is automatically selected. NetworkManager waits for your selection in case that all networks are untrusted.

If the encryption setting changes but the name and MAC address remain the same, NetworkManager attempts to connect, but first you are asked to confirm the new encryption settings and provide any updates, such as a new key.

If you switch from using a wireless connection to offline mode, NetworkManager blanks the SSID or ESSID. This ensures that the card is disconnected.

25.7.1 User and System Connections

NetworkManager knows two types of connections: `user` and `system` connections. User connections are connections that become available to NetworkManager when the first user logs in. Any required credentials are asked from the user and when the user logs out, the connections are disconnected and removed from NetworkManager. Connections that are defined as system connection can be shared by all users and are made available right after NetworkManager is started—before any users log in. In case of system connections, all credentials must be provided at the time the con-

nection is created. Such system connections can be used to automatically connect to networks that require authorization. For information how to configure user or system connections with NetworkManager, refer to Section 25.3, “Configuring Network Connections” (page 437).

25.7.2 Storing Passwords and Credentials

If you do not want to re-enter your credentials each time you want to connect to an encrypted network, you can use the desktop-specific tools GNOME Keyring Manager or KWalletManager to store your credentials encrypted on the disk, secured by a master password.

In KDE, you can configure if and how to store your credentials. To do so, left-click the NetworkManager icon and select *Manage Connections*. Click *Other > Connection Secrets* and select one of the following options:

Do Not Store (Always Prompt)

This is useful if you are working in an environment where storing credentials is considered a security risk.

In File (Unencrypted)

If you choose this option, your passwords are stored unencrypted in the respective connection file that is created for each connection.

WARNING: Security Risk

Storing your network credentials unencrypted is a security risk. Everybody who has access to your computer can reuse your connectivity and intercept your network connection.

In Secure Storage (Encrypted)

If you choose this option, your credentials are stored in KWalletManager.

25.8 Frequently Asked Questions

In the following, find some frequently asked questions about configuring special network options with NetworkManager.

How to tie a connection to a specific device?

By default, connections in NetworkManager are device type-specific: they apply to all physical devices with the same type. If more than one physical device per connection type is available (for example, your machine is equipped with two ethernet cards), you can tie a connection to a certain device.

To do so in GNOME, first look up the MAC address of your device by using the output of the command line tool `ifconfig`. Then start the dialog for configuring network connections and choose the connection you want to modify. On the *Wired* or *Wireless* tab, enter the *MAC Address* of the device and confirm your changes.

If you are using KDE, start the dialog for configuring network connections and choose the connection you want to modify. On the *Ethernet* or *Wireless* tab, use the *Restrict to Interface* option to select the network interface to which to tie the connection.

How to specify a certain access point in case multiple access points with the same ESSID are detected?

When multiple access points with different wireless bands (a/b/g/n) are available, the access point with the strongest signal is automatically chosen by default. To override this, use the *BSSID* field when configuring wireless connections.

The Basic Service Set Identifier (BSSID) uniquely identifies each Basic Service Set. In an infrastructure Basic Service Set, the BSSID is the MAC address of the wireless access point. In an independent (ad-hoc) Basic Service Set, the BSSID is a locally administered MAC address generated from a 46-bit random number.

Start the dialog for configuring network connections as described in Section 25.3, “Configuring Network Connections” (page 437). Choose the wireless connection you want to modify and click *Edit*. On the *Wireless* tab, enter the BSSID.

How to share network connections to other computers?

The primary device (the device which is connected to the Internet) does not need any special configuration. However, you need to configure the device that is connected to the local hub or machine as follows:

1. Start the dialog for configuring network connections as described in Section 25.3, “Configuring Network Connections” (page 437). Choose the connection you want to modify and click *Edit*. If you are using GNOME, switch to the *IPv4 Settings* tab and from the *Method* drop-down list, choose *Shared to*

other computers. If you are using KDE, switch to the *IPv4 Address* or *IPv4 Address* tab and from the *Method* drop-down list, choose *Shared*. That will enable IP traffic forwarding and run a DHCP server on the device. Confirm your changes in NetworkManager.

2. As the DHCP server uses port 67, make sure that it is not blocked by the firewall: On the machine sharing the connections, start YaST and select *Security and Users > Firewall*. Switch to the *Allowed Services* category. If *DCHPv4 Server* is not already shown as *Allowed Service*, select *DCHPv4 Server* from *Services to Allow* and click *Add*. Confirm your changes in YaST.

How to provide static DNS information with automatic (DHCP, PPP, VPN) addresses?

In case a DHCP server provides invalid DNS information (and/or routes), you can override it. Start the dialog for configuring network connections as described in Section 25.3, “Configuring Network Connections” (page 437). Choose the connection you want to modify and click *Edit*. If you are using GNOME, switch to the *IPv4 Settings* tab, and from the *Method* drop-down list, choose *Automatic (DHCP) addresses only*. If you are using KDE, switch to the *IPv4 Address* or *IPv4 Address* tab, and from the *Method* drop-down list, choose *Automatic (DHCP) addresses only*. Enter the DNS information in the *DNS Servers* and *Search Domains* fields. From the drop-down list at the top of the tab, select *Routes* if you want to *Ignore automatically obtained routes*. Confirm your changes.

How to make NetworkManager connect to password protected networks before a user logs in?

Define a `system connection` that can be used for such purposes. For more information, refer to Section 25.7, “NetworkManager and Security” (page 448).

25.9 Troubleshooting

Connection problems can occur. Some common problems related to NetworkManager include the front-end not starting or a missing VPN option. Methods for resolving and preventing these problems depend on the tool used.

NetworkManager Front-End Does Not Start

The GNOME and KDE NetworkManager front-ends start automatically if the network is set up for NetworkManager control. If the front-end does not start,

check if NetworkManager is enabled in YaST as described in Section 25.2, “Enabling or Disabling NetworkManager” (page 436). Then make sure that the appropriate package for your desktop environment is also installed. If you are using KDE 4, the package is `plasmoid-networkmanagement`. For GNOME users the package is `NetworkManager-gnome`.

If the desktop front-end is installed but is not running for some reason, start it manually with `Alt + F2` and `nm-applet` (GNOME) or by clicking the *Panel Tool Box* and choosing *Add Widgets*. (KDE) Double-click the *Network Management* entry and click the *Panel Tool Box* again.

NetworkManager Front-End Does Not Include the VPN Option

Support for NetworkManager, front-ends, and VPN for NetworkManager is distributed in separate packages. If your NetworkManager front-end does not include the VPN option, check if the packages with NetworkManager support for your VPN technology are installed. For more information, see Section 25.6, “NetworkManager and VPN” (page 447).

No Network Connection Available

If you have configured your network connection correctly and all other components for the network connection (router, etc.) are also up and running, it sometimes helps to restart the network interfaces on your computer. To do so, log in to a command line as `root` and run `rcnetwork restart`.

25.10 For More Information

More information about NetworkManager can be found on the following Web sites and directories:

NetworkManager Project Page

<http://projects.gnome.org/NetworkManager/>

KDE NetworkManager Front-End

<http://userbase.kde.org/NetworkManagement>

Package Documentation

Also check out the information in the following directories for the latest information about NetworkManager and the GNOME and KDE NetworkManager front-ends:

- `/usr/share/doc/packages/NetworkManager/`,
- `/usr/share/doc/packages/NetworkManager-gnome/`.

Using Tablet PCs

openSUSE® comes with support for Tablet PCs. In the following, learn how to install and configure your Tablet PC and discover some useful Linux* applications which accept input from digital pens.

The following Tablet PCs are supported:

- Tablet PCs with serial and USB Wacom tablet (pen based), touch-screen or multi-touch devices.
- Tablet PCs with FinePoint devices, such as Gateway C210X/M280E/CX2724 or HP Compaq TC1000.
- Tablet PCs with touch screen devices, such as Asus R2H, Clevo TN120R, Fujitsu Siemens Computers P-Series, LG C1, Samsung Q1/Q1-Ultra.

After you have installed the Tablet PC packages and configured your digitizer correctly, input with the pen (also called a stylus) can be used for the following actions and applications:

- Logging in to KDM or GDM
- Unlocking your screen on the KDE and GNOME desktops
- Actions that can also be triggered by other pointing devices (such as mouse or touch pad), for example, moving the cursor on the screen, starting applications, closing, resizing and moving windows, shifting window focus and dragging and dropping objects

- Using gesture recognition in applications of the X Window System
- Drawing with GIMP
- Taking notes or sketching with applications like Jarnal or Xournal or editing larger amounts of text with Dasher

26.1 Installing Tablet PC Packages

The packages needed for Tablet PCs are included in the `TabletPC` installation pattern—if this is selected during installation, the following packages should already be installed on your system:

- `cellwriter`: a character-based handwriting input panel
- `jarnal`: a Java-based note taking application
- `xournal`: an application for note taking and sketching
- `xstroke`: a gesture recognition program for the X Window System
- `xvkbd`: a virtual keyboard for the X Window System
- `x11-input-fujitsu`: the X input module for Fujitsu P-Series tablets
- `x11-input-evtouch`: the X input module for some Tablet PCs with touch screens
- `xorg-x11-driver-input`: the X input module for input devices, including the module for Wacom devices.

If these packages are not installed, manually install the packages you need from command line or select the `TabletPC` pattern for installation in YaST.

26.2 Configuring Your Tablet Device

During installation, your tablet or touch device is configured by default. If you have trouble with the configuration of your Wacom device, you use `xsetwacom` on the command line to change the settings.

26.3 Using the Virtual Keyboard

To log in to the KDE or GNOME desktop or to unlock the screen, you can either enter your username and password as usual or via the virtual keyboard (xvkbd) displayed below the login field. To configure the keyboard or to access the integrated help, click the *xvkbd* field at the left lower corner and open the xvkbd main menu.

If your input is not visible (or is not transferred to the window where you need it), redirect the focus by clicking the *Focus* key in xvkbd and then clicking into the window that should get the keyboard events.

Figure 26.1: *xvkbd Virtual Keyboard*



If you want to use xvkbd after login, start it from the main menu or with *xvkbd* from a shell.

26.4 Rotating Your Display

Use *KRandRTray* (KDE) or *gnome-display-properties* (GNOME) to rotate or resize your display manually on the fly. Both *KRandRTray* and *gnome-display-properties* are applets for the RANDR extension of the X server.

Start *KRandRTray* or *gnome-display-properties* from the main menu, or enter *kran-drtray* or *gnome-display-properties* to start the applet from a shell. After you have started the applet, the applet icon is usually added to your system tray. If the *gnome-display-properties* icon does not automatically appear in the system tray, make sure *Show Displays in Panel* is activated in the *Monitor Resolution Settings* dialog.

To rotate your display with *KRandRTray*, right-click the icon and select *Configure Display*. Select the desired orientation from the configuration dialog.

To rotate your display with `gnome-display-properties`, right-click the icon and select the desired orientation. Your display is immediately tilted to the new direction. The orientation of the graphics tablet changes also, so it can still interpret the movement of the pen correctly.

If you have problems changing the orientation of your desktop, refer to Section 26.7, “Troubleshooting” (page 462) for more information.

26.5 Using Gesture Recognition

openSUSE includes both CellWriter and `xstroke` for gesture recognition. Both applications accept gestures executed with the pen or other pointing devices as input for applications on the X Window System.

26.5.1 Using CellWriter

With CellWriter, you can write characters into a grid of cells—the writing is instantly recognized on a character basis. After you have finished writing, you can send the input to the currently focused application. Before you can use CellWriter for gesture recognition, the application needs to be trained to recognize your handwriting: You need to train each character of a certain map of keys (untrained characters are not activated and thus cannot be used).

Procedure 26.1: *Training CellWriter*

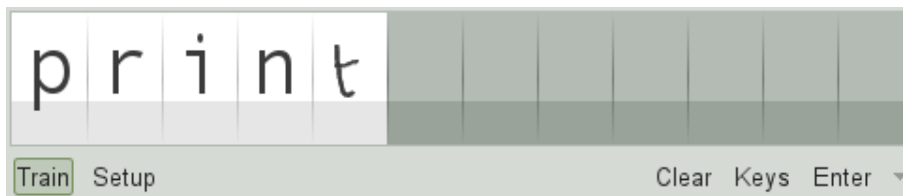
- 1 Start CellWriter from the main menu or with `cellwriter` from the command line. On the first start, CellWriter automatically starts in the training mode. In training mode it shows a set of characters of the currently chosen key map.
- 2 Enter the gesture you would like to use for a character into the respective character's cell. With the first input, the background changes its color to white, whereas the character itself is shown in light gray. Repeat the gesture multiple times until the character changes its color to black. Untrained characters are shown on a light gray or brown background (depending on the desktop's color scheme).
- 3 Repeat this step until you have trained CellWriter for all characters you need.
- 4 If you want to train CellWriter for another language, click the *Setup* button and select a language from the *Languages* tab. *Close* the configuration dialog. Click the

Train button and select the key map from the drop-down box at the bottom right corner of the *CellWriter* window. Now repeat your training for the new map of keys.

- 5 After having finished the training for the map of keys, click the *Train* button to switch to the normal mode.

In the normal mode, the *CellWriter* window shows a couple of empty cells in which to enter the gestures. The characters are not sent to another application until you click the *Enter* button, so you can correct or delete characters before you use them as input. Characters that have been recognized with a low degree of confidence will appear highlighted. To correct your input, use the context menu that appears on right-clicking a cell. To delete a character, either use your pen's eraser, or middle-click with the mouse to clear the cell. After finishing your input in *CellWriter*, define which application should receive the input by clicking into the application's window. Then send the input to the application by clicking *Enter*.

Figure 26.2: *Gesture Recognition with CellWriter*



If you click the *Keys* button in *CellWriter*, you get a virtual keyboard that can be used instead of the handwriting recognition.

To hide *CellWriter*, close the *CellWriter* window. The application now appears as icon in your system tray. To show the input window again, click the icon in the system tray.

26.5.2 Using Xstroke

With *xstroke*, you can use gestures with your pen or other pointing devices as input for applications on the X Window System. The *xstroke* alphabet is a unistroke alphabet that resembles the Graffiti* alphabet. When activated, *xstroke* sends the input to the currently focused window.

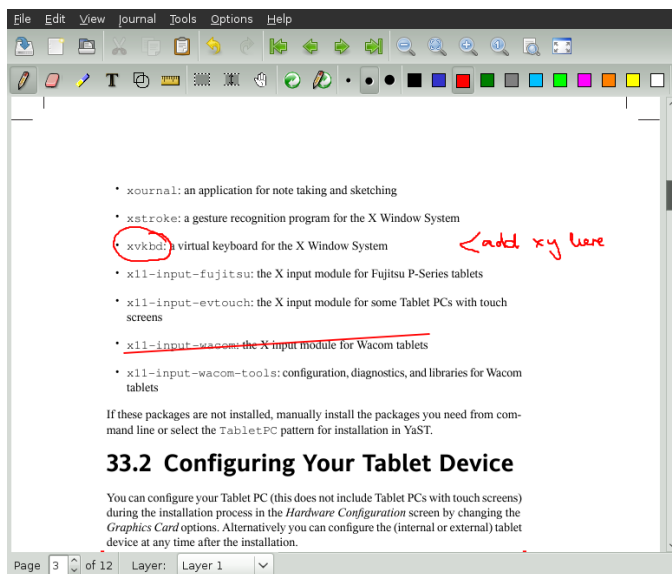
- 1 Start `xstroke` from the main menu or with `xstroke` from a shell. This adds a pencil icon to your system tray.
- 2 Start the application for which you want to create text input with the pen (for example, a terminal window, a text editor or an LibreOffice Writer).
- 3 To activate the gesture recognition mode, click the pencil icon once.
- 4 Perform some gestures on the graphics tablet with the pen or another pointing device. `xstroke` captures the gestures and transfers them to text that appears in the application window that has the focus.
- 5 To switch focus to a different window, click the desired window with the pen and hold for a moment (or use the keyboard shortcut defined in your desktop's control center).
- 6 To deactivate the gesture recognition mode, click the pencil icon again.

26.6 Taking Notes and Sketching with the Pen

To create drawings with the pen, you can use a professional graphics editor like GIMP or try one of the note-taking applications, Xournal or Jarnal. With both Xournal and Jarnal, you can take notes, create drawings or comment PDF files with the pen. As a Java-based application available for several platforms, Jarnal also offers basic collaboration features. For more information, refer to <http://www.dklevine.com/general/software/tc1000/jarnal-net.htm>. When saving your contents, Jarnal stores the data in an archive format (*.jaj) that also contains a file in SVG format.

Start Jarnal or Xournal from the main menu or by entering `jarnal` or `xournal` in a shell. To comment a PDF file in Xournal, for example, select *File > Annotate PDF* and open the PDF file from your file system. Use the pen or another pointing device to annotate the PDF and save your changes with *File > Print to PDF*.

Figure 26.3: *Annotating a PDF with Xournal*



Dasher is another useful application. It was designed for situations where keyboard input is impractical or unavailable. With a bit of training, you can rapidly enter larger amounts of text using only the pen (or other input devices—it can even be driven with an eye tracker).

Start Dasher from the main menu or with `dasher` from a shell. Move your pen in one direction and the application starts to zoom into the letters on the right side. From the letters passing the cross hairs in the middle, the text is created or predicted and is printed to the upper part of the window. To stop or start writing, click the display once with the pen. Modify the zooming speed at the bottom of the window.


```
xrandr -o normal && xsetwacom --set "Serial Wacom Tablet" Rotate
NONE
```

- For 90° rotation (clockwise, portrait):

```
xrandr -o right && xsetwacom --set "Serial Wacom Tablet" Rotate
CW
```

- For 180° rotation (landscape):

```
xrandr -o inverted && xsetwacom --set "Serial Wacom Tablet"
Rotate HALF
```

- For 270° rotation (counterclockwise, portrait):

```
xrandr -o left && xsetwacom set --"Serial Wacom Tablet" Rotate
CCW
```

Note that the commands above depend on the output of the `xsetwacom list` command. Replace "Serial Wacom Tablet" with the output for the stylus or the touch device. If you have a Wacom device with touch support (you can use your fingers on the tablet to move the cursor), you need to rotate also the touch device.

26.8 For More Information

Some of the applications mentioned here do not offer integrated online help, but you can find some useful information about usage and configuration in your installed system in `/usr/share/doc/package/packagename` or on the Web:

- For the Xournal manual, refer to <http://xournal.sourceforge.net/manual.html>
- The Jarnal documentation is located at <http://www.dklevine.com/general/software/tc1000/jarnal.htm#documentation>
- Find the xstroke man page at <http://davesource.com/Projects/xstroke/xstroke.txt>
- Find a HOWTO for configuring X on the Linux Wacom Web site: <http://linuxwacom.sourceforge.net/index.php/howto/x11>
- Find a very informative Web site about the Dasher project at <http://www.inference.phy.cam.ac.uk/dasher/>

- Find more information and documentation about CellWriter at <http://risujin.org/cellwriter/>
- Information on gnome-display-properties can be found at <http://old-en.opensuse.org/GNOME/Multiscreen>

Copying and Sharing Files

If using multiple operating systems (OS) simultaneously, it is often necessary to exchange files among them. Different systems may reside on different partitions on the same machine or on different machines across your network. There are various approaches to file exchange with different basic instructions and possible pitfalls.

WARNING: Scenarios for Private Home Networks Only

Do not use the following scenarios in networks other than your own private and trusted home network that is protected by a firewall. Implementing high security measures for the configurations featured in the following sections is beyond the scope of this document.

Exchanging data may encompass either one of the following tasks:

Copying

To copy your data means to transfer your data from one system to the other. This results in identical objects on both the source and the target system.

Synchronizing data is a special way to copy data. If you change a file on one computer, it is automatically changed on the other computer after the synchronization. For example, think of a laptop that contains your modified files and you want to have the same contents on your desktop computer.

Sharing

Sharing your files means establishing a client/server relationship. The server provides files that can be accessed by the client. When changing a file, you modify it

on the server, not locally on the client. File servers typically serve a large number of clients simultaneously.

27.1 Scenarios

The following list provides a number of possible scenarios involving file transfer:

Different OS on the Same Computer

Many users have an operating system preinstalled by their vendor and run Linux in a separate partition. Refer to Section 27.4, “Accessing Files on Different OS on the Same Computer” (page 469) for more information.

Different Computers Not Connected by a Network

Save the data to any media (CD, DVD, USB flash drive, or external hard disk) and connect these to the target machine to copy your files. This solution is inexpensive, intuitive, and straightforward. However, you need the appropriate drives or ports on both computers. Additionally the operating systems have to understand the filesystem.

Media are suited to occasional file transfers with limited file size. If you need a more permanent solution, consider connecting them with a network.

Different Computers Connected to the Same Network

Set up a server of any kind on one computer, connect the server and the client, and transfer the files from server to client. Choose from various protocols available and pick the one that best matches your needs and expertise.

The client/server setup requires more expertise and maintenance efforts, but is better suited to routine transfer needs and exchange with multiple systems. If you are looking for a permanent file exchange, choose a client/server-based method. This method does not impose any limits on the amount of data that can be transferred. See Section 27.2, “Access Methods” (page 467).

Different Computers on Different Networks

This scenario requires connection of different networks and is beyond the scope of this document. Transfer files as if the computers were not connected to a network.

27.2 Access Methods

The following methods and protocols are well-suited to file transfer and sharing.

FTP

Use FTP (File Transfer Protocol) if you need to exchange files very often and with different users. Set up an FTP server on one system and access it with clients. There are many graphical client applications available for FTP on Windows*, MacOS, and Linux. Depending on how your FTP server is used, enable read and write permissions. See Section 27.5.4, “Copying Files with FTP” (page 476) for more details on FTP.

NFS

NFS (Network File System) is a client/server system. A server exports one or more directories that can be imported by a client. For more information, see Chapter 18, *Sharing File Systems with NFS* (page 321).

Use NFS if you share files very often and for different users. Generally, this protocol is more common in the Linux world than in the Windows world. An NFS export integrates well into your Linux system and you can browse the imported directory structure like any other folder on your local machine. Depending on your configuration, enable either read or write permissions or both on the server. In general, for a home user it makes sense to allow read and write access.

rsync

Use rsync to transfer regularly large volumes of data that does not change considerably. It is available on Linux and Windows. A typical use case for rsync is managing data backups. Refer to the manual page of the `rsync` command and Section 27.5.2, “Transferring Files with rsync” (page 472) for more information.

Unison

Unison is an alternative to rsync. It is used to regularly synchronize files between different computers but has the advantage to behave bidirectionally. Refer to the manual page of the Unison command and Section 27.5.3, “Transferring Files with Unison” (page 474) for more information. Unison is available on Linux and Windows.

CSync

CSync is an alternative to Unison. Just like Unison it synchronizes files bidirectionally. However, its architecture is modular so it can be extended with plug-ins. See <http://www.csync.org> for more details.

SMB

Samba is a client/server system and an implementation of the SMB protocol. It is usually used in Windows networks, but is supported by several operating systems. Refer to Chapter 19, *Samba* (page 335) for more information about Samba.

Use Samba if you need to share files very often and with different users, especially to Windows systems. Samba as a Linux-only solution is uncommon, use NFS instead. For more information about setting up a Samba server, refer to Section 27.8, “Sharing Files between Linux and Windows with Samba” (page 482).

SSH

SSH (Secure Shell) enables a secure connection between computer. The SSH suite consists of several commands and uses public key encryption to authenticate users. For more information, see Chapter 12, *SSH: Secure Network Operations* (↑Security Guide).

Use SSH if you copy files occasionally over an untrusted network and if you are the only user doing so. Although there are graphical user interfaces available, SSH is mainly considered a command line utility and is available on Linux and Windows.

27.3 Accessing Files Using a Direct Connection

This section describes one way to exchange files between two computers using an Ethernet crossover cable.

You need:

- Ethernet crossover cable. For further information see: http://en.wikipedia.org/wiki/Ethernet_crossover_cable
- openSUSE on both computers
- An established connection.
- The SSH daemon running on both machines. To start the service, run the command `rcssh start as root`.

Proceed as follows:

Procedure 27.1: *GNOME*

- 1 Start Nautilus.
- 2 Click on *File > Connect to Server*.
- 3 Set the *Service Type* to *ssh*.
- 4 Enter the IP address and port of the remote computer (default is 22).
- 5 Specify the folder you want to open on the remote Computer.
- 6 Click *Connect*.

Procedure 27.2: *KDE*

- 1 Start Dolphin.
- 2 Click on *Network, Add Network*. Re-attach the pane if it is not available with *View > Panels > Places*.
- 3 Set the type of network to *Secure shell (ssh)*.
- 4 Enter any name and the correct user, IP address, port (default is 22) and folder of the remote Computer. It is also possible to create an icon for this connection by enabling the checkbox below. This connection icon appears in the *Network* tab in Dolphin.
- 5 Click on *Save & Connect* a dialog box opens and requests the password.

A new window containing the files of the remote computer will be opened.

27.4 Accessing Files on Different OS on the Same Computer

New computers generally ship with a preinstalled operating system, usually Windows. If you have installed Linux on a different partition, you might want to exchange files between the different operating systems.

Windows cannot read Linux partitions by default. If you want to exchange files between these two operating systems, you have to create an “exchange partition”. For a

more direct approach, see <http://www.fs-driver.org/> to get a driver supporting an ext2 filesystem on Windows. The following file systems are used by Windows and can be accessed from a Linux machine:

FAT

Various flavors of this file system are used by MS-DOS and Windows 95 and 98. You can create this type of file system with YaST. It is possible to read and write files on FAT partitions from Linux. The size of a FAT partition (and even the maximum size of a single file) is subject to restrictions, depending on the FAT version. See <http://en.wikipedia.org/wiki/VFAT> for more information about FAT file systems.

NTFS

The NTFS file system is used by Windows. openSUSE includes write access support to the NTFS file system. See <http://en.opensuse.org/SDB:NTFS> for more information about NTFS-3g.

During the installation of openSUSE, your Windows partitions are detected. After starting your Linux system, the Windows partitions usually are mounted. These are possible ways of accessing your Windows data:

KDE

Press **Alt + F2** and enter `sysinfo:/.` A new window opens displaying the characteristics of your machine. *Disk Information* lists your partitions. Look at those that are of the file system type `ntfs` or `vfat` and click these entries. If the partition is not already mounted, KDE mounts the partition now and displays the contents.

Command Line

Just list the contents of `/windows` to see one or more directories containing your Windows drives. The directory `/windows/c` maps to the Windows `C:\` drive, for example.

NOTE: Changing the Accessibility of Windows Partitions

Initially, Windows partitions are mounted read-only for normal users to avoid accidental damage to the file system. To grant normal users full access to a mounted Windows partition, change the mount behavior of this Windows partition. Refer to the manual page of the `mount` command for more information on mount options for `vfat` and to the manual page of `ntfs-3g` on mount options for NTFS.

27.5 Copying Files between Linux Computers

Linux offers a rich set of protocols you can use to copy files between computers. Which protocol you use depends on how much effort you want to invest and whether it needs to be compatible with future Windows installations. The following sections feature various methods to transfer files from and to Linux computers. Make sure that you have a working network connection, because otherwise they will not work. All scenarios rely on working name resolution in the network. If your network does not include a name service, use IP addresses directly or add the IP addresses along with respective hostnames to `/etc/hosts` on all clients.

The following example IP addresses and hostnames are used across this section:

Target Hostname	jupiter.example.com
Target IP	192.168.2.100
Source Hostname	venus.example.com
Source IP	192.168.2.101
User	tux

27.5.1 Copying Files with SSH

The following requirements must be met on both computers that are accessed via SSH:

1. If you use a hostname, make sure each hostname is listed in `/etc/hosts` on both computers (see Section 13.6.1.6, “`/etc/hosts`” (page 261).) If you use SSH with IP addresses, you do not need to change anything.
2. If you use a firewall, open the SSH port. To do so, start YaST, and select *Security and Users > Firewall*. Go to *Allowed Services* and check whether *SSH* is displayed as part of the list. If this is not the case, select *SSH* from *Service to Allow* and click *Add*. Apply your changes and leave YaST with *Next* and *Finish*.

To copy files from one computer to another, you need to know where the files are located. For example, to copy the single file `/srv/foo_file` from computer `jupiter.example.com` to the current directory, use the following `scp` command (the dot represents the current directory as the copy target location):

```
scp tux@jupiter.example.com:/srv/foo_file .
```

To copy a whole directory structure, use the recursive mode of `scp`:

```
scp -r tux@jupiter.example.com:/srv/foo_directory .
```

If your network does not provide name resolution, use the server's IP address directly:

```
scp tux@192.168.2.100:/srv/foo_file .
```

If you do not know exactly where your files are, use the `sftp` command. Copying files in KDE or GNOME with SFTP is very simple. Proceed as follows:

- 1 Press **Alt + F2**.
- 2 Enter the following at the address prompt (correct it to your own values):

```
sftp://tux@jupiter.example.com
```
- 3 Confirm the question regarding of authenticity and enter the password of `tux` on `jupiter.example.com`.
- 4 Drag and drop the desired files or directories to your desktop or a local directory.

KDE provides another protocol called `fish` that can be used if `sftp` is not available. The use of this protocol is similar to `sftp`. Just replace the `sftp` protocol prefix of the URL with `fish`:

```
fish://tux@jupiter.example.com
```

27.5.2 Transferring Files with `rsync`

`rsync` is useful for archiving or copying data and can also be used as a daemon to provide directories to the network (see Procedure 27.3, “Advanced Setup for `rsync` Synchronization” (page 473)).

Before using `rsync` to synchronize files and directories between different computers, make sure that the following requirements are met:

1. The package `rsync` is installed.
2. Identical users are available on both systems.

3. Enough disk space is available on the server.
4. If you want to benefit from rsync's full potential, make sure that rsyncd is installed on the system to use as the server.

27.5.2.1 rsync Basic Mode

The basic mode of operation of rsync does not require any special configuration. rsync mirrors complete directories onto another system. Its usage is not much different from a regular copying tool, such as scp. The following command creates a backup of the home directory of tux on a backup server called jupiter:

```
rsync -Hbaz -e ssh /home/tux/ tux@jupiter:backup
```

Use the following command to restore your backup (without option -b):

```
rsync -Haz -e ssh tux@jupiter:backup /home/tux/
```

27.5.2.2 rsync Daemon Mode

Start the rsyncd daemon on one of your systems to make use of the full functionality of rsync. In this mode, it is possible to create synchronization points (modules) that can be accessed without an account. To use the rsyncd daemon, proceed as follows:

Procedure 27.3: *Advanced Setup for rsync Synchronization*

- 1 Log in as root and install the rsync package.
- 2 Configure your synchronization points in /etc/rsyncd.conf. Add a point with its name in brackets and add the path keyword like in the following example:

```
[FTP]
path = /srv/ftp
comment = An Example
```

- 3 Start the rsyncd daemon as root with `rcrsyncd start`. To start the rsync service automatically during each system boot, run `insserv rsyncd`.
- 4 List all files located in the /srv/ftp directory (note the double colon):

```
rsync -avz jupiter::FTP
```
- 5 Initiate the transfer by providing a target directory (in this example, the current directory is represented by a dot):

```
rsync -avz jupiter::FTP .
```

By default, files are not deleted while synchronizing with `rsync`. To force file deletion, add the `--delete` option. To make sure that `--delete` does not accidentally remove newer files, use the `--update` option instead. Any conflicts that arise must be resolved manually.

27.5.3 Transferring Files with Unison

Before using Unison to synchronize files and directories between different computers, make sure that the following requirements are met:

1. The package `unison` is installed.
2. Enough disk space is available on your local and remote computer.
3. If you want to benefit from Unison's full potential, make sure that Unison is also installed and running on the remote computer.

In case you need help, run Unison with the `-doc topics` option to get a full list of available sections.

For permanent settings, Unison allows the creation of *profiles* that specify Unison preferences such as the directories (roots) to synchronize, which types of files to ignore, and other options. The profiles are stored as text files in `~/ .unison` with the file extension `*.prf`.

27.5.3.1 Using the GUI

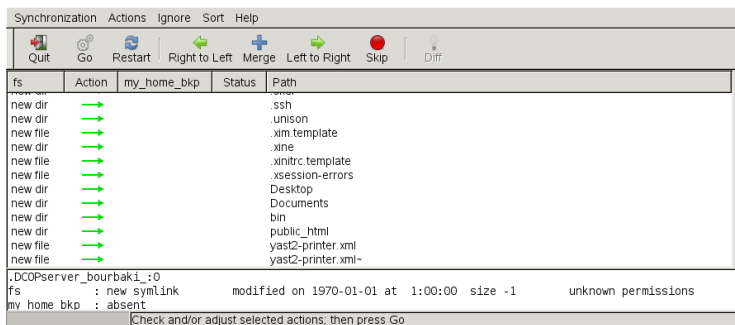
To synchronize different directories with Unison's GUI, proceed as follows:

- 1 Start Unison by pressing `Alt + F2` and entering `unison`.
- 2 If you run Unison for the first time and without any further options, you are prompted for a source directory. Enter the source directory you want to synchronize and click *OK*.
- 3 Enter the target directory. It can be either local or remote. If you want to synchronize to a remote directory, choose the method (SSH, RSH or Socket) and enter the hostname and an optional user.

- 4 If you have not synchronized these two directories before, a warning dialog appears, informing you that Unison will now compare the contents of those directories. Close the warning with *OK* and wait until Unison has collected the information from both directories and displays the differences in the main window.

The left column shows the source directory you have selected, the third column shows the target directory. If there are differences between the directories, the *Action* column shows a symbol, proposing an action. A green arrow indicates that a file has been modified, added or deleted in the source or the target directory. The direction of the arrow indicates the direction that the change would be propagated if you performed the synchronization now. A question mark indicates a conflict (both files have been changed and Unison cannot decide which one to overwrite).

Figure 27.1: *File Synchronization Proposal*



- 5 To modify the proposals Unison shows for each file (for example, if you want to change the direction), select the file and click *Right to Left* or *Left to Right*. With *Skip*, exclude a file from synchronization. The symbol in the *Action* column changes accordingly.
- 6 To start the synchronization, click *Go*.

The next time you start Unison, a dialog box shows the existing profiles, each specifying a pair of directories to be synchronized. Select a profile or create a new profile (for another pair of directories) and perform the synchronization as described above.

27.5.3.2 Using the Command Line

Unison can also be operated through the command line. To synchronize a local directory to a remote computer, proceed as follows:

- 1 Open a shell and enter the following command:

```
unison -ui text DIR
      ssh://tux@jupiter.example.com//PATH
```

Replace the placeholders with the respective values.

- 2 Unison asks you what to do with your files and directories, for example:

```
local                jupiter
      <---- new file  dir [f]
```

- 3 Press F if you want to follow Unison's recommendation. For other commands, press ?.
- 4 Proceed with y, if you want to propagate your updates.

27.5.4 Copying Files with FTP

Before configuring your FTP server, make sure that the following requirements are met:

1. The package `vsftpd` is installed.
2. You have `root` access to your FTP server.
3. Enough disk space is available on your computer.

WARNING: For Home Networks Only

This setup is suited for use in home networks only. Do not deploy it to sites unprotected by firewalls and do not enable world wide access.

To configure an FTP server, proceed as follows:

- 1 Prepare the FTP server:

- 1a Open a shell, log in as `root`, and save a backup copy of `/etc/vsftpd.conf`:

```
cp /etc/vsftpd.conf /etc/vsftpd.conf.bak
```

- 1b Create an access point for anonymous FTP

```
mkdir ~ftp/incoming
```

```
chown -R ftp:ftp ~ftp/incoming
```

- 2 Replace the configuration files according to the preferred scenario (refer to the manual page of `vsftpd.conf` for advanced configuration options):

Allowing Anonymous Read and Write Access

```
#
listen=YES

# Enable anonymous access to FTP server
anonymous_enable=YES

#
local_enable=YES
# Enable write access
write_enable=YES
anon_upload_enable=YES
anon_mkdir_write_enable=YES
dirmessage_enable=YES
# Write log file
xferlog_enable=YES
connect_from_port_20=YES
chown_uploads=YES
chown_username=ftp
ftpd_banner=Welcome to FTP service.
anon_root=/srv/ftp
```

Grant Restricted Permissions to FTP Users (Home Only)

```
chroot_local_users=YES
```

- 3 Restart the FTP server:

```
rcvsftpd start
```

On the client, just enter the URL `ftp://HOST` in your browser or FTP client. Replace `HOST` with the hostname or IP address of your server. There are many graphical user interfaces available that are suited to browsing the contents of your FTP server. For a list of them, just enter FTP at the search prompt of the YaST package manager.

27.6 Copying Files between Linux and Windows Computers with SSH

To transfer files from Linux to Windows using SSH, choose one of the following applications:

PuTTY

PuTTY is a suite of different command line tools for working with an SSH daemon. Download it from <http://www.chiark.greenend.org.uk/~sg-tatham/putty.html>.

WinSCP

WinSCP is very similar to PuTTY, but includes a graphical user interface. Choose from an Explorer or Norton Commander style. Download it from <http://winscp.net>.

To copy a file from Windows to Linux with PuTTY, proceed as follows (on the Windows machine):

- 1 Start PSCP.
- 2 Enter the hostname of your SSH server.
- 3 Enter your login and password to the SSH server.

To connect from Windows to Linux with WinSCP, proceed as follows (on the Windows machine):

- 1 Start WinSCP.
- 2 Enter the hostname of the SSH server and username.
- 3 Click *Login* and acknowledge the resulting warning.
- 4 Drag and drop any files or directories from or to your WinSCP window.

NOTE: SSH Fingerprint

With both PuTTY and WinSCP, you must accept the SSH fingerprint when you log in for the first time.

27.7 Sharing Files between Linux Computers

The following sections feature various methods for sharing data. Use one of these if you are looking for a permanent solution for data sharing.

27.7.1 Transferring Files with NFS

To configure the server, proceed as follows:

1 Prepare the system:

1a Open a shell, log in as `root`, and grant write permissions to all users:

```
mkdir /srv/nfs
chgrp users /srv/nfs
chmod g+w /srv/nfs
```

1b Make sure that your user name and user ID is known on the client as well as on the server. Refer to Chapter 10, *Managing Users with YaST* (↑Start-Up) for detailed instructions about how to create and manage user accounts.

2 Prepare the NFS server:

2a Start YaST as `root`.

2b Select *Network Services > NFS Server* (this module is not installed by default. If it is missing in YaST, install the package `yast2-nfs-server`).

2c Enable NFS services with *Start*.

2d Open the appropriate firewall port with *Open Port in Firewall* if you are using a firewall.

3 Export the directories:

3a Click *Add directory* and select `/srv/nfs`.

3b Set the export options to:

```
rw,root_squash,async
```

3c Repeat these steps, if you need to export more than one directory.

4 Apply your settings and leave YaST. Your NFS server is ready to use.

To manually start the NFS server, enter `rcnfsserver start` as root. To stop the server, enter `rcnfsserver stop`. By default, YaST takes care of starting this service at boot time.

To configure the client, proceed as follows:

- 1** Prepare the NFS client:
 - 1a** Start YaST as `root`.
 - 1b** Select *Network Services > NFS Client*.
 - 1c** Activate *Open Port in Firewall* if using a firewall.
- 2** Import the remote file system:
 - 2a** Click *Add*.
 - 2b** Enter the name or IP address of the NFS server or click *Choose* to automatically scan the network for NFS servers.
 - 2c** Enter the name of your remote file system or automatically choose it with *Select*.
 - 2d** Enter an appropriate mount point, for example `/mnt`. If you repeat this step with another exported file system, make sure you choose another mount point than `/mnt`.
 - 2e** Repeat these steps if you need to import more than one external directory.
- 3** Apply your settings and leave YaST. Your NFS client is ready to use.

To start the NFS client manually, enter `rcnfs start`.

NOTE: Consistent User Names

If your home network is used by just a small number of users, set up identical users manually on all machines. If, however, you need a larger consistent user base across a larger home network, consider using NIS or LDAP to manage user data. For further information, refer to Chapter 3, *Using NIS* (↑Security Guide) and Chapter 4, *LDAP—A Directory Service* (↑Security Guide).

27.7.2 Sharing Files with Samba

This section introduces various methods to access files on a Samba server. Both KDE and GNOME ship with graphical tools for working with Samba shares. There is also a command line tool for accessing Samba servers.

27.7.2.1 Accessing Shares with KDE and GNOME

Both the KDE and GNOME desktops can access Samba shares through their file browsers. To access your share, proceed as follows:

- 1 Press **Alt + F2** and enter `smb://jupiter.example.com/share`.

The syntax of this URL is `smb://HOST/SHARENAME` with *HOST* representing the hostname (`jupiter.example.com`) or IP address and *SHARENAME* representing the share. See Step 3b (page 483).

- 2 Log in with the username and password. The password is set in Step 4 (page 483) or just hit **Enter** if no password is needed.
- 3 Drag and drop any files or directories from or to your window.

If you do not know your workgroup, enter `smb:/` to list all workgroups available in your network. The Smb4K tool (package `smb4k`) can also be used to display all workgroups in your network and mount them on demand.

27.7.2.2 Accessing Shares from the Command Line

If you prefer using the command line, use the `smbclient` command. To log in to your Samba server, run:

```
smbclient //jupiter/share -U tux
```

Omit the `-U` option if you are the current user `tux`. After logging in successfully, use some basic commands like `ls` (list contents), `mkdir` (create directory), `get` (download file), and `put` (upload file). Use `help` to display all commands. Refer to the manual page of `smbclient` for more information.

27.8 Sharing Files between Linux and Windows with Samba

Samba is the first choice for transferring files between Windows and Linux machines. These are the most common use cases for Samba:

Transferring Files from Linux to Windows with the SMB Scheme

In the easiest case you do not have to configure a Linux server. Use the `smb://` scheme. For more information, see Section 27.7.2.1, “Accessing Shares with KDE and GNOME” (page 481). Make sure that your workgroup is identical on both systems and that your directories are shared.

Transferring Files from Windows to Linux Using a Server

Configure a Samba server on your Linux computer. See Procedure 27.4, “Setting Up a Samba Server” (page 482).

TIP: Using Default Registry Entries for Your Windows System

Some Windows versions (95, 98) require a small change in the registry for enabling a different password authentication method. Simplify this step by installing the `samba-doc` package and copy the file `/usr/share/doc/packages/samba/registry` to your Windows drive. Start Windows and incorporate the changes by double-clicking on this file.

Procedure 27.4: Setting Up a Samba Server

To set up a Samba server, do the following:

1 Prepare the Samba server:

1a Start YaST as `root`.

1b Install the `samba` package.

1c Create a directory (for example, `/srv/share`).

2 Create the server configuration:

2a Select *Network Services > Samba Server*.

2b Select one of the workgroups or enter a new one (for example, Penguin).

2c Check *Primary Domain Controller (PDC)*

2d Select *During Boot* if the Samba service should be started every time your computer boots. Otherwise select *Manually*.

2e Activate *Open Port in Firewall* if you use a firewall.

3 Create your Windows share:

3a Change to the *Shares* tab and click *Add*.

3b Enter a name and description. The *Share Name* is used for accessing the share from your clients. *Share Description* describes the purpose of the share.

3c Select your path (for example, `/src/share`).

3d Proceed with *OK*.

3e Activate *Allow Users to Share Their Directories*.

4 Provide a password for all users that are allowed to use this service:

```
smbpasswd -a tux
```

For easier configuration, just hit **Enter** to leave the password empty. Take into account that the usernames on your Windows and Linux computers are probably different. Configuring a consistent user base for both Windows and Linux is beyond the scope of this document.

5 Start the Samba server:

```
rcnmb start  
rcsmb start
```

To check if everything has been successfully configured, enter:

```
smbclient -L localhost
```

After you hit **Enter**, you should get something like the following:

```
Anonymous login successful
```

```

Domain=[PENGUIN] OS=[Unix] Server=[Samba 3.0.22-11-SUSE-CODE10]

Sharename      Type      Comment
-----
share          Disk      Shared directory
netlogon       Disk      Network Logon Service
IPC$           IPC       IPC Service (Samba 3.0.22-11-SUSE-CODE10)
ADMIN$         IPC       IPC Service (Samba 3.0.22-11-SUSE-CODE10)
Anonymous login successful
Domain=[PENGUIN] OS=[Unix] Server=[Samba 3.0.22-11-SUSE-CODE10]

Server          Comment
-----
SUSE-DESKTOP    Samba 3.0.22-11-SUSE-CODE10

Workgroup       Master
-----
TUX-NET         jupiter

```

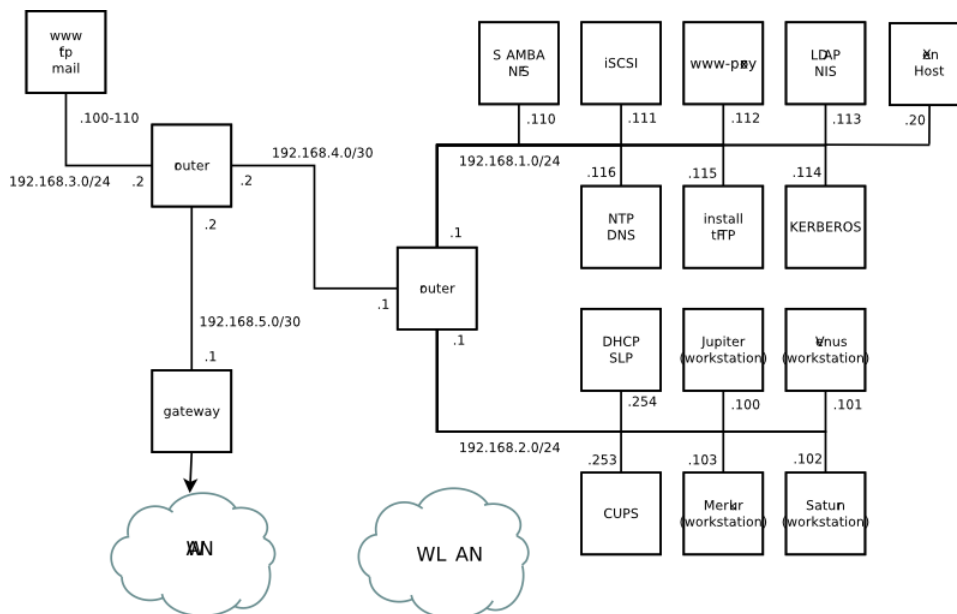
27.9 For More Information

- <http://en.wikipedia.org/wiki/VFAT>
- <http://en.wikipedia.org/wiki/NTFS>
- <http://en.wikipedia.org/wiki/Fstab>
- http://en.wikipedia.org/wiki/Network_File_System
- http://en.wikipedia.org/wiki/File_Transfer_Protocol
- <http://en.wikipedia.org/wiki/SSH>
- <http://en.wikipedia.org/wiki/Rsync>
- http://en.wikipedia.org/wiki/Samba_software

A

An Example Network

This example network is used across all network-related chapters of the openSUSE® documentation.





GNU Licenses

This appendix contains the GNU General Public License version 2 and the GNU Free Documentation License version 1.2.

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a)** You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b)** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c)** If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a)** Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b)** Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c)** Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program's name and an idea of what it does.  
Copyright (C) yyyy name of author
```

```
This program is free software; you can redistribute it and/or  
modify it under the terms of the GNU General Public License  
as published by the Free Software Foundation; either version 2  
of the License, or (at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free Software  
Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details  
type `show w'. This is free software, and you are welcome  
to redistribute it under certain conditions; type `show c'  
for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright  
interest in the program `Gnomovision'  
(which makes passes at compilers) written  
by James Hacker.
```

```
signature of Ty Coon, 1 April 1989  
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License [<http://www.fsf.org/licenses/lgpl.html>] instead of this License.

GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties; any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever

to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users be-

yond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```


If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

