

openSUSE

11.4

www.suse.com

October 30, 2011

Reference



Reference

Copyright © 2006– 2011 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>. Linux* is a registered trademark of Linus Torvalds. All other third party trademarks are the property of their respective owners. A trademark symbol (®, ™ etc.) denotes a Novell trademark; an asterisk (*) denotes a third party trademark.

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

Contents

About This Guide	v
------------------	---

Part 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	8
1.4	System Start-Up for Installation	8
1.5	The Boot Screen	8
1.6	Welcome	12
1.7	Installation Mode	13
1.8	Clock and Time Zone	15
1.9	Desktop Selection	16
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	43
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	64

3 Advanced Disk Setup **67**

3.1	Using the YaST Partitioner	67
3.2	LVM Configuration	75
3.3	Soft RAID Configuration	80

Part II Managing and Updating Software **85**

4 Installing or Removing Software **87**

4.1	Definition of Terms	88
4.2	Using the KDE Interface (Qt)	89
4.3	Using the GNOME Interface (GTK+)	95
4.4	Managing Software Repositories and Services	101
4.5	Keeping the System Up-to-date	105

5 YaST Online Update **111**

5.1	The Online Update Dialog	112
5.2	Installing Patches	115
5.3	Automatic Online Update	116

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://www.novell.com/documentation/opensuse114> 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 1)

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 Novell 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 documentation updates at <http://www.novell.com/documen->

[tation](#) 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 against the component *Documentation* for 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 and suggestions about 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.novell.com/documentation/feedback.html> and enter your comments there.

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

- , + 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 `susedoc` that is shipped with openSUSE.

5 Source Code

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

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

6 Acknowledgments

With a lot of voluntary commitment, the developers of Linux cooperate on a global scale to promote the development of Linux. We thank them for their efforts—this

distribution would not exist without them. Furthermore, we thank Frank Zappa and Pawar. Special thanks, of course, 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 6). 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 7).

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 8).

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 6), Section 1.2.2, “Installing from a Network Source without SLP” (page 7), or Chapter 2, *Remote Installation* (page 35) for details.

Installing with `openSUSE 11.4 Installer` from Windows

Choose this installation option if you prefer a smooth transition from using Windows to using Linux. `openSUSE 11.4 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 11.4 Installer` from Windows” (page 7) 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 11.4 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.

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 43)), 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 11.4 Installer from Windows

openSUSE 11.4 *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 11.4 *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 11.4 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 11.4 *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 Windows 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 “Resizing a Windows Partition” (page 18) 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 to boot it. The relevant options are:

Installation

The normal installation mode. All modern hardware functions are enabled. In case the installation fails, see *F5Kernel* (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.

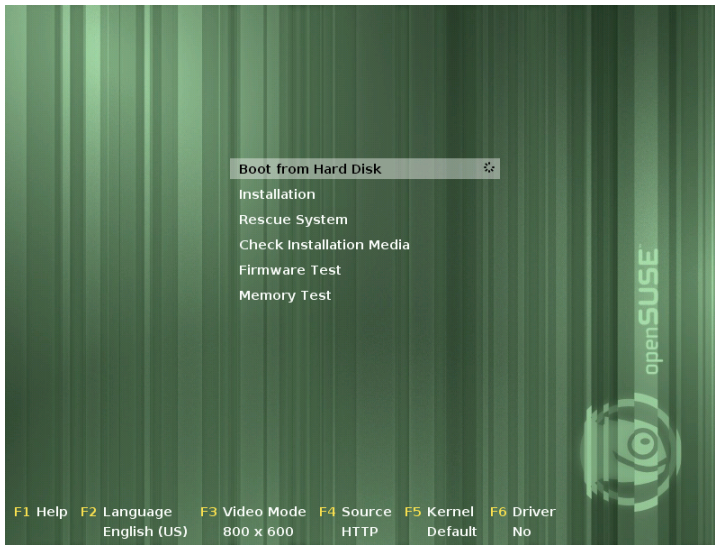
Firmware Test

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 24, *SLP Services in the Network* (page 395).

F5Kernel

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.

F6Driver

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.

F7Arch

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

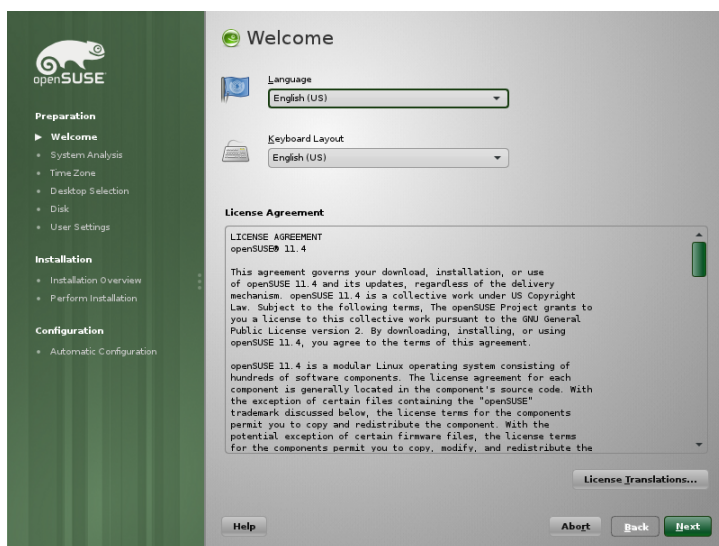
selection fields contain a letter with an underscore. Use Alt + Letter to select a button or a selection directly instead of navigating there with the Tab button.

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 10, *Changing Language and Country Settings with YaST* (page 165).

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.

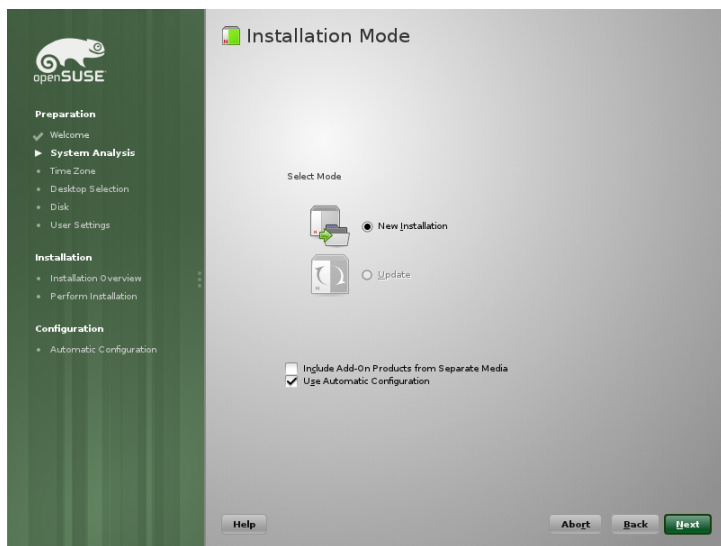
New installation

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 15, *Upgrading the System and System Changes* (page 209).

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 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 15).

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 “Network Setup” (page 14). 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 7, *Installing Add-On Products* (page 123).

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.

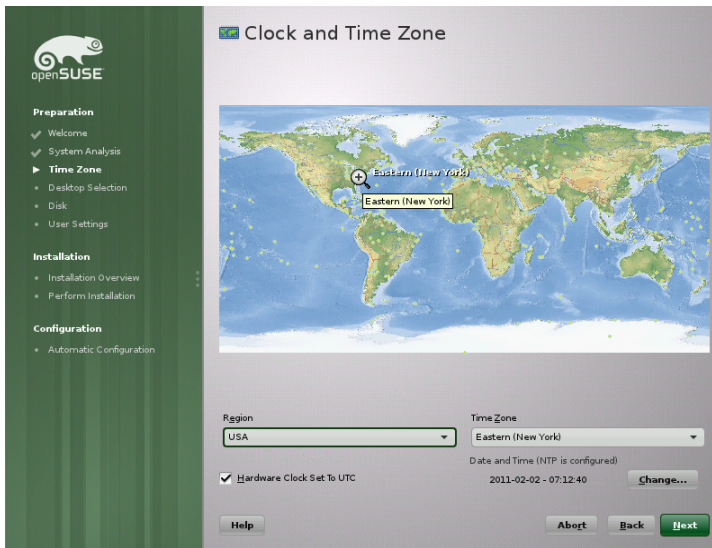
If an ethernet network adapter is not already connected, a warning will open. Make sure the network cable is plugged in and choose *Yes, Use It*. If your network is equipped with a DHCP server, choose *Automatic Address Setup (via DHCP)*. To manually set up the network choose *Static Address Setup* and 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 check box *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 27, *Time Synchronization with NTP* (page 433) for more information on configuring the NTP service. When finished, click *Accept* to continue the installation.

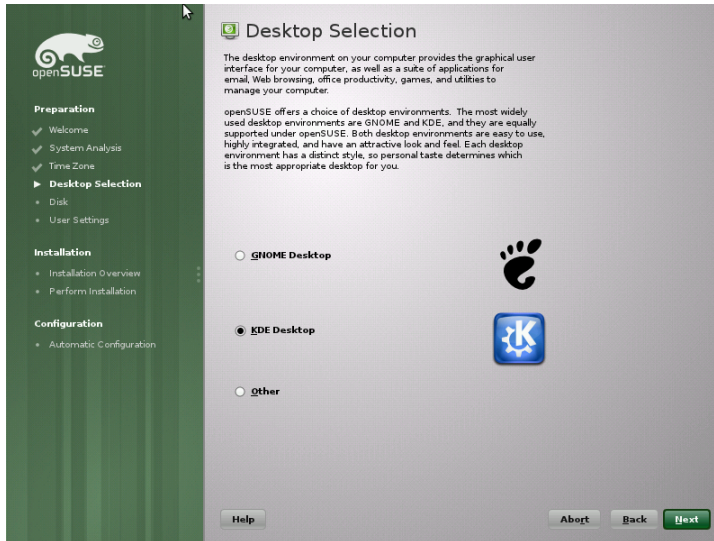
1.9 Desktop Selection

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*



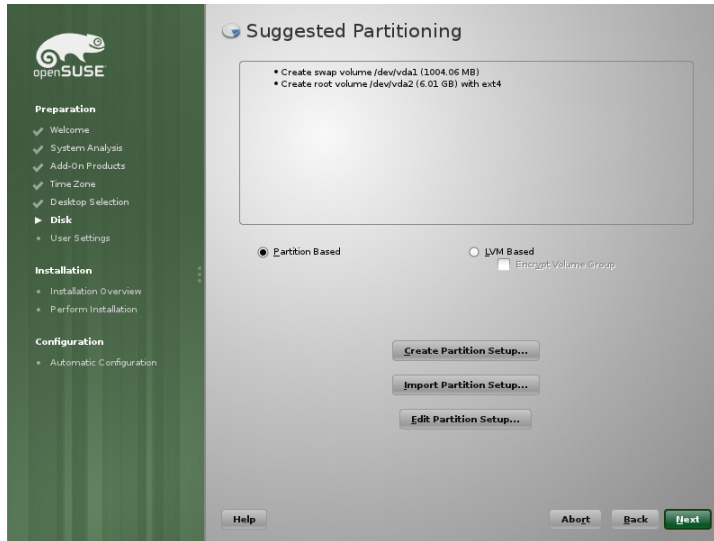
1.10 Suggested Partitioning

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 Windows 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 75) 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 67) 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 existing 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.

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.

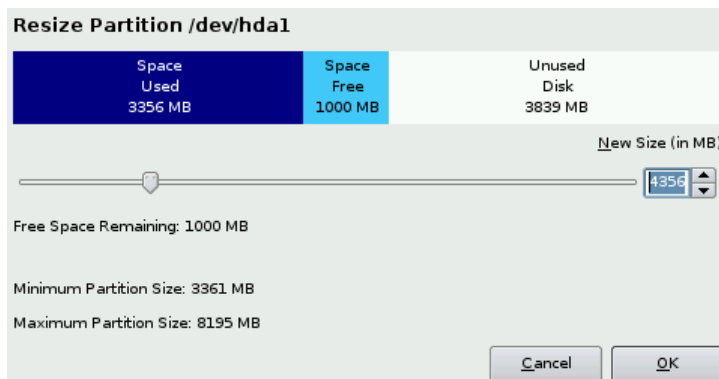
To *Shrink the Windows Partition*, you need to interrupt the installation and boot Windows to prepare the partition before shrinking it. For all Windows file systems, proceed as follows:

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*



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 67)

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).

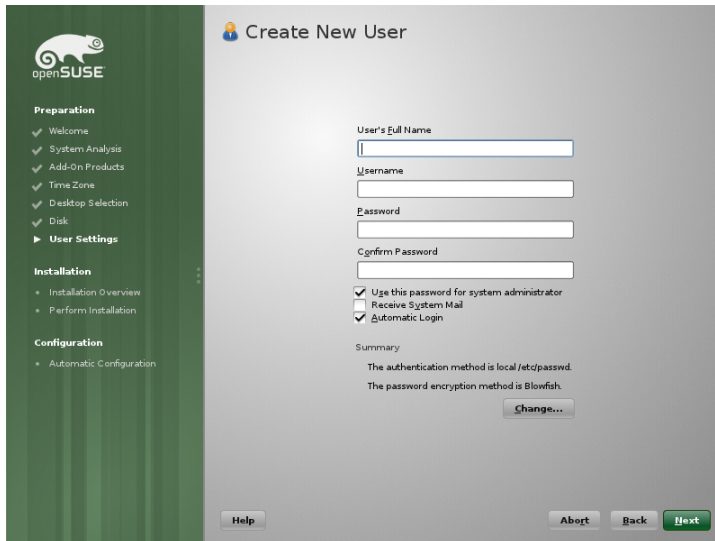
After entering the first name and last name, either accept the proposal or specify a new *Username* that will be used to log in. Finally, enter a password for the user. Reenter it for confirmation (to ensure that you did not type something else by mistake). To provide effective security, a password should 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*



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 9, *Managing Users with YaST* (page 149).

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*.

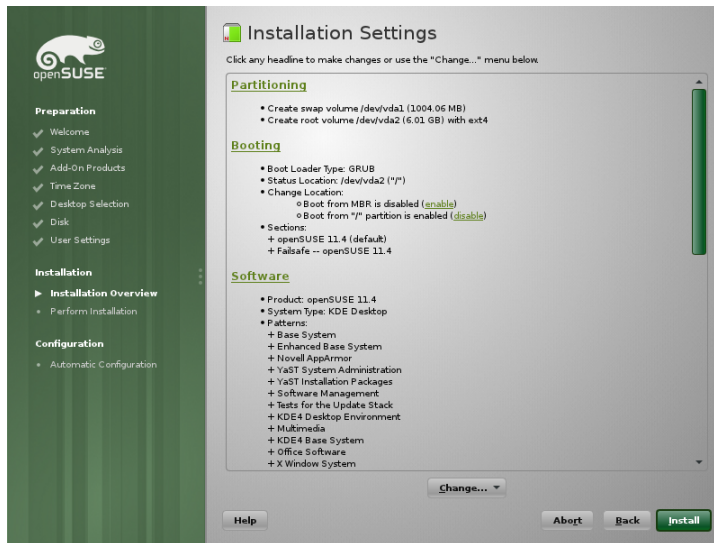
WARNING: The root User

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

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 67).

1.12.2 *Booting*

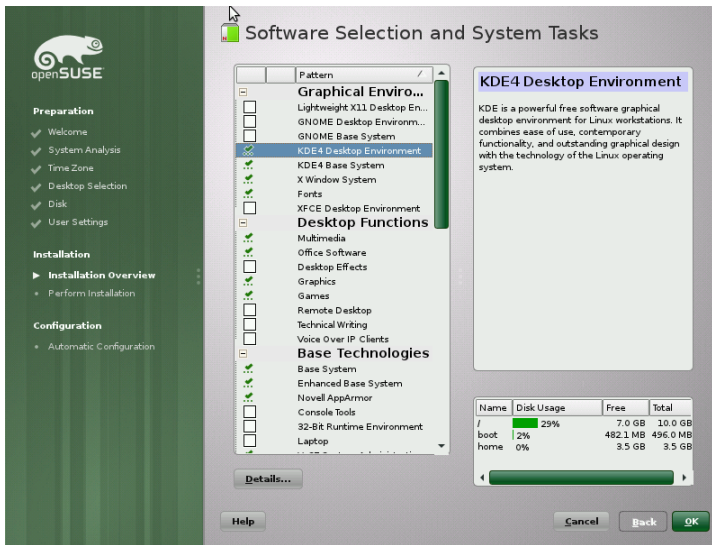
YaST proposes a boot configuration for your system. Other operating systems found on your computer, such as Microsoft Windows or other Linux installations, will automatically be detected and added to the boot loader. However, openSUSE will be booted by default. Normally, you can leave these settings unchanged. If you need a custom setup, modify the proposal for your system. For information, see Section 18.2, “Configuring the Boot Loader with YaST” (page 250). 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 4, *Installing or Removing Software* (page 87).

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 10, *Changing Language and Country Settings with YaST* (page 165).

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 15).

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. Refer to Section 17.2.3, “Configuring System Services (Runlevel) with YaST” (page 233) for more information about runlevel configuration.

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 system, 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.

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, 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.

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 23.4.1, “Configuring the Network Card with YaST” (page 357).

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 35, *Using NetworkManager* (page 557). 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 23.2, “IPv6—The Next Generation Internet” (page 346).

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 14, *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 “Network Setup” (page 14)) 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 network 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.

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 5, *YaST Online Update* (page 111). Directives on how to perform an online update in the installed system are available at Section 4.5, “Keeping the System Up-to-date” (page 105) or Chapter 5, *YaST Online Update* (page 111). 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.

New Local User

If no local user was created in phase one, you can create one in this dialog—otherwise this step is skipped. To create more users, manage groups, modify defaults for new users and set up network authentication, launch *User Management*. Refer to

Chapter 9, *Managing Users with YaST* (page 149) for more information about user management. To skip this step, click *Next* without entering any data.

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.

Hardware Configuration

At the end of the installation, YaST opens a dialog for the configuration of a printer, the sound system and TV cards. 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 12, *Setting Up Hardware Components with YaST* (page 179). 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.

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 GNOME Quick Start (↑GNOME Quick Start) and the KDE Quick Start (↑KDE Quick Start). 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.

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 43). 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 de-

scribed 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 64).
- 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 43). 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 64).
- 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 43). 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 64).
- 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 43). 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 “Connecting to the Installation Program” (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.

2.1.5 Simple Remote Installation via SSH —Dynamic Network Configuration

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

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

- Remote 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 43). 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 “Connecting to the Installation Program” (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.

2.1.6 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 43). 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 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.

2.2 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 configura-

tion. 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 28, *Sharing File Systems with NFS* (page 441).

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

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

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

- 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 24, *SLP Services in the Network* (page 395). More Information about NFS, refer to Chapter 28, *Sharing File Systems with NFS* (page 441).

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 31, *Setting up an FTP server with YaST*

(page 509) 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 30.1.2, “Installation” (page 468).
 - 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/sl.p.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 `rcslp 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 , 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/product
medium 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 26, *DHCP* (page 423).

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.

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.

Setting Up a TFTP Server Using YaST

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

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

Setting Up a TFTP Server Manually

- 1 Log in as `root` and install the packages `tftp` and `xinetd`.
- 2 If unavailable, create `/srv/tftpboot` and `/srv/tftpboot/pxelinux.cfg` directories.
- 3 Add the appropriate files needed for the boot image as described in Section 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/tftpboot
    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 `gfxboot`, `readinfo`, and `framebuffer`.

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

```
insmod=kernel module
```

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

```
netdevice=interface
```

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

```
install=nfs://ip_instserver/path_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, “Booting 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 `vncpassword` 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 hddisk

# 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 harddisk
    localboot 0

implicit      0
display       message
prompt        1
timeout       100
```

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_kernel my_kernel_options
  initrd myinitrd
```

PXELINUX uses the following syntax:

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

Labels are mangled as if they were filenames and they must be unique after mangling. For example, the two labels “v2.6.30” 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 COMBOOT file.

APPEND -

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

LOCALBOOT *type*

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

Argument	Description
0	Perform a normal boot

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 `or` 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 8).

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.

Table 2.1 *Installation (Boot) Scenarios Used in This Chapter*

Installation Scenario	Parameters Needed for Booting	Boot Options
Chapter 1, <i>Installation with YaST</i> (page 3)	None: system boots automatically	None needed

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.

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.

Connecting to the Installation Program

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

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

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

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

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

```
ip_address:display_number
```

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

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

To perform a VNC installation, proceed as follows:

- 1 Launch your preferred Web browser.
- 2 Enter the following at the address prompt:

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

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

2.5.2 SSH Installation

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

Preparing for SSH Installation

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

Connecting to the Installation Program

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

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

Replace *ip_address_of_target* with the actual IP address of the installation target.
- 3 When prompted for a username, enter `root`.
- 4 When prompted for the password, enter the password that has been set with the SSH boot option. After you have successfully authenticated, a command line prompt for the installation target appears.
- 5 Enter `yast` to launch the installation program. A window opens showing the normal YaST screens as described in Chapter 1, *Installation with YaST* (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 (see the chapter about multipath I/O in Storage Administration Guide), and there is also the option to use iSCSI as a networked disk.

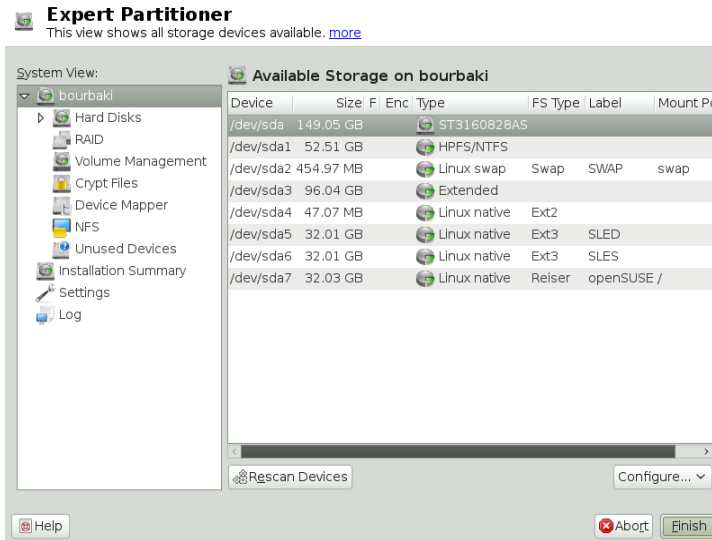
3.1 Using the YaST Partitioner

With the expert partitioner, shown in Figure 3.1, “The YaST Partitioner” (page 68), manually modify the partitioning of one or several hard disks. Partitions can be added, deleted, resized, and edited. Also access the soft RAID and LVM configuration from this YaST module.

WARNING: Repartitioning the Running System

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

Figure 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 NFS.

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 15 on SCSI, SATA, and Firewire disks and 63 on (E)IDE disks. It does not matter which types of partitions are used for Linux. Primary and logical partitions both 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*. If several hard disks are connected, a selection dialog appears in which select a hard disk for the new partition.
- 2 Specify the partition type (primary or extended). Create up to four primary partitions or up to three primary partitions and one extended partition. Within the extended partition, create several logical partitions (see Section 3.1.1, “Partition Types” (page 68)).
- 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 70).

- 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.

3.1.3 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. Possible values include *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.

WARNING: Changing the file system

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

For details on the various file systems, refer to Storage Administration Guide.

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 11, *Encrypting Partitions and Files* (↑Security Guide).

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

When using the mount option *LABEL* to mount a partition, define an appropriate label for the selected partition. 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 9.3.5, “Managing Quotas” (page 157).

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.

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 and its 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.

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–1 GB).

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 parti-

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

Procedure 3.1 Adding a Swap File Manually

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

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

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

- 2** Initialize this swap file with the command

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

- 3** Activate the swap with the command

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

To disable this swap file, 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. However, 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.

For more details about LVM, see the Storage Administration Guide.

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 77), 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 68) and Section 3.1, “Using the YaST Partitioner” (page 67).

Figure 3.2 *Physical Partitioning versus LVM*

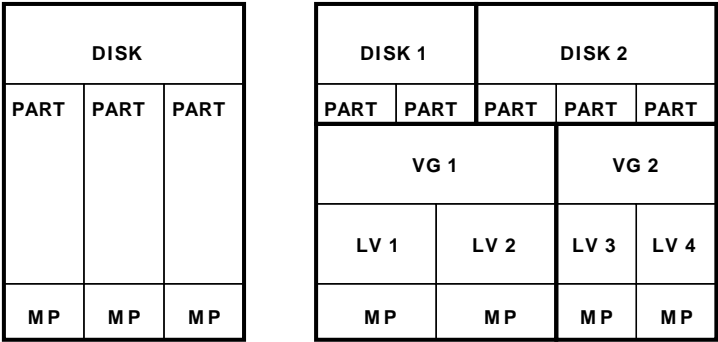


Figure 3.2, “Physical Partitioning versus LVM” (page 76) compares physical partitioning (left) with LVM segmentation (right). On the left side, one single disk has been divided into three physical partitions (PART), each with a mount point (MP) assigned so that the operating system can gain access. On 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 67)) 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.

Creating Volume Groups

If no volume group exists on your system, you must add one (see Figure 3.3, “Creating a Volume Group” (page 78)). 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.

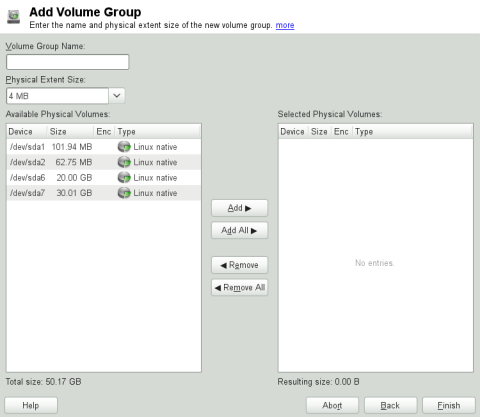
- 1 Enter a name for the VG, for example, `system`.
- 2 Select the desired *Physical Extend Size*. This value defines the size of a physical block in the volume group. All the disk space in a volume group is handled in blocks of this size.

TIP: Logical Volumes and Block Sizes

The possible size of an LV depends on the block size used in the volume group. The default is 4 MB and allows for a maximum size of 256 GB for physical and LVs. The physical size should be increased, for example, to 8, 16, or 32 MB, if you need LVs larger than 256 GB.

- 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 3.3 *Creating a Volume Group*

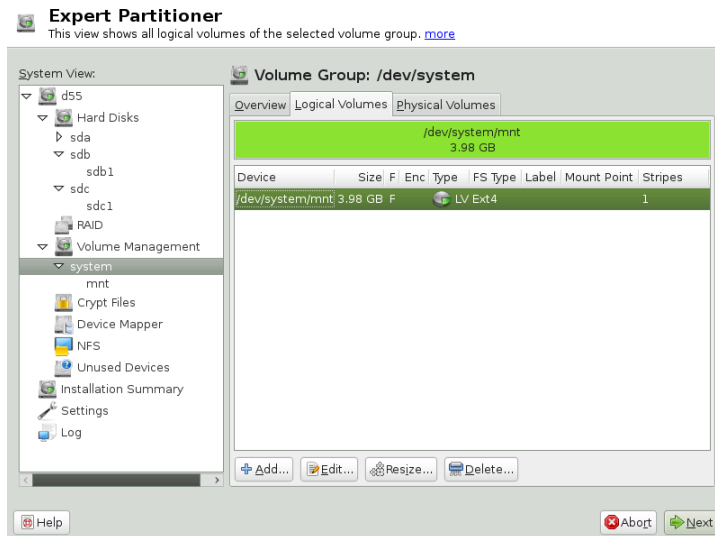


If you have multiple volume groups defined and want to add or remove PVs, select the volume group in the *Volume Management* list. Then change to the *Overview* tab and select *Resize*. In the following window, you can add or remove PVs to the selected volume group.

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 3.4 *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, be-

cause it does not provide data backup, but the name *RAID 0* for this type of system 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 2 and RAID 3

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

RAID 4

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

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.

Other RAID Levels

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

3.3.1 Soft RAID Configuration with YaST

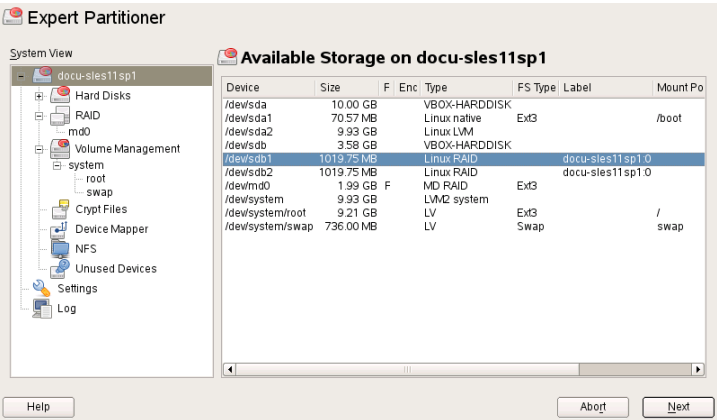
The YaST *RAID* configuration can be reached from the YaST Expert Partitioner, described in Section 3.1, “Using the YaST Partitioner” (page 67). This partitioning tool enables you to edit and delete existing partitions and create new ones to be used with soft RAID:

- 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 3.5 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 HOW-TOs at:

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

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

Part II. Managing and Updating Software

Installing or Removing Software

Use YaST's software management tool to search for software components you want to add or remove. YaST resolves all dependencies for you. To install packages not shipped with the installation media, add additional software repositories to your setup and let YaST manage them. Keep your system up-to-date by managing software updates with the update applet.

Change the software collection of your system with YaST Software Manager. This YaST module is available in three toolkit flavors: Qt (for KDE desktops), GTK+ (for GNOME desktops), and ncurses (providing a pseudo-graphical user interface in text mode). This chapter describes Qt and GTK+ flavors—for details on the ncurses YaST, see Chapter 11, *YaST in Text Mode* (page 173).

TIP: Changing the Toolkit Flavor

By default, YaST is started with the toolkit matching your desktop (GTK+ under GNOME, Qt under KDE). To alter this default setting system-wide, change the variable `WANTED_GUI` in `/etc/sysconfig/yast2` to either `qt` or `gtk`.

If you do not want to change the system-wide settings, you can nevertheless start YaST in the desired flavor from command line by using the `--gtk` or `--qt`. For example: `yast2 --gtk`.

NOTE: Confirmation and Review of Changes

When installing, updating or removing packages, any changes in the Software Manager are not applied immediately but only after confirming them

with *Accept* or *Apply* respectively. YaST maintains a list with all actions, allowing you to review and modify your changes before applying them to the system.

4.1 Definition of Terms

Repository

A local or remote directory containing packages, plus additional information about these packages (package meta-data).

(Repository) Alias

A short name for a repository used by various zypper commands. The alias can be chosen by the user when adding a repository and must be unique.

Product

Represents a whole product, for example openSUSE®.

Pattern

A pattern is an installable group of packages dedicated to a certain purpose. For example, the `Laptop` pattern contains all packages that are needed in a mobile computing environment. Patterns define package dependencies (such as required or recommended packages) and come with a preselection of packages marked for installation. This ensures that the most important packages needed for a certain purpose are available on your system after installation of the pattern. However, not necessarily all packages in a pattern are preselected for installation and you can manually select or deselect packages within a pattern according to your needs and wishes.

Package

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

Patch

A patch consists of one or more packages and may be applied by means of `deltarpm`s. It may also introduce dependencies to packages that are not installed yet.

Resolvable

An generic term for product, pattern, package or patch. The most commonly used type of resolvable is a package or a patch.

deltarpm

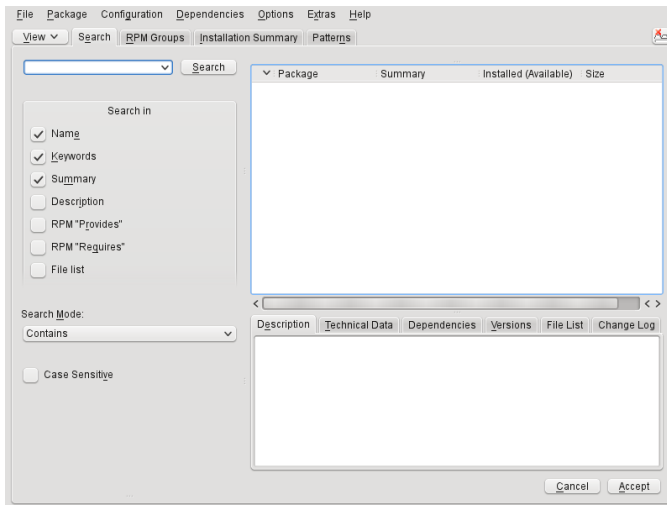
A deltarpm consists only of the binary diff between two defined versions of a package, and therefore has the smallest download size. Before being installed, the full RPM package is rebuilt on the local machine.

Package Dependencies

Certain packages are dependent on other packages, such as shared libraries. In other terms, a package may *require* other packages—if the required packages are not available, the package cannot be installed. In addition to dependencies (package requirements) that must be fulfilled, some packages *recommend* other packages. These recommended packages are only installed if they are actually available, otherwise they are just ignored and the package recommending them is installed nevertheless.

4.2 Using the KDE Interface (Qt)

The YaST Qt interface is started by default when using the desktops KDE, icewm, and others. It is also used when invoking YaST from a remote terminal. Start the software manager from the YaST Control Center by choosing *Software > Software Management*.



4.2.1 Views for Searching Packages or Patterns

The YaST software manager can install packages or patterns from all currently enabled repositories. It offers different views and filters to make it easier to find the software you are searching for. The *Search* view is the default view of the window. To change view, click *View* and select one of the following entries from the drop-down list. The selected view opens in a new tab.

Patterns

Lists all patterns available for installation on your system.

Package Groups

Lists all packages sorted by groups such as *Graphics*, *Programming*, or *Security*.

RPM Groups

Lists all packages sorted by functionality with groups and subgroups. For example *Networking* > *Email* > *Clients*.

Languages

Filter to list all packages needed to add a new system language.

Repositories

Filter to list packages by repository. In order to select more than one repository, hold the **key** while clicking on repository names. The “pseudo repository” *@System* lists all packages currently installed.

Search

Lets you search for a package according to certain criteria. Enter a search term and press **.** Refine your search by specifying where to *Search In* and by changing the *Search Mode*. For example, if you do not know the package name but only the name of the application that you are searching for, try including the package *Description* in the search process.

Installation Summary

If you have already selected packages for installation, update or removal, this view shows the changes that will be applied to your system as soon as you click *Accept*. To filter for packages with a certain status in this view, activate or deactivate the respective check boxes. Hit **+ F1** for details on the status flags.

TIP: Finding Packages Not Belonging to an Active Repository

To list all packages that do not belong to an active repository, choose *View > Repositories > @System* and then choose *Secondary Filter > Unmaintained Packages*. This is useful, for example, if you have deleted a repository and would like to make sure no packages from that repository remain installed.

4.2.2 Installing and Removing Packages or Patterns

Certain packages are dependent on other packages, such as shared libraries. On the other hand, some packages cannot coexist with others on the system. If possible, YaST automatically resolves these dependencies or conflicts. If your choice results in a dependency conflict that cannot be automatically solved, you need to solve it manually as described in Section 4.2.4, “Checking Software Dependencies” (page 94).

NOTE: Removal of Packages

When removing any packages, by default YaST only removes the selected packages. If you want YaST to also remove any other packages that become unneeded after removal of the specified package, select *Options > Cleanup when deleting packages*.

- 1 Search for packages as described in Section 4.2.1, “Views for Searching Packages or Patterns” (page 90).
- 2 The packages found are listed in the right pane. To install a package or remove it, right-click it and choose *Install* or *Delete*. If the relevant option is not available, check the package status indicated by the symbol in front of the package name—hit + F1 for help.

TIP: Applying an Action to All Packages Listed

To apply an action to all packages listed in the right pane, choose an action from *Package > All in This List*.

- 3 To install a pattern, right-click the pattern name and choose *Install*.
- 4 It is not possible to remove a pattern per se. Instead, select the packages of a pattern you want to remove and mark them for removal.
- 5 In order to select more packages, repeat the steps mentioned above.
- 6 Before applying your changes, you can review or modify them by clicking *View > Installation Summary*. By default, all packages that will change status, are listed.
- 7 In order to revert the status for a package, right-click the package and select one of the following entries: *Keep* if the package was scheduled to be deleted or updated, or *Do Not Install* if it was scheduled for installation. To abandon all changes and close the Software Manager, click *Cancel* and *Abandon*.
- 8 When you are finished, click *Accept* to apply your changes.
- 9 In case YaST found dependencies on other packages, a list of packages that have additionally been chosen for installation, update or removal is presented. Click *Continue* to accept them.

After all selected packages are installed, updated or removed, the YaST Software Manager automatically terminates.

NOTE: Installing Source Packages

Installing source packages with YaST Software Manager is not possible at the moment. Use the command line tool `zypper` for this purpose. For more information, see Section “Installing Source Packages” (page 128).

4.2.3 Updating Packages

Instead of updating individual packages, you can also update all installed packages or all packages from a certain repository. When mass updating packages, the following aspects are generally considered:

- priorities of the repositories that provide the package
- architecture of the package (for example, x86_64, i686, i586)
- version number of the package

- package vendor

Which of the aspects has the highest importance for choosing the update candidates depends on the respective update option you choose.

- 1 To update all installed packages to the latest version, choose *Package > All Packages > Update if Newer Version Available* from the main menu.

All repositories are checked for possible update candidates, using the following policy: YaST first tries to restrict the search to packages with the same architecture and vendor like the installed one. If the search is positive, the “best” update candidate from those is selected according to the process below. However, if no comparable package of the same vendor can be found, the search is expanded to all packages with the same architecture. If still no comparable package can be found, all packages are considered and the “best” update candidate is selected according to the following criteria:

1. Repository priority: Prefer the package from the repository with the highest priority.
2. If more than one package results from this selection, choose the one with the “best” architecture (best choice: matching the architecture of the installed one; otherwise: x86_64 > i686 > i586).

If the resulting package has a higher version number than the installed one, the installed package will be updated and replaced with the selected update candidate.

This option tries to avoid changes in architecture and vendor for the installed packages, but under certain circumstances, they are tolerated.

NOTE: Update Unconditionally

If you choose *Package > All Packages > Update Unconditionally* instead, basically the same criteria apply but candidate package found is installed unconditionally. Thus, choosing this option might actually lead to downgrading some packages.

- 2 To make sure that the packages for a mass update derive from a certain repository:
 - 2a Choose the repository from which to update as described in Section 4.2.1, “Views for Searching Packages or Patterns” (page 90) .

- 2b** On the right hand side of the window, click *Switch system packages to the versions in this repository*. This explicitly allows YaST to change the package vendor when replacing the packages.

As soon as you proceed with *Accept*, all installed packages will be replaced by packages deriving from this repository, if available. This may lead to changes in vendor and architecture and even to downgrading some packages.

- 2c** To refrain from this, click *Cancel switching system packages to the versions in this repository*. Note that you can only cancel this until you press the *Accept* button.

- 3** Before applying your changes, you can review or modify them by clicking *View > Installation Summary*. By default, all packages that will change status, are listed.
- 4** If all options are set according to your wishes, confirm your changes with *Accept* to start the mass update.

4.2.4 Checking Software Dependencies

Most packages are dependent on other packages. If a package, for example, uses a shared library, it is dependent on the package providing this library. On the other hand some packages cannot coexist with each other, causing a conflict (for example, you can only install one mail transfer agent: sendmail or postfix). When installing or removing software, the Software Manager makes sure no dependencies or conflicts remain unsolved to ensure system integrity.

In case there exists only one solution to resolve a dependency or a conflict, it is resolved automatically. Multiple solutions always cause a conflict which needs to be resolved manually. If solving a conflict involves a vendor or architecture change, it also needs to be solved manually. When clicking *Accept* to apply any changes in the Software Manager, you get an overview of all actions triggered by the automatic resolver which you need to confirm.

By default, dependencies are automatically checked. A check is performed every time you change a package status (for example, by marking a package for installation or removal). This is generally useful, but can become exhausting when manually resolving a dependency conflict. To disable this function, uncheck *Dependencies > Autocheck*. Manually perform a dependency check with *Dependencies > Check Now*.

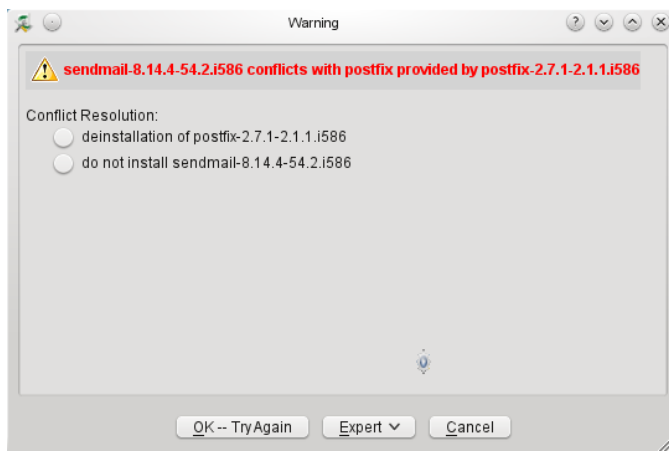
A consistency check is always performed when you confirm your selection with *Accept*.

To review a package's dependencies, right-click it and choose *Show Solver Information*. A map showing the dependencies opens. Packages that are already installed are displayed in a green frame.

NOTE: Manually Solving Package Conflicts

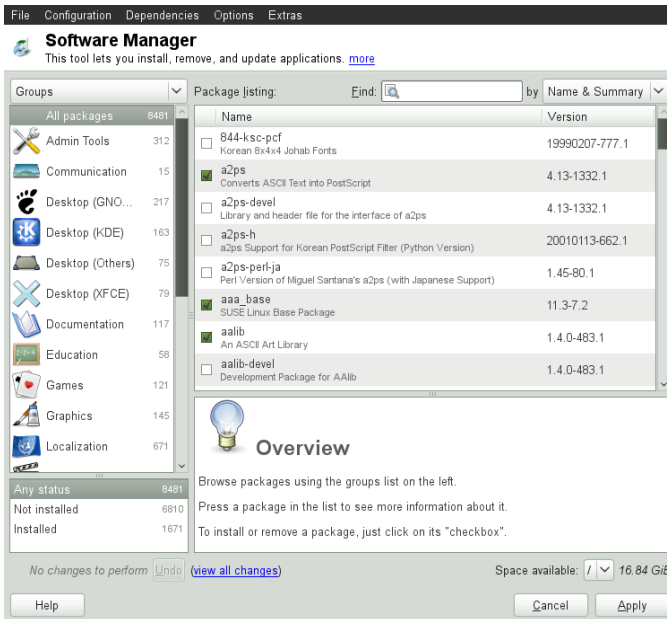
Unless you are very experienced, follow the suggestions YaST makes when handling package conflicts, otherwise you may not be able to resolve them. Keep in mind that every change you make, potentially triggers other conflicts, so you can easily end up with a steadily increasing number of conflicts. In case this happens, *Cancel* the Software Manager, *Abandon* all your changes and start again.

Figure 4.1 *Conflict Management of the Software Manager*



4.3 Using the GNOME Interface (GTK+)

The YaST GTK+ interface is started by default when using the desktops GNOME and XFCE. Start the software manager from the YaST Control Center by clicking *Software > Software Management*.



4.3.1 Views for Searching Packages or Patterns

The easiest way to find a package is to use the search field in the upper right corner of the software manager. Enter a search term and press . By default it will search package names and summaries. Press the search item to change this filter and search the file lists, for example.

The software manager also offers different views and filters for displaying package lists. These are available from the pull-down menu in the upper left corner:

Groups

The default view lists all packages sorted by groups such as *Admin Tools*, *Graphics*, *Programming*, or *Security*.

RPM Groups

Lists all packages sorted by functionality with groups and subgroups. For example *Networking > Email > Clients*.

Repositories

Filter to list packages by repository. In order to select more than one repository, hold the **key** while clicking on repository names. The “pseudo repository” *@System* lists all packages currently installed.

To add, edit, or remove available repositories, click *Edit Repositories*.

Patterns

Lists all patterns available for installation on your system.

Languages

Filter to list all packages needed to add a new system language.

The box in the lower right corner of the dialog also allows to filter for packages that are *Installed*, *Not Installed* or *Upgradable*. If you select the *No Status* entry, all available packages from the configured repositories are displayed, independent of their status.

4.3.2 Installing and Removing Packages or Patterns

Certain packages are dependent on other packages, such as shared libraries. On the other hand, some packages cannot coexist with others on the system. If possible, YaST automatically resolves these dependencies or conflicts. If your choice results in a dependency conflict that cannot be automatically solved, you need to solve it manually as described in Section 4.2.4, “Checking Software Dependencies” (page 94).

- 1 Search for packages as described in Section 4.3.1, “Views for Searching Packages or Patterns” (page 96).
- 2 The packages found are listed in the right pane. To further filter the search results according to package status (*Any Status*, *Not Installed*, *Installed*, *Upgradable*), select one of the entries in the box at the lower left corner of the dialog. For details about a package, click the package in the list. Information like available versions, authors and changelog of the package are displayed in the lower right corner of the window.

To mark a package for installation, re-installation, removal, or upgrade, right-click the package and choose the appropriate action from the menu.

TIP: Applying an Action to All Packages Listed

To apply an action to all packages listed in the right pane, right-click a package, choose *Select All*, right-click again and choose an action.

- 3 To install a pattern, right-click the pattern name and choose *Install*.
- 4 It is not possible to remove a pattern per se. Instead, select the packages of a pattern you want to remove and mark them for removal.
- 5 In order to select more packages, repeat the steps mentioned above.
- 6 Before applying your changes, you can review or modify them by clicking *View All Changes* at the bottom of the dialog. By default, all packages that will change status are listed.
- 7 To revert changes for a package, click the *Undo* icon with the yellow arrow. To finish the review, click *Close*.
- 8 When you are finished with the selection of packages to install or remove, *Apply* your changes.
- 9 In case YaST found dependencies on other packages, a list of packages that have additionally been chosen for installation, update or removal is presented. Click *Apply* to accept them.

After all selected packages are installed, updated or removed, the YaST Software Manager automatically terminates.

NOTE: Installing Source Packages

Installing source packages with YaST Software Manager is not possible at the moment. Use the command line tool `zypper` for this purpose. For more information, see Section “Installing Source Packages” (page 128).

4.3.3 Updating Packages

Instead of updating individual packages, you can also update all installed packages or all packages from a certain repository. When mass updating packages, the following aspects are generally considered:

- priorities of the repositories that provide the package
- architecture of the package (for example, x86_64, i686, i586)
- version number of the package
- package vendor

Which of the aspects has the highest importance for choosing the update candidates depends on the respective update option you choose.

- 1** To view the list of packages that can be updated (packages with higher versions are available), select *Upgradable* in the bottom left box.
- 2** To update all packages listed there, click *Upgrade All*.

To install only upgradable packages for which an official patch has been issued, click *Upgrade Patches*. Those packages are marked by a patch tag next to their version number. Choosing this option is equivalent to doing an online update with YaST as described in Chapter 5, *YaST Online Update* (page 111).

If no patches have been issued since last applying patches, the button is disabled.

- 3** To make sure that the packages for a mass update derive from a certain repository:
 - 3a** Choose the repository from which to update as described in Section 4.3.1, “Views for Searching Packages or Patterns” (page 96) .
 - 3b** On the right hand side of the window, click *Switch system packages to the versions in this repository*. This explicitly allows YaST to change the package vendor when replacing the packages.

All installed packages will be replaced by packages deriving from this repository, if available. This may lead to changes in vendor and architecture and even to downgrading some packages.

- 4** Before applying the changes, you can review or modify them by clicking *View All Changes* at the bottom of the dialog. By default, all packages that will change status are listed.
- 5** To refrain from switching the system packages to the versions in this repository, click the *Undo* button next to the respective option.

- 6 If all options are set according to your wishes, confirm your changes with *Apply* to start the mass update.

4.3.4 Checking Software Dependencies

Most packages are dependent on other packages. If a package, for example, uses a shared library, it will be dependent on the package providing this library. On the other hand, some packages cannot coexist with each other, causing a conflict (for example, you can only install one mail transfer agent: sendmail or postfix). When installing or removing software, the Software Manager makes sure no dependencies or conflicts remain unresolved to ensure system integrity.

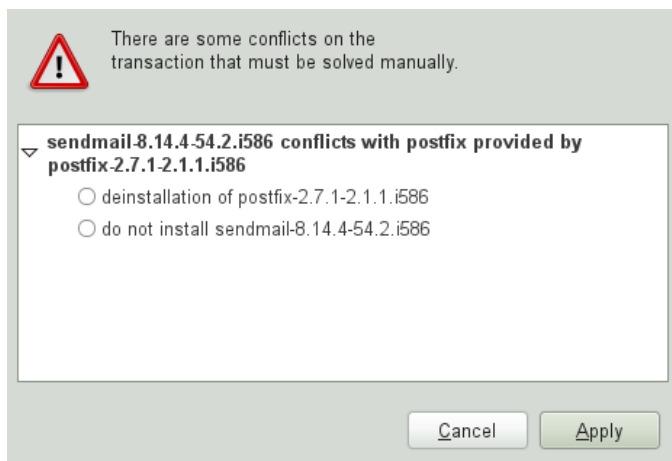
In case there exists only one solution to resolve a dependency or a conflict, it is resolved automatically. Multiple solutions always cause a conflict which needs to be resolved manually. If solving a conflict involves a vendor or architecture change, it also needs to be solved manually. When clicking *Apply* to apply any changes in the Software Manager, you get an overview of all actions triggered by the automatic resolver which you need to confirm.

By default, dependencies are automatically checked. A check is performed every time you change a package status (for example, by marking a package for installation or removal). This is generally useful, but can become exhausting when manually resolving a dependency conflict. To disable this function, uncheck *Dependencies > Autocheck*. Manually perform a dependency check with *Dependencies > Check Now*. A consistency check is always performed when you confirm your selection with *Apply*.

NOTE: Manually Solving Package Conflicts

Unless you are very experienced, follow the suggestions YaST makes when handling package conflicts, otherwise you may not be able to resolve them. Keep in mind that every change you make, potentially triggers other conflicts, so you can easily end up with a steadily increasing number of conflicts. In case this happens, click *Cancel* and *Quit* the software manager. Relaunch it to start again.

Figure 4.2 *Conflict Management of the Software Manager*



4.4 Managing Software Repositories and Services

If you want to install third-party software, add additional software repositories to your system. By default, the product repositories such as openSUSE-DVD 11.4 and a matching update repository are automatically configured during the installation. Depending on the initially selected product, a separate language add-on repository with translations, dictionaries, etc. might also be configured.

To manage repositories, start YaST and select *Software > Software Repositories*. The *Configured Software Repositories* dialog opens. Here, you can also manage subscriptions to so-called *Services* by changing the *View* at the right corner of the dialog to *All Services*. A Service in this context is a *Repository Index Service* (RIS) that can offer one or more software repositories. Such a Service can be changed dynamically by its administrator or vendor.

Each repository provides files describing content of the repository (package names, versions, etc.). These repository description files are downloaded to a local cache that is used by YaST. To ensure their integrity, software repositories can be signed with the GPG Key of the repository maintainer. Whenever you add a new repository, YaST offers the ability to import its key.

WARNING: Trusting External Software Sources

Before adding external software repositories to your list of repositories, make sure this repository can be trusted. openSUSE is not responsible for any potential problems arising from software installed from third-party software repositories.

4.4.1 Adding Software Repositories

You can either add repositories from a local harddisk, from a removable medium (like a CD, DVD or a USB mass storage) or from a network.

To add repositories from the *Configured Software Repositories* dialog in YaST proceed as follows:

- 1 Click *Add*.
- 2 From the list of available *Media Types* specify the type matching your repository:

For network sources, it is usually sufficient to use the default option, *Specify URL*.

To add a repository from a removable medium or a local harddisk, choose the relevant option and insert the medium or connect the USB device to the machine, respectively.
- 3 You can choose to *Download Repository Description Files* now. If the option is unchecked, YaST will automatically download the files later, if needed. Click *Next* to proceed.
- 4 When adding a repository from the network, enter the data you are prompted for. Continue with *Next*.
- 5 Depending on the repository you have added, you might be asked if you want to import the GPG key with which is signed or asked to agree to a license.

After confirming these messages, YaST will download and parse the metadata and add the repository to the list of *Configured Repositories*..

- 6 If needed, adjust the repository *Properties* as described in Section 4.4.2, “Managing Repository Properties” (page 103) or confirm your changes with *OK* to close the configuration dialog.

Now you can install software from this repository as described in Section 4.2, “Using the KDE Interface (Qt)” (page 89) or in Section 4.3, “Using the GNOME Interface (GTK+)” (page 95).

YaST also offers a list of predefined popular repositories available under *Community Repositories*. Among others, it includes driver repositories for nVidia and ATI graphics cards, and popular projects from the openSUSE® Build Service, such as the Mozilla repository (containing packages with the most recent versions of Firefox and Thunderbird).

Procedure 4.1 *Configuring Community Repositories*

- 1** Start YaST and select *Software > Software Repositories*.
- 2** Click *Add*.
- 3** Choose *Community Repositories* and proceed with *Next*.
- 4** From the preconfigured list of repositories choose the ones you want to add by ticking the respective check boxes. The Mozilla repository, for example, is listed as *openSUSE BuildService - Mozilla*.

Confirm with *OK*.
- 5** Accept to *Import* the GnuPG key. You need to import a key for each repository you have chosen.
- 6** The new software repositories are now listed in the *Configured Software Repositories* overview. Click *OK* to leave the software repositories configuration.

4.4.2 Managing Repository Properties

The *Configured Software Repositories* overview of the *Software Repositories* lets you change the following repository properties:

Status

The repository status can either be *Enabled* or *Disabled*. You can only install packages from repositories that are enabled. To turn a repository off temporarily click *Disable*. You can also double-click on a repository name to toggle its status. If you want to remove a repository completely, click *Delete*.

Refresh

When refreshing a repository, its content description (package names, versions, etc.) is downloaded to a local cache that is used by YaST. It is sufficient to do this once for static repositories such as CDs or DVDs, whereas repositories whose content changes often should be refreshed frequently. Enable The easiest way to keep a repository's cache up-to-date is to choose *Automatically Refresh*. To do a manual refresh click *Refresh* and select one of the options.

Keep Downloaded Packages

Packages from remote repositories are downloaded before being installed. By default, they are deleted upon a successful installation. Activating *Keep Downloaded Packages* prevents the deletion of downloaded packages. The download location is configured in `/etc/zypp/zypp.conf`, by default it is `/var/cache/zypp/packages`.

Priority

The *Priority* of a repository is a value between 1 and 200, with 1 being the highest priority and 200 the lowest priority. Any new repositories that are added with YaST get a priority of 99 by default. If you do not care about a priority value for a certain repository, you can also set the value to 0 to apply the default priority to that repository (99). If a package is available in more than one repository, then the repository with the highest priority takes precedence. This is useful if you want to avoid downloading packages unnecessarily from the Internet by giving a local repository (for example, a DVD) a higher priority.

IMPORTANT: Priority vs. Version

The repository with the highest priority takes precedence in any case. Therefore, make sure that the update repository always has the highest priority (20 by default), otherwise you might install an outdated version that will not be updated until the next online update.

If you add repositories providing new versions for programs shipped with openSUSE (for example a repository with the latest KDE or GNOME version), make sure they have a higher priority than the standard repositories, otherwise packages from these repositories will not be installed by default.

Name and URL

To change a repository name or its URL, select it from the list with a single-click and then click *Edit*.

4.4.3 Managing Repository Keys

To ensure their integrity, software repositories can be signed with the GPG Key of the repository maintainer. Whenever you add a new repository, YaST offers to import its key. Verify it as you would do with any other GPG key and make sure it does not change. If you detect a key change, something might be wrong with the repository. Disable the repository as an installation source until you know the cause of the key change.

To manage all imported keys, click *GPG Keys...* in the *Configured Software Repositories* dialog. Select an entry with the mouse to show the key properties at the bottom of the window. *Add*, *Edit* or *Delete* keys with a click on the respective buttons.

4.5 Keeping the System Up-to-date

openSUSE offers a continuous stream of software security patches for your product. The update applet informs you about the availability of patches and lets you easily install them with just a few clicks.

4.5.1 Using the KDE Software Updater

The Software Updater icon resides in the system tray of your panel depicting a gear-wheel with a green arrow. To start Software Updater manually, choose *System Settings > SoftwareManagement > Software Updates* from the main menu. Alternatively, press + F2 and enter `kpk_update`.

NOTE: Icon visibility

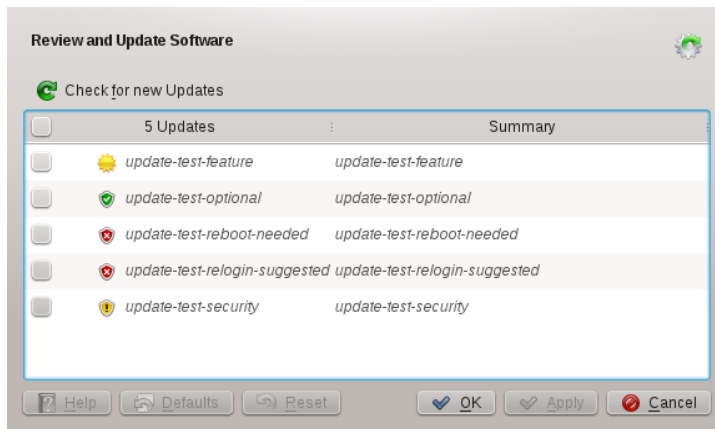
The Software Updater icon is only visible in the system tray, if patches are available. Hover over the icon to see the number of patches available.

Installing Patches

- 1 Whenever software updates are available, the applet icon appears in the panel. Left-click the Software Updater icon to launch the *Review and Update* software window.

- 2 Select a patch for installation by ticking its checkbox. Get detailed information on a patch by clicking on its title. To select all available patches for installation, tick the checkbox in the table header.
- 3 Click *Apply* to start the patch installation.
- 4 In case you have started the patch installation for the first time, you will be asked to enter the `root` password twice in order to proceed. If you also check *Remember authorization* you will never be asked again to provide the password.
- 5 The *Additional Changes* window showing an installation summary opens. Click *Continue* to finish the installation.

Figure 4.3 KDE Software Updater



The YaST Online Update offers advanced features to customize the patch installation. Please refer to Chapter 5, *YaST Online Update* (page 111) for more information.

Configuring the KDE Software Updater

By default Software Updater checks for updates every 24 hours, notifies you when patches are available and does not automatically install patches. These settings can be changed with the *Software Management settings*. To open the *Software Management settings* choose *System Settings > Software Management > Settings* from the main menu. Alternatively, press `+ F2` and enter `kpk_settings`. The settings for Software Updater are available in the *Update Settings* section.

IMPORTANT: Patch Origin

The *Software Management settings* also allows you to configure the repositories (*Origin of Packages*) to be used. This setting not only applies to Software Updater but also to the KDE Software Management module (*Get and Remove Software*).

Make sure the repository *Updates for openSUSE 11.4* is always selected—otherwise you will not receive patches.

4.5.2 Using the GNOME Update Applet

The update applet resides in the notification area of the panel. Its icon changes depending on the availability and relevance of patches and the status of the update. To invoke the applet manually, choose *Computer > More Applications > System > Software Update*.

NOTE: Icon visibility

By default, the update applet icon is only visible in the notification area, if patches are available.

Open box with a globe

The update is busy (for example checking for updates or installing software).

Red Star with Exclamation Mark

Security patches are available.

Orange Star with an Up Arrow

Important patches are available.

Yellow Star with a Down Arrow

Trivial patches are available.

Yellow Triangle with Exclamation Mark

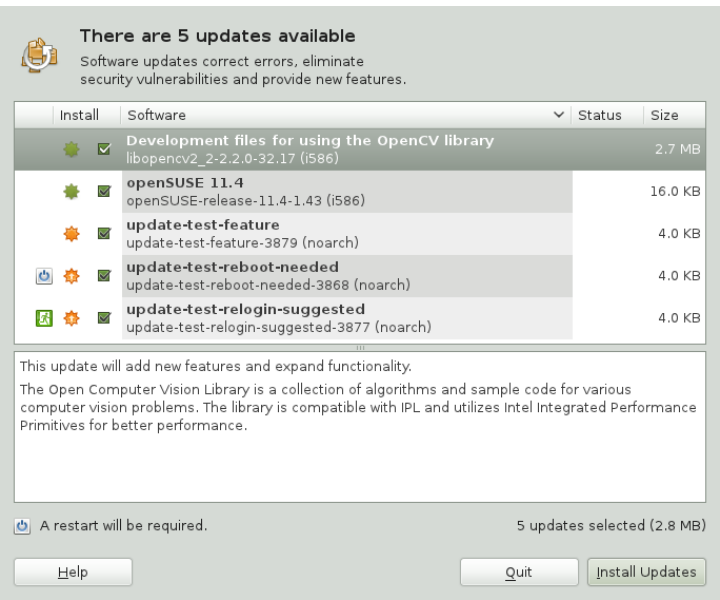
An error has occurred.

Installing Patches

Procedure 4.2 Installing Patches

- 1 Whenever new patches are available, a notification message will appear and the Update Applet icon will be visible in the notification area. Either click on *Install updates* in the notification message or click on the to open the *Software Update* window.
- 2 All security and important patches are preselected. It is strongly recommended to install these patches. Trivial patches can be manually selected by ticking the respective check boxes. Get detailed information on a patch by clicking on its title.
- 3 Click *Install Updates* to start the patch installation.
- 4 The *Additional Confirmation Required* window showing an installation summary opens. Click *Continue* to proceed.
- 5 Enter the `root` password in the authentication screen and proceed with *Authenticate*.

Figure 4.4 GNOME Update Applet



The YaST Online Update offers advanced features to customize the patch installation. Please refer to Chapter 5, *YaST Online Update* (page 111) for more information.

Configuring the Software Update Applet

To configure the update applet, right-click the update icon in the panel and choose *Preferences*. The configuration dialog lets you modify the following settings:

Check for Updates

Choose how often a check for updates is performed: *Hourly*, *Daily*, *Weekly*, or *Never*.

Automatically Install

Configure whether patches are installed automatically or not (default). Automatic installation can be chosen for either security patches only or for all patches.

Check for Major Upgrades

Choose how often a check for major upgrades is performed: *Daily*, *Weekly*, or *Never*.

Check for updates when using mobile broadband

This configuration option is only available on mobile computers. Turned off by default.

More options are configurable using `gconf-editor`: `apps > gnome-packagekit`.

YaST Online Update

openSUSE offers a continuous stream of software security updates for your product. By default, the update applet is used to keep your system up-to-date. Refer to Section 4.5, “Keeping the System Up-to-date” (page 105) for further information on the update applet. This chapter covers the alternative tool for updating software packages: YaST Online Update.

The current patches for openSUSE® are available from an update software repository, which is automatically configured during the installation. Alternatively, you can manually add an update repository from a source you trust. To add or remove repositories, start the Repository Manager with *Software > Software Repositories* in YaST. Learn more about the Repository Manager in Section 4.4, “Managing Software Repositories and Services” (page 101).

openSUSE provides updates with different relevance levels:

Security Updates

Fix severe security hazards and should definitely be installed.

Recommended Updates

Fix issues that could compromise your computer.

Optional Updates

Fix non-security relevant issues or provide enhancements.

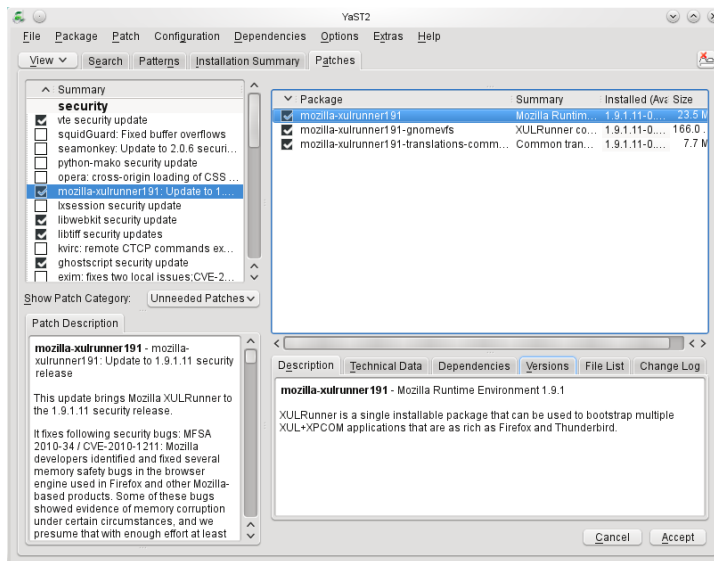
5.1 The Online Update Dialog

The YaST *Online Update* dialog is available in two toolkit flavors: GTK (for GNOME) and Qt (for KDE). Both interfaces differ in look and feel but basically provide the same functions. The following sections provide a brief description of each. To open the dialog, start YaST and select *Software > Online Update*. Alternatively, start it from the command line with `yast2 online_update`.

5.1.1 KDE Interface (Qt)

The *Online Update* window consists of four sections.

Figure 5.1 YaST *Online Update*—Qt Interface



The *Summary* section on the left lists the available patches for openSUSE. The patches are sorted by security relevance: security, recommended, and optional. You can change the view of the *Summary* section by selecting one of the following options from *Show Patch Category*:

Needed Patches (default view)

Non-installed patches that apply to packages installed on your system.

Unneeded Patches

Patches that either apply to packages not installed on your system, or patches that have requirements which have already been fulfilled (because the relevant packages have already been updated from another source).

All Patches

All patches available for openSUSE.

Each list entry in the *Summary* section consists of a symbol and the patch name. For an overview of the possible symbols and their meaning, press + F1. Actions required by *Security* and *Recommended* patches are automatically preset. These actions are *Autoinstall*, *Autoupdate* and *Autodelete*.

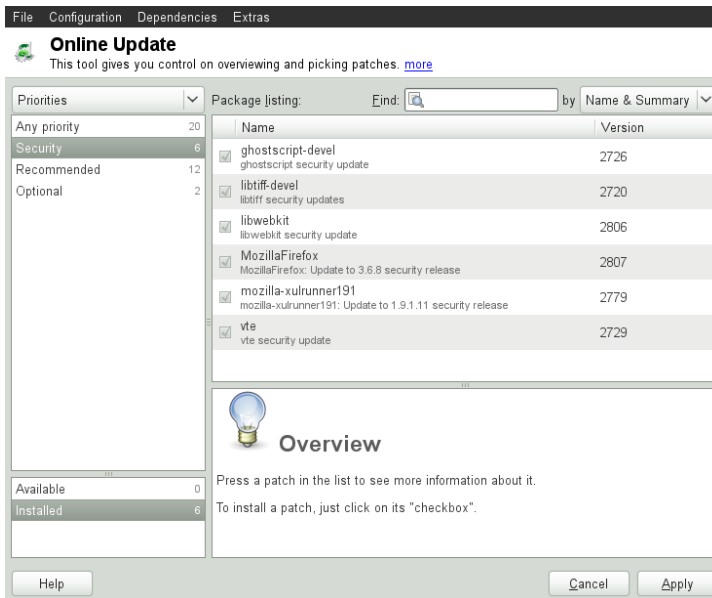
If you install an up-to-date package from a repository other than the update repository, the requirements of a patch for this package may be fulfilled with this installation. In this case a check mark is displayed in front of the patch summary. The patch will be visible in the list until you mark it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

Select an entry in the *Summary* section to view a short *Patch Description* at the bottom left corner of the dialog. The upper right section lists the packages included in the selected patch (a patch can consist of several packages). Click an entry in the upper right section to view details about the respective package that is included in the patch.

5.1.2 GNOME Interface (GTK)

The *Online Update* window consists of four main sections.

Figure 5.2 YaST Online Update—GTK Interface



The upper right section lists the available (or already installed) patches for openSUSE. To filter patches according to their security relevance, click the corresponding *Priority* entry in the upper right section of the window: *Security*, *Recommended*, *Optional* or *Any priority*.

If all available patches are already installed, the *Package listing* in the upper right section will show no entries. The box in the bottom left-hand section shows the number of both available and already installed patches and lets you toggle the view to either *Available* or *Installed* patches.

Select an entry in the *Package listing* section to view a patch description and further details at the bottom right corner of the dialog. As a patch can consist of several packages, click the *Applies to* entry in the lower right section to see which packages are included in the respective patch.

Click on a patch entry to open a row with detailed information about the patch in the bottom of the window. Here you can see a detailed patch description as well as the versions available. You can also choose to *Install* optional patches—security and recommended patches are already preselected for installation.

5.2 Installing Patches

The YaST Online Update dialog allows you to either install all available patches at one go or to manually select the patches that you want to apply to your system. You may also revert patches that have been applied to the system.

By default, all new patches (except the `optional` ones) that are currently available for your system are already marked for installation. They will be applied automatically once you click *Accept* or *Apply*.

Procedure 5.1 *Applying Patches with YaST Online Update*

- 1** Start YaST and select *Software > Online Update*.
- 2** To automatically apply all new patches (except the `optional` ones) that are currently available for your system, proceed with *Apply* or *Accept* to start the installation of the preselected patches.
- 3** To first modify the selection of patches that you want to apply:
 - 3a** Use the respective filters and views the GTK and Qt interfaces provide. For details, refer to Section 5.1.1, “KDE Interface (Qt)” (page 112) and Section 5.1.2, “GNOME Interface (GTK)” (page 113).
 - 3b** Select or deselect patches according to your needs and wishes by activating or deactivating the respective checkbox (GNOME) or by right-clicking the patch and choosing the respective action from the context menu (KDE).

IMPORTANT: Always Apply Security Updates

However, do not deselect any `security`-related patches if you do not have a very good reason for doing so. They fix severe security hazards and prevent your system from exploits.

- 3c** Most patches include updates for several packages. If you want to change actions for single packages, right-click a package in the package view and choose an action (KDE).
- 3d** To confirm your selection and to apply the selected patches, proceed with *Apply* or *Accept*.

- 4 After the installation is complete, click *Finish* to leave the YaST *Online Update*. Your system is now up-to-date.

TIP: Disabling deltarpm

By default updates are downloaded as deltarpm. Since rebuilding rpm packages from deltarpm is a memory and CPU time consuming task, certain setups or hardware configurations might require you to disable the usage of deltarpm for performance sake.

To disable the use of deltarpm edit the file `/etc/zypp/zypp.conf` and set `download.use_deltarpm` to `false`.

5.3 Automatic Online Update

YaST also offers the possibility to set up an automatic update with daily, weekly or monthly schedule. To use the respective module, you need to install the `yast2-online-update-configuration` package first.

Procedure 5.2 *Configuring the Automatic Online Update*

- 1 After installation, start YaST and select *Software > Online Update Configuration*.

Alternatively, start the module with
`yast2 online_update_configuration` from the command line.

- 2 Activate *Automatic Online Update*.
- 3 Choose whether to update *Daily*, *Weekly*, or *Monthly*.

Some patches, such as kernel updates or packages requiring license agreements, require user interaction, which would cause the automatic update procedure to stop.

- 4 To automatically accept any license agreements, activate *Agree with Licenses*.
- 5 Select if you also want to *Skip Interactive Patches* in case you want the update procedure to proceed fully automatically.

IMPORTANT: Skipping Patches

If you select to skip any packages that require interaction, run a manual *Online Update* from time to time in order to install those patches, too. Otherwise you might miss important patches.

- 6 Confirm your configuration with *OK*.

Installing Packages From the Internet

By default, it is only possible to install packages from configured and enabled repositories. Apart from the official repositories that are configured during the installation, numerous other repositories exist. The openSUSE® Build Service hosts several hundred ones and a lot of third party repositories exist, too—see http://en.opensuse.org/Additional_package_repositories.

openSUSE offers two easy ways to install from these repositories without the need to first subscribe to them. The `1-Click Install` method allows you to install packages directly from a Web browser, while the YaST Package Search lets you query almost all known repositories for openSUSE. You can directly install any package found by the package search module.

WARNING: Trusting External Software Sources

Before installing from any external software repository, make sure it can be trusted. openSUSE is not responsible for any potential problems arising from software installed from third-party software repositories.

6.1 1-Click Install

Installation using `1-Click Install` is available from a lot of repositories available on Web interfaces. A very popular interface is the openSUSE Build Service software search.

Procedure 6.1 *Installing Packages from the openSUSE Build Service via 1-Click Install*

- 1 Start the openSUSE Build Service search interface at <http://software.opensuse.org/search> .
- 2 Select your system version from the drop-down menu, for example openSUSE 11.4.
- 3 Enter the name of the package you want to install, for example the OpenStreetMap editor `josc`.
- 4 To refine the search, adjust the *Search Options* according to your wishes.
- 5 Click *Search*.
- 6 From the results list select the preferred item by clicking its *1-Click Install* button.
- 7 In the Web browser's download dialog, select to open the file with the *YaST Meta Package Handler*.

The 1-click installer opens the *Additional Software Repositories* dialog. It shows the repositories providing the packages you want to install per *1-Click Install*. They are activated per default. To proceed with the installation, keep the repository selection. By default, you remain subscribed to these repositories after the installation has finished and you will receive updates from them in the future.

- 8 If you want to use the new repositories only once instead, uncheck *Remain Subscribed to These Repositories after Installation*. Click *Next* to proceed.
- 9 Now select the software packages that should be installed. Normally there is no need to change the default selection. Click *Next* to proceed.
- 10 The *Proposal* screen summarizes the choices you made. Click *Customize* to restart the configuration steps from above. Click *Next* and *Yes* to proceed with the installation.
- 11 Enter the `root` password to start the installation. In case a new repository has been added you also need to confirm the import of the repository's GnuPG key.

During the installation several progress pop-ups appear that do not need any interaction. After reading the “Installation was successful” message, click *Finish*.

TIP: Disabling 1-Click Install Feature

If you want to disable the 1-Click install feature, uninstall the `yast2-metapackage-handler` package using YaST. Alternatively, enter the following command as `root`:

```
rpm -e yast2-metapackage-handler
```

6.2 YaST Package Search

Provided you are connected to the internet, you can also search and install packages from almost all known repositories for openSUSE directly via YaST's Package Search. This module is a YaST frontend for the Webpin package search available at <http://packages.opensuse-community.org/>. The YaST module is not available by default—you need to install the package `yast2-packager-webpin` first.

Procedure 6.2 *Installing Packages Using the YaST Package Search*

- 1 Start the Package Search from the YaST Control Center via *Software > Package Search*.
- 2 Search for a software package by entering its name into the *Search Expression* field and clicking *Search*.
- 3 The search result is listed on the *Found Packages* tab. Click on a package name to see the repository URL, the package version number and the architecture in the *Package Description* pane.

WARNING: Double Check the Package Information

Make sure to double check whether the software is hosted on a repository you trust before you install it. Also check if the architecture complies with your system (x86_64 packages can only be installed on 64bit systems).

- 4 Mark a package for installation by activating its checkbox. You can mark several packages at once. You can even start a new search for other packages without losing your current selection, which is always available on the *All Selected Packages*. Once you have finished the package selection, proceed with *Next*.

The *Additional Software Repositories* dialog shows the repositories providing the packages you want to install. They are activated per default. To proceed with the installation, keep the repository selection. By default, you remain subscribed to these repositories after the installation has finished, so you will receive updates from them in the future.

- 5 If instead you want to use the new repositories only once, uncheck *Remain Subscribed to These Repositories after Installation*. Click *Next* to proceed.
- 6 Now select the software packages that should be installed. Normally there is no need to change the default selection. Click *Next* to proceed.
- 7 The *Proposal* screen summarizes the choices you made. Click *Customize* to restart the configuration steps from above. Click *Next* and *Yes* to proceed with the installation.
- 8 Confirm the next dialog. In case a new repository is used, you will also need to confirm the import of the repository's GnuPG key. During the installation several progress pop-ups appear that do not need any interaction. After reading the “Installation was successful” message, click *Finish*.

Installing Add-On Products

Add-on products are system extensions. You can install a third party add-on product or a special system extension of openSUSE® (for example, a CD with support for additional languages or a CD with binary drivers). To install a new add-on, start YaST and select *Software > Add-On Products*. You can select various types of product media, like CD, FTP, USB mass storage devices (such as USB flash drives or disks) or a local directory. You can work also directly with ISO files. To add an add-on as ISO file media, select *Local ISO Image* then enter the *Path to ISO Image*. The *Repository Name* is arbitrary.

7.1 Add-Ons

To install a new add-on, proceed as follows:

- 1 In YaST select *Software > Add-On Products* to see an overview of already installed add-on products.
- 2 To install a new add-on product, click *Add*.
- 3 From the list of available *Media Types* specify the type matching your repository.
- 4 To add a repository from a removable medium, choose the relevant option and insert the medium or connect the USB device to the machine, respectively.
- 5 You can choose to *Download Repository Description Files* now. If the option is unchecked, YaST will automatically download the files later, if needed. Click *Next* to proceed.

6 When adding a repository from the network, enter the data you are prompted for. Continue with *Next*.

7 Depending on the repository you have added, you might be asked if you want to import the GPG key with which is signed or asked to agree to a license.

After confirming these messages, YaST will download and parse the metadata and add the repository to the list of *Configured Repositories*.

8 If needed, adjust the repository *Properties* as described in Section 4.4.2, “Managing Repository Properties” (page 103) or confirm your changes with *OK* to close the configuration dialog.

9 After having successfully added the repository for the add-on media, the software manager starts and you can install packages. Refer to Chapter 4, *Installing or Removing Software* (page 87) for details.

7.2 Binary Drivers

Some hardware needs binary-only drivers to function properly. If you have such hardware, refer to the release notes for more information about availability of binary drivers for your system. To read the release notes, open YaST and select *Miscellaneous > Release Notes*.

Managing Software with Command Line Tools

This chapter describes Zypper and RPM, two command line tools for managing software. For a definition of the terminology used in this context (for example, `repository`, `patch`, or `update`) refer to Section 4.1, “Definition of Terms” (page 88).

8.1 Using Zypper

Zypper is a command line package manager for installing, updating and removing packages as well as for managing repositories. It is especially useful for accomplishing remote software management tasks or managing software from shell scripts.

8.1.1 General Usage

The general syntax of Zypper is:

```
zypper [global-options] command [command-options] [arguments] ...
```

The components enclosed in brackets are not required. The simplest way to execute Zypper is to type its name, followed by a command. For example, to apply all needed patches to the system type:

```
zypper patch
```

Additionally, you can choose from one or more global options by typing them just before the command. For example, `--non-interactive` means running the command without asking anything (automatically applying the default answers):

```
zypper --non-interactive patch
```

To use the options specific to a particular command, type them right after the command. For example, `--auto-agree-with-licenses` means applying all needed patches to the system without asking to confirm any licenses (they will automatically be accepted):

```
zypper patch --auto-agree-with-licenses
```

Some commands require one or more arguments. When using the `install` command, for example, you need to specify which package(s) to install:

```
zypper install mplayer
```

Some options also require an argument. The following command will list all known patterns:

```
zypper search -t pattern
```

You can combine all of the above. For example, the following command will install the `mplayer` and `amarok` packages from the `factory` repository while being verbose:

```
zypper -v install --from factory mplayer amarok
```

The `--from` option makes sure to keep all repositories enabled (for solving any dependencies) while requesting the package from the specified repository.

Most Zypper commands have a `dry-run` option that does a simulation of the given command. It can be used for test purposes.

```
zypper remove --dry-run MozillaFirefox
```

8.1.2 Installing and Removing Software with Zypper

To install or remove packages use the following commands:

```
zypper install package_name  
zypper remove package_name
```

Zypper knows various ways to address packages for the `install` and `remove` commands:

by the exact package name (and version number)

```
zypper install MozillaFirefox
```


or

```
zypper install MozillaFirefox-3.5.3
```

by repository alias and package name

```
zypper install mozilla:MozillaFirefox
```

Where `mozilla` is the alias of the repository from which to install.

by package name using wildcards

The following command will install all packages that have names starting with “Moz”. Use with care, especially when removing packages.

```
zypper install Moz*
```

by capability

For example, if you would like to install a perl module without knowing the name of the package, capabilities come in handy:

```
zypper install 'perl(Time::ParseDate)'
```

by capability and/or architecture and/or version

Together with a capability you can specify an architecture (such as `i586` or `x86_64`) and/or a version. The version must be preceded by an operator: `<` (lesser than), `<=` (lesser than or equal), `=` (equal), `>=` (greater than or equal), `>` (greater than).

```
zypper install 'firefox.x86_64'
zypper install 'firefox>=3.5.3'
zypper install 'firefox.x86_64>=3.5.3'
```

by path to the RPM file

You can also specify a local or remote path to a package:

```
zypper install /tmp/install/MozillaFirefox.rpm
zypper install http://download.opensuse.org/repositories/mozilla/
SUSE_Factory/x86_64/MozillaFirefox-3.5.3-1.3.x86_64.rpm
```

To install and remove packages simultaneously use the `+/-` modifiers. To install `emacs` and remove `vim` simultaneously, use:

```
zypper install emacs -vim
```

To remove `emacs` and install `vim` simultaneously, use:

```
zypper remove emacs +vim
```

To prevent the package name starting with the `-` being interpreted as a command option, always use it as the second argument. If this is not possible, precede it with `--`:

```
zypper install -emacs +vim      # Wrong
zypper install vim -emacs       # Correct
zypper install -- -emacs +vim   # same as above
zypper remove emacs +vim        # same as above
```

If (together with a certain package) you automatically want to remove any packages that become unneeded after removing the specified package, use the `--clean-deps` option:

```
rm package_name --clean-deps
```

By default, Zypper asks for a confirmation before installing or removing a selected package, or when a problem occurs. You can override this behavior using the `--non-interactive` option. This option must be given before the actual command (install, remove, and patch) as in the following:

```
zypper --non-interactive install package_name
```

This option allows the use of Zypper in scripts and cron jobs.

WARNING: Do not Remove Mandatory System Packages

Do not remove packages such as `glibc`, `zypper`, `kernel`, or similar packages. These packages are mandatory for the system and, if removed, may cause the system to become unstable or stop working altogether.

Installing Source Packages

If you want to install the corresponding source package of a package, use:

```
zypper source-install package_name
```

That command will also install the build dependencies of the specified package. If you do not want this, add the switch `-D`. To install only the build dependencies use `-d`.

```
zypper source-install -D package_name # source package only
zypper source-install -d package_name # build dependencies only
```

Of course, this will only work if you have the repository with the source packages enabled in your repository list (it is added by default, but not enabled). See Section 8.1.4, “Managing Repositories with Zypper” (page 132) for details on repository management.

A list of all source packages available in your repositories can be obtained with:

```
zypper search -t srcpackage
```

Utilities

To verify whether all dependencies are still fulfilled and to repair missing dependencies, use:

```
zypper verify
```

In addition to dependencies that must be fulfilled, some packages “recommend” other packages. These recommended packages are only installed if actually available and installable. In case recommended packages were made available after the recommending package has been installed (by adding additional packages or hardware), use the following command:

```
zypper install-new-recommends
```

This command is very useful after plugging in a webcam or WLAN device. It will install drivers for the device and related software, if available. Drivers and related software are only installable if certain hardware dependencies are fulfilled.

8.1.3 Updating Software with Zypper

There are three different ways to update software using Zypper: by installing patches, by installing a new version of a package or by updating the entire distribution. The latter is achieved with the `zypper dist-upgrade` command which is discussed in Section 15.1, “Upgrading the System” (page 209).

Installing Patches

To install all officially released patches applying to your system, just run:

```
zypper patch
```

In this case, all patches available in your repositories are checked for relevance and installed, if necessary. The above command is all you must enter in order to apply them when needed.

Zypper knows three different commands to query for the availability of patches:

`zypper patch-check`

Lists the number of needed patches (patches, that apply to your system but are not yet installed)

```
~ # zypper patch-check
Loading repository data...
Reading installed packages...
5 patches needed (1 security patch)
```

`zypper list-patches`

Lists all needed patches (patches, that apply to your system but are not yet installed)

```
~ # zypper list-patches
Loading repository data...
Reading installed packages...

Repository | Name | Version | Category | Status
-----+-----+-----+-----+-----
Updates for openSUSE 11.3 11.3-1.82 | lxsession | 2776 | security | needed
```

`zypper patches`

Lists all patches available for openSUSE, regardless of whether they are already installed or apply to your installation.

It is also possible to list and install patches relevant to specific issues. To list specific patches, use the `zypper list-patches` command with the following options:

`--bugzilla [=number]`

Lists all needed patches for Bugzilla issues. Optionally, you can specify a bug number if you only want to list patches for this specific bug.

`--cve [=number]`

Lists all needed patches for CVE (Common Vulnerabilities and Exposures) issues, or only patches matching a certain CVE number, if specified.

To install a patch for a specific Bugzilla or CVE issue, use the following commands:

```
zypper patch --bugzilla=number
```

or

```
zypper patch --cve=number
```

For example, to install a security patch with the CVE number CVE-2010-2713, execute:

```
zypper patch --cve=CVE-2010-2713
```

Installing Updates

If a repository contains only new packages, but does not provide patches, `zypper patch` does not show any effect. To update all installed packages with newer available versions, use:

```
zypper update
```

To update individual packages, specify the package with either the `update` or `install` command:

```
zypper update package_name  
zypper install package_name
```

A list of all new installable packages can be obtained with the command:

```
zypper list-updates
```

Note that this command only packages lists packages that match the following criteria:

- has the same vendor like the already installed package
- is provided by repositories with at least the same priority than the already installed package
- is installable (all dependencies are satisfied)

A list of *all* new available packages (regardless whether installable or not) can be obtained with:

```
zypper list-updates --all
```

To find out why a new package cannot be installed, just use the `zypper install` or `zypper update` command as described above.

Upgrading to a New Product Version

To easily upgrade your installation to a new product version (for example, from openSUSE 11.2 to openSUSE 11.3), first adjust your repositories to match the current openSUSE repositories. For details, refer to Section 8.1.4, “Managing Repositories with Zypper” (page 132). Then use the `zypper dist-upgrade` command with the required repositories. This command ensures that all packages will be installed from the repositories currently enabled. For detailed instructions, refer to Section 15.1.4, “Distribution Upgrade with zypper” (page 212).

To restrict the distribution upgrade to packages from a certain repository while considering also the other repositories for satisfying dependencies, use the `--from` option and specify the repository by either its alias, its number or URI.

NOTE: Differences between `zypper update` and `zypper dist-upgrade`

Choose `zypper update` to update packages to newer versions available for your product version while maintaining system integrity. `zypper update` will honor the following rules:

- no vendor changes
- no architecture changes
- no downgrades
- keep installed packages

When executing `zypper dist-upgrade`, all packages will be installed from the repositories currently enabled. This rule is enforced, so packages might change vendor or architecture or even might get downgraded. All packages that have unfulfilled dependencies after the upgrade will be uninstalled.

8.1.4 Managing Repositories with Zypper

All installation or patch commands of Zypper rely on a list of known repositories. To list all repositories known to the system, use the command:

```
zypper repos
```

The result will look similar to the following output:

Example 8.1 Zypper—List of Known Repositories

#	Alias	Name	Enabled	Refresh
1	Updates	Updates	Yes	Yes
2	openSUSE 11.2-0	openSUSE 11.2-0	No	No
3	openSUSE-11.2-Debug	openSUSE-11.2-Debug	No	Yes
4	openSUSE-11.2-Non-Oss	openSUSE-11.2-Non-Oss	Yes	Yes
5	openSUSE-11.2-Oss	openSUSE-11.2-Oss	Yes	Yes
6	openSUSE-11.2-Source	openSUSE-11.2-Source	No	Yes

When specifying repositories in various commands, an alias, URI or repository number from the `zypper repos` command output can be used. A repository alias is a short version of the repository name for use in repository handling commands. Note that the repository numbers can change after modifying the list of repositories. The alias will never change by itself.

By default, details as the URI or the priority of the repository are not displayed. Use the following command to list all details:

```
zypper repos -d
```

Adding Repositories

To add a repository, run

```
zypper addrepo URI alias
```

URI can either be an Internet repository, a network resource, a directory or a CD or DVD (see http://en.opensuse.org/openSUSE:Libzypp_URIs for details). The *alias* is a shorthand and unique identifier of the repository. You can freely choose it, with the only exception that it has to be unique. Zypper will issue a warning if you specify an alias that is already in use.

Removing Repositories

If you want to remove a repository from the list, use the command `zypper remove-repo` together with the alias or number of the repository you want to delete. For example, to remove the repository listed as third entry in Example 8.1, “Zypper—List of Known Repositories” (page 132), use the following command:

```
zypper remove-repo 3
```

Modifying Repositories

Enable or disable repositories with `zypper modify-repo`. You can also alter the repository's properties (such as refreshing behavior, name or priority) with this command. The following command will enable the repository named `updates`, turn on auto-refresh and set its priority to 20:

```
zypper modify-repo -er -p 20 'updates'
```

Modifying repositories is not limited to a single repository—you can also operate on groups:

-a: all repositories
-l: local repositories
-t: remote repositories
-m *TYPE*: repositories of a certain type (where *TYPE* can be one of the following:
http, https, ftp, cd, dvd, dir, file, cifs, smb, nfs, hd, iso)

To rename a repository alias, use the `renamerepo` command. The following example changes the alias from `Mozilla Firefox` to just `firefox`:

```
zypper renamerepo 'Mozilla Firefox' firefox
```

8.1.5 Querying Repositories and Packages with Zypper

Zypper offers various methods to query repositories or packages. To get lists of all products, patterns, packages or patches available, use the following commands:

```
zypper products  
zypper patterns  
zypper packages  
zypper patches
```

To query all repositories for certain packages, use `search`. It works on package names, or, optionally, on package summaries and descriptions. Using the wildcards `*` and `?` with the search term is allowed. By default, the search is not case-sensitive.

```
zypper search firefox      # simple search for "firefox"  
zypper search "**fire*"    # using wildcards  
zypper search -d fire      # also search in package descriptions and  
    summaries              #  
zypper search -u firefox   # only display packages not already installed
```

To search for packages which provide a special capability, use the command `what-provides`. For example, if you would like to know which package provides the `perl` module `SVN::Core`, use the following command:

```
zypper what-provides 'perl(SVN::Core)'
```

To query single packages, use `info` with an exact package name as an argument. It displays detailed information about a package. To also show what is required/recommended by the package, use the options `--requires` and `--recommends`:

```
zypper info --requires MozillaFirefox
```

The `what-provides package` is similar to `rpm -q --whatprovides package`, but `rpm` is only able to query the RPM database (that is the database of

all installed packages). Zypper, on the other hand, will tell you about providers of the capability from any repository, not only those that are installed.

8.1.6 Configuring Zypper

Zypper now comes with a configuration file, allowing you to permanently change Zypper's behavior (either system-wide or user-specific). For system-wide changes, edit `/etc/zypp/zypper.conf`. For user-specific changes, edit `~/.zypper.conf`. If `~/.zypper.conf` does not yet exist, you can use `/etc/zypp/zypper.conf` as template: copy it to `~/.zypper.conf` and adjust it to your liking. Refer to the comments in the file for help about the available options.

8.1.7 Troubleshooting

In case you have problems to access packages from configured repositories (for example, zypper cannot find a certain package though you know that it exists in one the repositories), it can help to refresh the repositories with:

```
zypper refresh
```

If that does not help, try

```
zypper refresh -fdb
```

This forces a complete refresh and rebuild of the database, including a forced download of raw metadata.

8.1.8 For More Information

For more information on managing software from the command line, enter `zypper help`, `zypper help command` or refer to the `zypper(8)` manpage. For a complete and detailed command reference, including cheat sheets with the most important commands, and information on how to use Zypper in scripts and applications, refer to http://en.opensuse.org/SDB:Zypper_usage. A list of software changes for the latest openSUSE version can be found at http://en.opensuse.org/openSUSE:Zypper_versions].

8.2 RPM—the Package Manager

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

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

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

TIP: Software Development Packages

For a number of packages, the components needed for software development (libraries, headers, include files, etc.) have been put into separate packages. These development packages are only needed if you want to compile software yourself (for example, the most recent GNOME packages). They can be identified by the name extension `-devel`, such as the packages `alsa-devel`, `gimp-devel`, and `libkde4-devel`.

8.2.1 Verifying Package Authenticity

RPM packages have a GnuPG signature. To verify the signature of an RPM package, use the command `rpm --checksig package-1.2.3.rpm` to determine whether the package originates from Novell/SUSE or from another trustworthy facility. This is especially recommended for update packages from the Internet.

8.2.2 Managing Packages: Install, Update, and Uninstall

Normally, the installation of an RPM archive is quite simple: `rpm -i package.rpm`. With this command the package is installed, but only if its depen-

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

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

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

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

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

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

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

8.2.3 RPM and Patches

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

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

Is the patch RPM suitable for my system?

To check this, first query the installed version of the package. For `pine`, this can be done with

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

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

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

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

Which files are replaced by the patch?

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

```
rpm -qpPl pine-4.44-224.i586.patch.rpm
```

```
/etc/pine.conf
/etc/pine.conf.fixed
/usr/bin/pine
```

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

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

How can a patch RPM be installed in the system?

Patch RPMs are used just like normal RPMs. The only difference is that a suitable RPM must already be installed.

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

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

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

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

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

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

NOTE: Official Updates for openSUSE

In order to make the download size of updates as small as possible, official updates for openSUSE are not provided as Patch RPMs, but as Delta RPM packages. For details, see Section 8.2.4, “Delta RPM Packages” (page 139).

8.2.4 Delta RPM Packages

Delta RPM packages contain the difference between an old and a new version of an RPM package. Applying a delta RPM onto an old RPM results in a completely new

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

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

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

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

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

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

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

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

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

8.2.5 RPM Queries

With the `-q` option `rpm` initiates queries, making it possible to inspect an RPM archive (by adding the option `-p`) and also to query the RPM database of installed packages. Several switches are available to specify the type of information required. See Table 8.1, “The Most Important RPM Query Options” (page 140).

Table 8.1 *The Most Important RPM Query Options*

<code>-i</code>	Package information
-----------------	---------------------

For example, the command `rpm -q -i wget` displays the information shown in Example 8.2, “`rpm -q -i wget`” (page 141).

Example 8.2 *rpm -q -i wget*

```
Name           : wget                               Relocations: (not relocatable)
Version        : 1.11.4                             Vendor: openSUSE
Release       : 1.70                                Build Date: Sat 01 Aug 2009
              09:49:48 CEST
Install Date: Thu 06 Aug 2009 14:53:24 CEST      Build Host: build18
Group         : Productivity/Networking/Web/Utilities Source RPM:
              wget-1.11.4-1.70.src.rpm
Size          : 1525431                             License: GPL v3 or later
Signature     : RSA/8, Sat 01 Aug 2009 09:50:04 CEST, Key ID b88b2fd43dbdc284
Packager      : http://bugs.opensuse.org
URL           : http://www.gnu.org/software/wget/
Summary       : A Tool for Mirroring FTP and HTTP Servers
Description   :
Wget enables you to retrieve WWW documents or FTP files from a server.
This can be done in script files or via the command line.
[...]
```

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

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

results in:

```
rpm-4.8.0-4.3.x86_64
wget-1.11.4-11.18.x86_64
```

If only part of the filename is known, use a shell script as shown in Example 8.3, “Script to Search for Packages” (page 141). Pass the partial filename to the script shown as a parameter when running it.

Example 8.3 *Script to Search for Packages*

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

The command `rpm -q --changelog rpm` displays a detailed list of change information about a specific package (in this case, the `rpm` package), sorted by date.

With the help of the installed RPM database, verification checks can be made. Initiate these with `-V`, `-y` or `--verify`. With this option, `rpm` shows all files in a pack-

age that have been changed since installation. `rpm` uses eight character symbols to give some hints about the following changes:

Table 8.2 RPM Verify Options

5	MD5 check sum
---	---------------

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

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

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

8.2.6 Installing and Compiling Source Packages

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

NOTE: Installed Source Packages

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

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

SOURCES

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

SPECS

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

BUILD

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

RPMS

where the completed binary packages are stored

SRPMS

here are the source RPMs

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

WARNING

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

The following example uses the `wget.src.rpm` package. After installing the source package, you should have files similar to those in the following list:

```
/usr/src/packages/SOURCES/wget-1.11.4.tar.bz2
/usr/src/packages/SOURCES/wgetrc.patch
/usr/src/packages/SPECS/wget.spec
```

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

`-bp`

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

`-bc`

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

`-bi`

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

`-bb`

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

`-ba`

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

`--short-circuit`

Skip some steps.

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

8.2.7 Compiling RPM Packages with build

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

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

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

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

8.2.8 Tools for RPM Archives and the RPM Database

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

A full-featured package manager is available as a YaST module. For details, see Chapter 4, *Installing or Removing Software* (page 87).

Part II. Administration

Managing Users with YaST

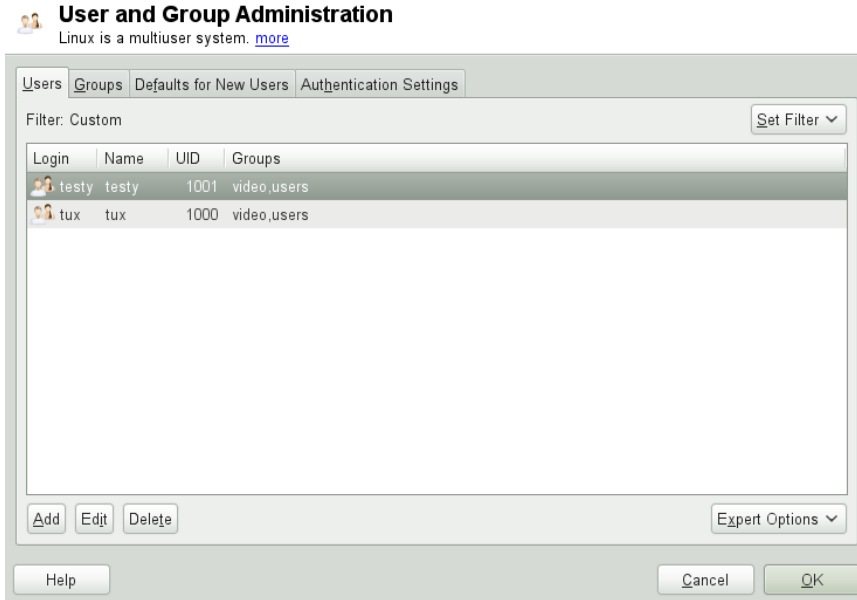
During installation, you chose a method for user authentication. This method is either local (via `/etc/passwd`) or, if a network connection is established, via NIS, LDAP, Kerberos or Samba (see Section 1.11, “Create New User” (page 20)). You can create or modify user accounts and change the authentication method with YaST at any time.

Every user is assigned a system-wide user ID (UID). Apart from the users which can log in to your machine, there are also a number of *system users* for internal use only. Each user is assigned to one or more groups. Similar to *system users*, there are also *system groups* for internal use.

9.1 User and Group Administration Dialog

To administer users or groups, start YaST and click *Security and Users > User and Group Management*. Alternatively, start the *User and Group Administration* dialog directly by running `yast2 users` from a command line.

Figure 9.1 YaST User and Group Administration



Depending on the set of users you choose to view and modify with, the dialog (local users, network users, system users), the main window shows several tabs. These allow you to execute the following tasks:

Managing User Accounts

From the *Users* tab create, modify, delete or temporarily disable user accounts as described in Section 9.2, “Managing User Accounts” (page 151). Learn about advanced options like enforcing password policies, using encrypted home directories, using fingerprint authentication, or managing disk quotas in Section 9.3, “Additional Options for User Accounts” (page 153).

Changing Default Settings

Local users accounts are created according to the settings defined on the *Defaults for New Users* tab. Learn how to change the default group assignment, or the default path and access permissions for home directories in Section 9.4, “Changing Default Settings for Local Users” (page 160).

Assigning Users to Groups

Learn how to change the group assignment for individual users in Section 9.5, “Assigning Users to Groups” (page 160).

Managing Groups

From the *Groups* tab, you can add, modify or delete existing groups. Refer to Section 9.6, “Managing Groups” (page 161) for information on how to do this.

Changing the User Authentication Method

When your machine is connected to a network that provides user authentication methods like NIS or LDAP, you can choose between several authentication methods on the *Authentication Settings* tab. For more information, refer to Section 9.7, “Changing the User Authentication Method” (page 162).

For user and group management, the dialog provides similar functionality. You can easily switch between the user and group administration view by choosing the appropriate tab at the top of the dialog.

Filter options allow you to define the set of users or groups you want to modify: On the *Users* or *Group* tab, click *Set Filter* to view and edit users or groups according to certain categories, such as *Local Users* or *LDAP Users*, for instance (if you are part of a network which uses LDAP). With *Set Filter > Customize Filter* you can also set up and use a custom filter.

Depending on the filter you choose, not all of the following options and functions will be available from the dialog.

9.2 Managing User Accounts

YaST offers to create, modify, delete or temporarily disable user accounts. Do not modify user accounts unless you are an experienced user or administrator.

NOTE: Changing User IDs of Existing Users

File ownership is bound to the user ID, not to the user name. After a user ID change, the files in the user's home directory are automatically adjusted to reflect this change. However, after an ID change, the user no longer owns the files he created elsewhere in the file system unless the file ownership for those files are manually modified.

In the following, learn how to set up default user accounts. For some further options, such as auto login, login without password, setting up encrypted home directories or

managing quotas for users and groups, refer to Section 9.3, “Additional Options for User Accounts” (page 153).

Procedure 9.1 *Adding or Modifying User Accounts*

- 1 Open the YaST *User and Group Administration* dialog and click the *Users* tab.
- 2 With *Set Filter* define the set of users you want to manage. The dialog shows a list of users in the system and the groups the users belong to.
- 3 To modify options for an existing user, select an entry and click *Edit*.

To create a new user account, click *Add*.

- 4 Enter the appropriate user data on the first tab, such as *Username* (which is used for login) and *Password*. This data is sufficient to create a new user. If you click *OK* now, the system will automatically assign a user ID and set all other values according to the default.
- 5 Activate *Receive System Mail* if you want any kind of system notifications to be delivered to this user's mailbox. This creates a mail alias for `root` and the user can read the system mail without having to first log in as `root`.
- 6 If you want to adjust further details such as the user ID or the path to the user's home directory, do so on the *Details* tab.

If you need to relocate the home directory of an existing user, enter the path to the new home directory there and move the contents of the current home directory with *Move to New Location*. Otherwise, a new home directory is created without any of the existing data.

- 7 To force users to regularly change their password or set other password options, switch to *Password Settings* and adjust the options. For more details, refer to Section 9.3.2, “Enforcing Password Policies” (page 154).
- 8 If all options are set according to your wishes, click *OK*.
- 9 Click *Expert Options > Write Changes Now* to save all changes without exiting the *User and Group Administration* dialog. Click *OK* to close the administration dialog and to save the changes. A newly added user can now log in to the system using the login name and password you created.

TIP: Matching User IDs

For a new (local) user on a laptop which also needs to integrate into a network environment where this user already has a user ID, it is useful to match the (local) user ID to the ID in the network. This ensures that the file ownership of the files the user creates “offline” is the same as if he had created them directly on the network.

Procedure 9.2 *Disabling or Deleting User Accounts*

- 1 Open the YaST *User and Group Administration* dialog and click the *Users* tab.
- 2 To temporarily disable a user account without deleting it, select the user from the list and click *Edit*. Activate *Disable User Login*. The user cannot log into your machine until you enable the account again.
- 3 To delete a user account, select the user from the list and click *Delete*. Choose if you also want to delete the user's home directory or if you want to retain the data.

9.3 Additional Options for User Accounts

In addition to the settings for a default user account, openSUSE® offers further options, such as options to enforce password policies, use encrypted home directories or define disk quotas for users and groups.

9.3.1 Automatic Login and Passwordless Login

If you use the KDE or GNOME desktop environment you can configure *Auto Login* for a certain user as well as *Passwordless Login* for all users. Auto login causes a user to become automatically logged in to the desktop environment on boot. This functionality can only be activated for one user at a time. Login without password allows all users to log in to the system after they have entered their username in the login manager.

WARNING: Security Risk

Enabling *Auto Login* or *Passwordless Login* on a machine that can be accessed by more than one person is a security risk. Without the need to authenticate, any user can gain access to your system and your data. If your system contains confidential data, do not use this functionality.

If you want to activate auto login or login without password, access these functions in the YaST *User and Group Administration* with *Expert Options > Login Settings*.

9.3.2 Enforcing Password Policies

On any system with multiple users, it is a good idea to enforce at least basic password security policies. Users should change their passwords regularly and use strong passwords that cannot easily be exploited. For local users, proceed as follows:

Procedure 9.3 *Configuring Password Settings*

- 1 Open the YaST *User and Group Administration* dialog and select the *Users* tab.
- 2 Select the user for which to change the password options and click *Edit*.
- 3 Switch to the *Password Settings* tab. The user's last password change is displayed on the tab.
- 4 To make the user change his password at next login, activate *Force Password Change*.
- 5 To enforce password rotation, set a *Maximum Number of Days for the Same Password* and a *Minimum Number of Days for the Same Password*.
- 6 To remind the user to change his password before it expires, set a number of *Days before Password Expiration to Issue Warning*.
- 7 To restrict the period of time the user can log in after his password has expired, change the value in *Days after Password Expires with Usable Login*.
- 8 You can also specify a certain expiration date for a password. Enter the *Expiration Date* in *YYYY-MM-DD* format.

9 For more information about the options and about the default values, click *Help*.

10 Apply your changes with *OK*.

9.3.3 Managing Encrypted Home Directories

To protect data in home directories against theft and hard disk removal, you can create encrypted home directories for users. These are encrypted with LUKS (Linux Unified Key Setup), which results in an image and an image key being generated for the user. The image key is protected with the user's login password. When the user logs into the system, the encrypted home directory is mounted and the contents are made available to the user.

NOTE: Fingerprint Reader Devices and Encrypted Home Directories

If you want to use a fingerprint reader device, you must not use encrypted home directories. Otherwise logging in will fail, because decrypting during login is not possible in combination with an active fingerprint reader device.

With YaST, you can create encrypted home directories for new or existing users. To encrypt or modify encrypted home directories of already existing users, you need to know the user's current login password. By default, all existing user data is copied to the new encrypted home directory, but it is not deleted from the unencrypted directory.

WARNING: Security Restrictions

Encrypting a user's home directory does not provide strong security from other users. If strong security is required, the system should not be physically shared.

Find background information about encrypted home directories and which actions to take for stronger security in Section “Using Encrypted Home Directories” (Chapter 11, *Encrypting Partitions and Files*, ↑Security Guide).

Procedure 9.4 *Creating Encrypted Home Directories*

1 Open the YaST *User and Group Management* dialog and click the *Users* tab.

- 2 To encrypt the home directory of an existing user, select the user and click *Edit*.

Otherwise, click *Add* to create a new user account and enter the appropriate user data on the first tab.

- 3 In the *Details* tab, activate *Use Encrypted Home Directory*. With *Directory Size in MB*, specify the size of the encrypted image file to be created for this user.

Existing Local User
Additional user data includes: User ID (uid): Each user is known to the system by a unique number, the user ID. [more](#)

User Data Details Password Settings Plug-Ins

User ID (uid): 1000

Home Directory: /home/tux [Browse...](#)

☒ Move to New Location

Directory Size in MB: 100

☒ Use Encrypted Home Directory

Additional User Information:

Login Shell: /bin/bash

Default Group: users

Additional Groups:

- ☐ users
- ☐ at
- ☐ audio
- ☐ avahi
- ☐ beagleindex
- ☐ bin
- ☐ cdrom
- ☐ console
- ☐ daemon
- ☐ dialout
- ☐ disk
- ☐ floppy
- ☐ ftp
- ☐ games
- ☐ audio

Help Cancel OK

- 4 Apply your settings with *OK*.
- 5 Enter the user's current login password to proceed if YaST prompts for it.
- 6 Click *Expert Options > Write Changes Now* to save all changes without exiting the administration dialog. Click *OK* to close the administration dialog and save the changes.

Procedure 9.5 *Modifying or Disabling Encrypted Home Directories*

Of course, you can also disable the encryption of a home directory or change the size of the image file at any time.

- 1 Open the YaST *User and Group Administration* dialog in the *Users* view.

- 2 Select a user from the list and click *Edit*.
- 3 If you want to disable the encryption, switch to the *Details* tab and disable *Use Encrypted Home Directory*.

If you need to enlarge or reduce the size of the encrypted image file for this user, change the *Directory Size in MB*.
- 4 Apply your settings with *OK*.
- 5 Enter the user's current login password to proceed if YaST prompts for it.
- 6 Click *Expert Options > Write Changes Now* to save all changes without exiting the *User and Group Administration* dialog. Click *OK* to close the administration dialog and to save the changes.

9.3.4 Using Fingerprint Authentication

If your system includes a fingerprint reader you can use biometric authentication in addition to standard authentication via login and password. After registering their fingerprint, users can log into the system either by swiping a finger on the fingerprint reader or by typing in a password.

Fingerprints can be registered with YaST. Find detailed information about configuration and use of fingerprint authentication in Chapter 7, *Using the Fingerprint Reader* (↑Security Guide). For a list of supported devices, refer to http://reactivated.net/fprint/wiki/Supported_devices.

9.3.5 Managing Quotas

To prevent system capacities from being exhausted without notification, system administrators can set up quotas for users or groups. Quotas can be defined for one or more file systems and restrict the amount of disk space that can be used and the number of inodes (index nodes) that can be created there. Inodes are data structures on a file system that store basic information about a regular file, directory, or other file system object. They store all attributes of a file system object (like user and group ownership, read, write, or execute permissions), except file name and contents.

openSUSE allows usage of `soft` and `hard` quotas. Soft quotas usually define a warning level at which users are informed that they are nearing their limit, whereas hard quotas define the limit at which write requests are denied. Additionally, grace intervals can be defined that allow users or groups to temporarily violate their quotas by certain amounts.

Procedure 9.6 Enabling Quota Support for a Partition

In order to configure quotas for certain users and groups, you need to enable quota support for the respective partition in the YaST Expert Partitioner first.

- 1** In YaST, select *System > Partitioner* and click *Yes* to proceed.
- 2** In the *Expert Partitioner*, select the partition for which to enable quotas and click *Edit*.
- 3** Click *Fstab Options* and activate *Enable Quota Support*. If the `quota` package is not already installed, it will be installed once you confirm the respective message with *Yes*.
- 4** Confirm your changes and leave the *Expert Partitioner*.

Procedure 9.7 Setting Up Quotas for Users or Groups

Now you can define soft or hard quotas for specific users or groups and set time periods as grace intervals.

- 1** In the YaST *User and Group Administration*, select the user or the group you want to set the quotas for and click *Edit*.
- 2** On the *Plug-Ins* tab, select the *Manage User Quota* entry and click *Launch* to open the *Quota Configuration* dialog.
- 3** From *File System*, select the partition to which the quota should apply.



Quota Configuration

Here, configure quota settings of the user on selected file systems. [more](#)

File System:

Size Limits

Soft limit:

Hard limit:

Days: Hours: Minutes: Seconds:

I-nodes Limits

Soft limit:

Hard limit:

Days: Hours: Minutes: Seconds:

- 4 Below *Size Limits*, restrict the amount of disk space. Enter the number of 1 KB blocks the user or group may have on this partition. Specify a *Soft Limit* and a *Hard Limit* value.
- 5 Additionally, you can restrict the number of inodes the user or group may have on the partition. Below *Inodes Limits*, enter a *Soft Limit* and *Hard Limit*.
- 6 You can only define grace intervals if the user or group has already exceeded the soft limit specified for size or inodes. Otherwise, the time-related input fields are not activated. Specify the time period for which the user or group is allowed to exceed the limits set above.
- 7 Confirm your settings with *OK*.
- 8 Click *Expert Options > Write Changes Now* to save all changes without exiting the *User and Group Administration* dialog. Click *OK* to close the administration dialog and to save the changes.

openSUSE also ships command line tools like `repquota` or `warnquota` with which system administrators can control the disk usage or send e-mail notifications to users exceeding their quota. With `quota_nld`, administrators can also forward

kernel messages about exceeded quotas to D-BUS. For more information, refer to the `repquota`, the `warnquota` and the `quota_nld` man page.

9.4 Changing Default Settings for Local Users

When creating new local users, several default settings are used by YaST. These include, for example, the primary group and the secondary groups the user belongs to, or the access permissions of the user's home directory. You can change these default settings to meet your requirements:

- 1 Open the YaST *User and Group Administration* dialog and select the *Defaults for New Users* tab.
- 2 To change the primary group the new users should automatically belong to, select another group from *Default Group*.
- 3 To modify the secondary groups for new users, add or change groups in *Secondary Groups*. The group names must be separated by commas.
- 4 If you do not want to use `/home/username` as default path for new users' home directories, modify the *Path Prefix for Home Directory*.
- 5 To change the default permission modes for newly created home directories, adjust the umask value in *Umask for Home Directory*. For more information about umask, refer to Chapter 10, *Access Control Lists in Linux* (↑Security Guide) and to the `umask` man page.
- 6 For information about the individual options, click *Help*.
- 7 Apply your changes with *OK*.

9.5 Assigning Users to Groups

Local users are assigned to several groups according to the default settings which you can access from the *User and Group Administration* dialog on the *Defaults for New Users* tab. In the following, learn how to modify an individual user's group assignment. If you need to change the default group assignments for new users, refer to Section 9.4, “Changing Default Settings for Local Users” (page 160).

Procedure 9.8 *Changing a User's Group Assignment*

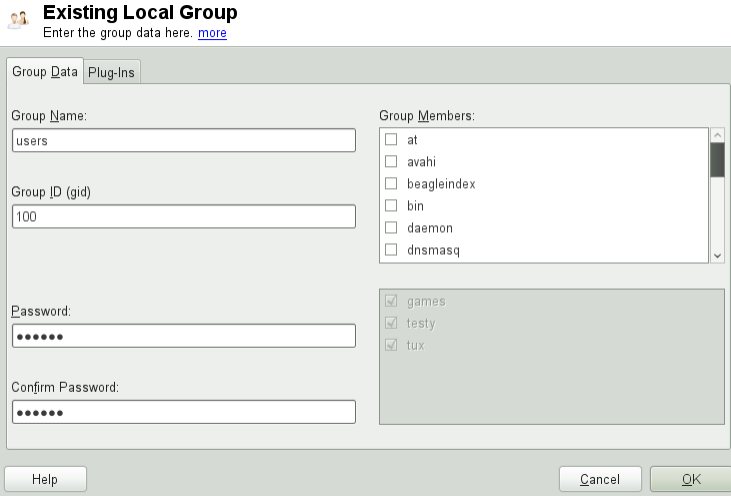
- 1 Open the YaST *User and Group Administration* dialog and click the *Users* tab. It shows a list of users and of the groups the users belong to.
- 2 Click *Edit* and switch to the *Details* tab.
- 3 To change the primary group the user belongs to, click *Default Group* and select the group from the list.
- 4 To assign the user additional secondary groups, activate the corresponding check boxes in the *Additional Groups* list.
- 5 Click *OK* to apply your changes.
- 6 Click *Expert Options > Write Changes Now* to save all changes without exiting the *User and Group Administration* dialog. Click *OK* to close the administration dialog and save the changes.

9.6 Managing Groups

With YaST you can also easily add, modify or delete groups.

Procedure 9.9 *Creating and Modifying Groups*

- 1 Open the YaST *User and Group Management* dialog and click the *Groups* tab.
- 2 With *Set Filter* define the set of groups you want to manage. The dialog shows a list of groups in the system.
- 3 To create a new group, click *Add*.
- 4 To modify an existing group, select the group and click *Edit*.
- 5 In the following dialog, enter or change the data. The list on the right shows an overview of all available users and system users which can be members of the group.



- 6 To add existing users to a new group select them from the list of possible *Group Members* by checking the corresponding box. To remove them from the group just uncheck the box.
- 7 Click *OK* to apply your changes.
- 8 Click *Expert Options > Write Changes Now* to save all changes without exiting the *User and Group Administration* dialog.

In order to delete a group, it must not contain any group members. To delete a group, select it from the list and click *Delete*. Click *Expert Options > Write Changes Now* to save all changes without exiting the *User and Group Administration* dialog. Click *OK* to close the administration dialog and to save the changes.

9.7 Changing the User Authentication Method

When your machine is connected to a network, you can change the authentication method you set during installation. The following options are available:

NIS

Users are administered centrally on a NIS server for all systems in the network. For details, see Chapter 3, *Using NIS* (↑Security Guide).

LDAP

Users are administered centrally on an LDAP server for all systems in the network. For details about LDAP, see Chapter 4, *LDAP—A Directory Service* (↑Security Guide).

You can manage LDAP users with the YaST user module. All other LDAP settings, including the default settings for LDAP users, have to be defined with the YaST LDAP client module as described in Section “Configuring an LDAP Client with YaST” (Chapter 4, *LDAP—A Directory Service*, ↑Security Guide).

Kerberos

With Kerberos, a user registers once and then is trusted in the entire network for the rest of the session.

Samba

SMB authentication is often used in mixed Linux and Windows networks. For details, see Chapter 29, *Samba* (page 455) and Chapter 5, *Active Directory Support* (↑Security Guide).

To change the authentication method, proceed as follows:

- 1 Open the *User and Group Administration* dialog in YaST.
- 2 Click the *Authentication Settings* tab to show an overview of the available authentication methods and the current settings.
- 3 To change the authentication method, click *Configure* and select the authentication method you want to modify. This takes you directly to the client configuration modules in YaST. For information about the configuration of the appropriate client, refer to the following sections:

NIS: Section “Configuring NIS Clients” (Chapter 3, *Using NIS*, ↑Security Guide)

LDAP: Section “Configuring an LDAP Client with YaST” (Chapter 4, *LDAP—A Directory Service*, ↑Security Guide)

- 4 After accepting the configuration, return to the *User and Group Administration* overview.

5 Click *OK* to close the administration dialog.

Changing Language and Country Settings with YaST

10

Working in different countries or having to work in a multilingual environment requires your computer to be set up to support this. openSUSE® can handle different `locales` in parallel. A locale is a set of parameters that defines the language and country settings reflected in the user interface.

The main system language was selected during installation and keyboard and time zone settings were adjusted. However, you can install additional languages on your system and determine which of the installed languages should be the default.

For those tasks, use the YaST language module as described in Section 10.1, “Changing the System Language” (page 166). Install secondary languages to get optional localizations if you need to start applications or desktops in languages other than the primary one.

Apart from that, the YaST timezone module allows you to adjust your country and timezone settings accordingly. It also lets you synchronize your system clock against a time server. For details, refer to Section 10.2, “Changing the Country and Time Settings” (page 170).

10.1 Changing the System Language

Depending on how you use your desktop and whether you want to switch the entire system to another language or just the desktop environment itself, there are several ways to achieve this:

Changing the System Language Globally

Proceed as described in Section 10.1.1, “Modifying System Languages with YaST” (page 166) and Section 10.1.2, “Switching the Default System Language” (page 168) to install additional localized packages with YaST and to set the default language. Changes are effective after relogin. To ensure that the entire system reflects the change, reboot the system or close and restart all running services, applications, and programs.

Changing the Language for the Desktop Only

Provided you have previously installed the desired language packages for your desktop environment with YaST as described below, you can switch the language of your desktop using the desktop's control center. After the X server has been restarted, your entire desktop reflects your new choice of language. Applications not belonging to your desktop framework are not affected by this change and may still appear in the language that was set in YaST.

Temporarily Switching Languages for One Application Only

You can also run a single application in another language (that has already been installed with YaST). To do so, start it from the command line by specifying the language code as described in Section 10.1.3, “Switching Languages for Individual Applications” (page 169).

10.1.1 Modifying System Languages with YaST

YaST knows two different language categories:

Primary Language

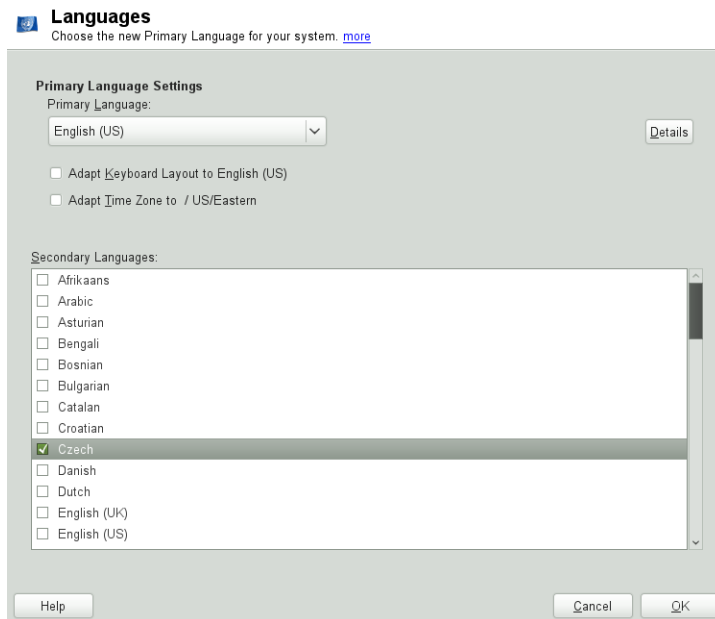
The primary language set in YaST applies to the entire system, including YaST and the desktop environment. This language is used whenever available unless you manually specify another language.

Secondary Languages

Install secondary languages to make your system multilingual. Languages installed as secondary languages can be selected manually for a specific situation. For example, use a secondary language to start an application in a certain language in order to do word processing in this language.

Before installing additional languages, determine which of them should be the default system language (primary language) after you have installed them.

To access the YaST language module, start YaST and click *System > Language*. Alternatively, start the *Languages* dialog directly by running `yast2 language` as user `root` from a command line.



Procedure 10.1 *Installing Additional Languages*

When installing additional languages, YaST also allows you to set different locale settings for the user `root`, see Step 4 (page 168). The option *Locale Settings for User root* determines how the locale variables (`LC_*`) in the file `/etc/sysconfig/language` are set for `root`. You can either set them to the same locale as for normal users, keep it unaffected by any language changes or only set the variable `RC_LC_CTYPE` to the same values as for the normal users. This variable sets the localization for language-specific function calls.

- 1 To add additional languages in the YaST language module, select the *Secondary Languages* you wish to install.
- 2 To make a language the default language, set it as *Primary Language*.
- 3 Additionally, adapt the keyboard to the new primary language and adjust the time zone, if appropriate.

TIP

For advanced keyboard or time zone settings, select *Hardware > Keyboard Layout* or *System > Date and Time* in YaST to start the respective dialogs. For more information, refer to Section 10.2, “Changing the Country and Time Settings” (page 170).

- 4 To change language settings specific to the user `root`, click *Details*.
 - 4a Set *Locale Settings for User root* to the desired value. For more information, click *Help*.
 - 4b Decide if you want to *Use UTF-8 Encoding* for `root` or not.
- 5 If your locale was not included in the list of primary languages available, try specifying it with *Detailed Locale Setting*. However, some of these localizations may be incomplete.
- 6 Confirm your changes in the dialogs with *OK*. If you have selected secondary languages, YaST installs the localized software packages for the additional languages.

The system is now multilingual. However, to start an application in a language other than the primary one, you need to set the desired language explicitly as explained in Section 10.1.3, “Switching Languages for Individual Applications” (page 169).

10.1.2 Switching the Default System Language

- 1 To globally switch the default system language, start the YaST language module.

- 2 Select the desired new system language as *Primary Language*.

IMPORTANT: Deleting Former System Languages

If you switch to a different primary language, the localized software packages for the former primary language will be removed from the system. If you want to switch the default system language but want to keep the former primary language as additional language, add it as *Secondary Language* by enabling the respective checkbox.

- 3 Adjust the keyboard and time zone options as desired.
- 4 Confirm your changes with *OK*.
- 5 After YaST has applied the changes, restart any X sessions (for example, by logging out and logging in again) to make YaST and the desktop applications reflect your new language settings.

10.1.3 Switching Languages for Individual Applications

After you have installed the respective language with YaST, you can run a single application in another language.

Standard X and GNOME Applications

Start the application from the command line by using the following command:

```
LANG=language application
```

For example, to start f-spot in German, run `LANG=de_DE f-spot`. For other languages, use the appropriate language code. Get a list of all language codes available with the `locale -av` command.

KDE Applications

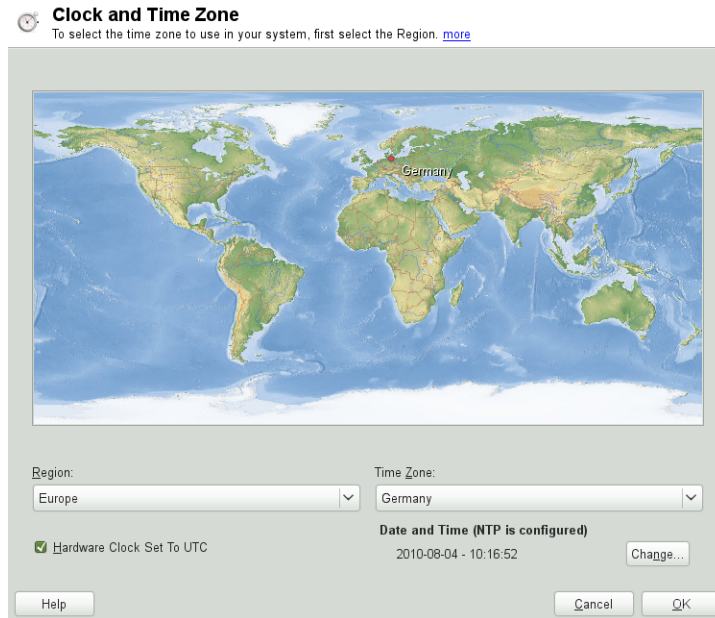
Start the application from the command line by using the following command:

```
KDE_LANG=language application
```

For example, to start digiKam in German, run `KDE_LANG=de digikam`. For other languages, use the appropriate language code.

10.2 Changing the Country and Time Settings

Using the YaST date and time module, adjust your system date, clock and time zone information to the area you are working in. To access the YaST module, start YaST and click *System > Date and Time*. Alternatively, start the *Clock and Time Zone* dialog directly by running `yast2 timezone` & as user `root` from a command line.



First, select a general region, such as *Europe*. Choose an appropriate country that matches the one you are working in, for example, *Germany*.

Depending on which operating systems run on your workstation, adjust the hardware clock settings accordingly:

- If you run another operating system on your machine, such as Microsoft Windows*, it is likely your system does not use UTC, but local time. In this case, uncheck *Hardware Clock Set To UTC*.
- 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.

You can change the date and time manually or opt for synchronizing your machine against an NTP server, either permanently or just for adjusting your hardware clock.

Procedure 10.2 Manually Adjusting Time and Date

- 1 In the YaST timezone module, click *Change* to set date and time.
- 2 Select *Manually* and enter date and time values.
- 3 Confirm your changes with *Accept*.

Procedure 10.3 Setting Date and Time With NTP Server

- 1 Click *Change* to set date and time.
- 2 Select *Synchronize with NTP Server*.
- 3 Enter the address of an NTP server, if not already populated.

Change Date and Time
The current system time and date are displayed. [more](#)

☐ **Manually**

Current Time:
[10] : [19]

Current Date:
[2010] - [08] - [04] [v]

☒ Change the Time Now

☒ **Synchronize with NTP Server**

NTP Server Address:
[1.opensuse.pool.ntp.org] [v] [Synchronize now]

☐ Run NTP as daemon [Configure...]

☒ Save NTP Configuration

[Help] [Cancel] [Accept]

- 4 Click *Synchronize Now*, to get your system time set correctly.
- 5 If you want to make use of NTP permanently, enable *Save NTP Configuration*.

- 6** With the *Configure* button, you can open the advanced NTP configuration. For details, see Section 27.1, “Configuring an NTP Client with YaST” (page 433).
- 7** Confirm your changes with *Accept*.

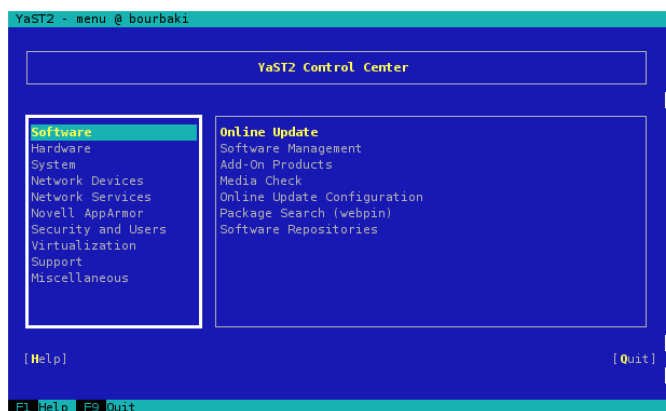
11

YaST in Text Mode

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

YaST in text mode uses the ncurses library to provide an easy pseudo-graphical user interface. The ncurses library is installed by default. The minimum supported size of the terminal emulator in which to run YaST is 80x25 characters.

Figure 11.1 *Main Window of YaST in Text Mode*



When you start YaST in text mode, the YaST Control Center appears (see Figure 11.1). The main window consists of three areas. The left frame features the categories to which the various modules belong. This frame is active when YaST is start-

ed and therefore it is marked by a bold white border. The active category is highlighted. The right frame provides an overview of the modules available in the active category. The bottom frame contains the buttons for *Help* and *Quit*.

When you start the YaST Control Center, the category *Software* is selected automatically. Use ↓ and ↑ to change the category. To select a module from the category, activate the right frame with → and then use ↓ and ↑ to select the module. Keep the arrow keys pressed to scroll through the list of available modules. The selected module is highlighted. Press **Enter** to start the active module.

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

11.1 Navigation in Modules

The following description of the control elements in the YaST modules assumes that all function keys and key combinations work and are not assigned to different global functions. Read Section 11.2, “Restriction of Key Combinations” (page 175) for information about possible exceptions.

Navigation among Buttons and Selection Lists

Use **Tab** to navigate among the buttons and frames containing selection lists. To navigate in reverse order, use **+** **Tab** or **+** **Tab** combinations.

Navigation in Selection Lists

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

Buttons, Radio Buttons, and Check Boxes

To select buttons with empty square brackets (check boxes) or empty parentheses (radio buttons), press **Space** or **.** Alternatively, radio buttons and check boxes can be selected directly with **+** highlighted_letter. In this case, you do not

need to confirm with `.` If you navigate to an item with `Tab`, press `.` to execute the selected action or activate the respective menu item.

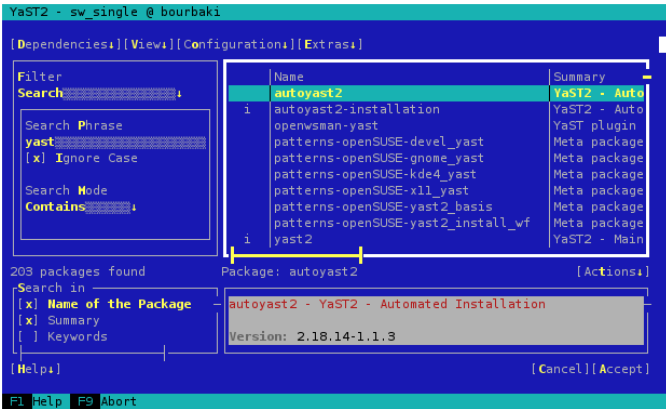
Function Keys

The F keys (F1 through F12) enable quick access to the various buttons. Available F key shortcuts are shown in the bottom line of the YaST screen. Which function keys are actually mapped to which buttons depend on the active YaST module, because the different modules offer different buttons (Details, Info, Add, Delete, etc.). Use F10 for *Accept*, *OK*, *Next*, and *Finish*. Press F1 to access the YaST help.

Using Navigation Tree in ncurses Mode

Some YaST modules use a navigation tree in the left part of the window to select configuration dialogs. Use the arrow keys (`↑` and `↓`) to navigate in the tree. Use `Space` to open or close tree items. In ncurses mode, `.` must be pressed after a selection in the navigation tree in order to show the selected dialog. This is an intentional behavior to save time consuming redraws when browsing through the navigation tree.

Figure 11.2 *The Software Installation Module*



11.2 Restriction of Key Combinations

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

Replacing with Esc

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

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

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

Restriction of Function Keys

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

11.3 YaST Command Line Options

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

```
yast -h
```

11.3.1 Starting the Individual Modules

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

```
yast <module_name>
```

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

11.3.2 Installing Packages from the Command Line

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

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

or

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

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

If you need a command-line based software management utility with functionality beyond what YaST provides, consider using `zypper`. This new utility uses the same software management library that is also the foundation for the YaST package manager. The basic usage of `zypper` is covered in Section 8.1, “Using Zypper” (page 125).

11.3.3 Command Line Parameters of the YaST Modules

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

```
yast <module_name> help
```

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

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


Setting Up Hardware Components with YaST

YaST allows you to configure hardware items at installation time as well as on an already-installed system. Configure audio hardware, printers or scanner support or learn which hardware components are connected to your computer by using the YaST Hardware Information module.

TIP: Graphics card, monitor, mouse and keyboard settings

Graphics card, monitor, mouse and keyboard can be configured with either KDE or GNOME tools.

12.1 Hardware Information

Use the YaST hardware information module if you want to know more about your hardware or if you need to find out details like vendor and model of a certain piece of hardware to be able to properly configure it.

- 1 Start YaST and click *Hardware > Hardware Information*. Hardware probing starts immediately and it will take some time until you see the hardware information tree in a separate window.
- 2 In the hardware information tree recursively click on the entries to expand the information about a specific device.
- 3 Click *Save to File...* to save the output to a file.

4 Click *Close* to leave the hardware information overview.

12.2 Setting Up Sound Cards

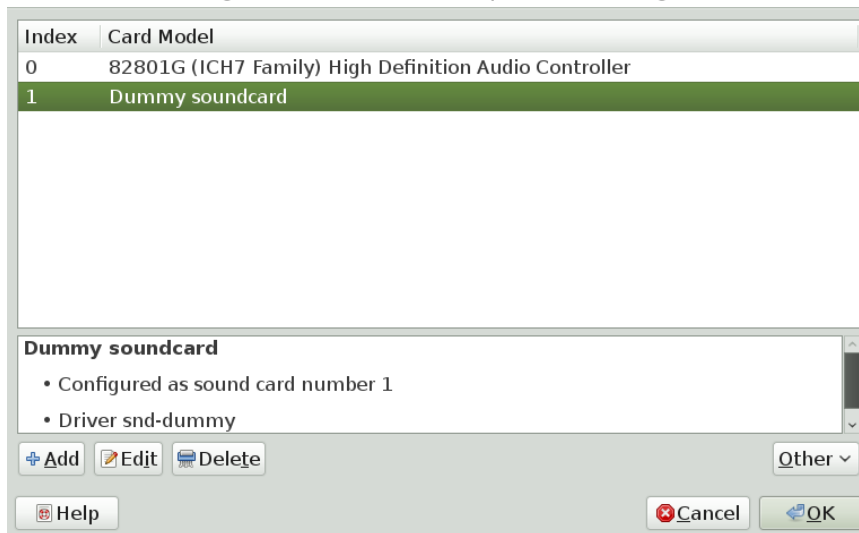
YaST detects most sound cards automatically and configures them with the appropriate values. If you want to change the default settings, or need to set up a sound card that could not be configured automatically, use the YaST sound module. There, you can also set up additional sound cards or switch their order.

To start the sound module, start YaST and click *Hardware > Sound*. Alternatively, start the *Sound Configuration* dialog directly by running `yast2 sound` & as user `root` from a command line.



Sound Configuration

Select an unconfigured card from the list and press Edit to configure it. [more](#)



The dialog shows all sound cards that are already configured.

If YaST cannot detect your sound card automatically, configure it manually as described in Procedure 12.1, “Configuring a New Sound Card” (page 181). For this, you need to know your sound card vendor and model. If in doubt, refer to your sound card documentation for the required information. During configuration, you can choose between various setup options:

Quick Automatic Setup

You are not required to go through any of the further configuration steps—the sound card is configured automatically. You can set the volume or any options you want to change later.

Normal Setup

Allows you to adjust the output volume and play a test sound during the configuration.

Advanced setup with possibility to change options

For experts only. Allows you to customize all parameters of the sound card.

IMPORTANT: Advanced Configuration

Only use this option if you know exactly what you are doing. Otherwise leave the parameters untouched and use the normal or the automatic setup options.

Procedure 12.1 *Configuring a New Sound Card*

- 1 In the YaST *Sound Configuration* dialog, click *Add*.
- 2 Select your sound card vendor and model and click *Next*.

For a reference list of sound cards supported by ALSA with their corresponding sound modules, see <http://www.alsa-project.org/main/index.php/Matrix:Main> .

- 3 Choose one of the setup options and click *Next*.
- 4 If you have chosen *Normal Setup*, you can now *Test* your sound configuration and make adjustments to the volume. You should start at about ten percent volume to avoid damage to your hearing or the speakers.
- 5 If all options are set according to your wishes, click *Next*.

The *Sound Configuration* dialog shows the newly configured or modified sound card.

- 6 To remove a sound card configuration that you no longer need, select the respective entry and click *Delete*.
- 7 Click *OK* to save the changes and leave the YaST sound module.

Procedure 12.2 *Modifying Sound Card Configurations*

- 1 To change the configuration of an individual sound card (for experts only!), select the sound card entry in the *Sound Configuration* dialog and click *Edit*.

This takes you to the *Sound Card Advanced Options* where you can fine-tune a number of parameters. For more information, click *Help*.

- 2 To adjust the volume of an already configured sound card or to test the sound card, select the sound card entry in the *Sound Configuration* dialog and click *Other*. Select the respective menu item.
- 3 For playback of MIDI files, select *Other > Start Sequencer*.
- 4 When a supported sound card is detected (like a Creative Soundblaster Live, Audigy or AWE sound card), you can also install SoundFonts for playback of MIDI files:

4a Insert the original driver CD-ROM into your CD or DVD drive.

4b Select *Other > Install SoundFonts* to copy SF2 SoundFonts™ to your hard disk. The SoundFonts are saved in the directory `/usr/share/sf-bank/creative/`.

- 5 If you have configured more than one sound card in your system you can adjust the order of your sound cards. To set a sound card as primary device, select the sound card in the *Sound Configuration* and click *Other > Set as the Primary Card*. The sound device with index 0 is the default device and thus used by the system and the applications.
- 6 To enable or disable the PulseAudio sound system, click *Other > PulseAudio Configuration*. If enabled, PulseAudio daemon is used to play sounds. Disable *PulseAudio Support* in case you want to use something else system-wide.

The volume and configuration of all sound cards are saved when you click *OK* and leave the YaST sound module. The mixer settings are saved to the file `/etc/asound.conf` and the ALSA configuration data is appended to the end of the files `/etc/modprobe.d/sound` and `/etc/sysconfig/hardware`.

12.3 Setting Up a Scanner

You can configure a USB or SCSI scanner with YaST. The `sane-backends` package contains hardware drivers and other essentials needed to use a scanner. Scanners connected to a parallel port cannot be configured with YaST. If you own a HP All-In-One device, see Section 12.3.1, “Configuring an HP All-In-One Device” (page 183), instructions on how to configure a network scanner are available at Section 12.3.3, “Scanning over the Network” (page 184).

Procedure 12.3 *Configuring a USB or SCSI Scanner*

- 1 Connect your USB or SCSI scanner to your computer and turn it on.
- 2 Start YaST and select *Hardware > Scanner*. YaST builds the scanner database and tries to detect your scanner model automatically.

If a USB or SCSI scanner is not properly detected, try *Other > Restart Detection*.

- 3 To activate the scanner select it from the list of detected scanners and click *Edit*.
- 4 Choose your model from the list and click *Next* and *Finish*.
- 5 Use *Other > Test* to make sure you have chosen the correct driver.
- 6 Leave the configuration screen with *OK*.

12.3.1 Configuring an HP All-In-One Device

An HP All-In-One device can be configured with YaST even if it is connected to the parallel port or is made available via the network. If you own a USB HP All-In-One device, start configuring as described in Procedure 12.3, “Configuring a USB or SCSI Scanner” (page 183). If it is detected properly and the *Test* succeeds, it is ready to use.

If your USB device is not properly detected, or your HP All-In-One device is connected to the parallel port or the network, run the HP Device Manager:

- 1 Start YaST and select *Hardware > Scanner*. YaST loads the scanner database.

- 2 Start the HP Device Manager with *Other > Run hp-setup* and follow the on-screen instructions. After having finished the HP Device Manager, the YaST scanner module automatically restarts the auto detection.
- 3 Test it by choosing *Other > Test*.
- 4 Leave the configuration screen with *OK*.

12.3.2 Sharing a Scanner over the Network

openSUSE allows the sharing of a scanner over the network. To do so, configure your scanner as follows:

- 1 Configure the scanner as described in Section 12.3, “Setting Up a Scanner” (page 182).
- 2 Choose *Other > Scanning via Network*.
- 3 Enter the hostnames of the clients (separated by a comma) that should be allowed to use the scanner under *Server Settings > Permitted Clients for saned* and leave the configuration dialog with *OK*.

12.3.3 Scanning over the Network

To use a scanner that is shared over the network, proceed as follows:

- 1 Start YaST and select *Hardware > Scanner*.
- 2 Open the network scanner configuration menu by *Other > Scanning via Network*.
- 3 Enter the hostname of the machine the scanner is connected to under *Client Settings > Servers Used for the net Metadriver*.
- 4 Leave with *OK*. The network scanner is now listed in the Scanner Configuration window and is ready to use.

Printer Operation

openSUSE® supports printing with many types of printers, including remote network printers. Printers can be configured manually or with YaST. For configuration instructions, refer to Section 13.4, “Setting Up a Printer” (page 188). Both graphical and command line utilities are available for starting and managing print jobs. If your printer does not work as expected, refer to Section 13.8, “Troubleshooting” (page 199).

CUPS (Common Unix Printing System) is the standard print system in openSUSE.

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

PostScript Printers

PostScript is the printer language in which most print jobs in Linux and Unix are generated and processed by the internal print system. If PostScript documents can be processed directly by the printer and do not need to be converted in additional stages in the print system, the number of potential error sources is reduced.

Standard Printers (Languages Like PCL and ESC/P)

Although these printer languages are quite old, they are still undergoing expansion to address new features in printers. In the case of known printer languages, the print system can convert PostScript jobs to the respective printer language with the help of Ghostscript. This processing stage is referred to as interpreting.

The best-known languages are PCL (which is mostly used by HP printers and their clones) and ESC/P (which is used by Epson printers). These printer languages are usually supported by Linux and produce an adequate print result. Linux may not be able to address some special printer functions. Except for HP developing HPLIP (HP Linux Imaging and Printing), there are currently no printer manufacturers who develop Linux drivers and make them available to Linux distributors under an open source license.

Proprietary Printers (Also Called GDI Printers)

These printers do not support any of the common printer languages. They use their own undocumented printer languages, which are subject to change when a new edition of a model is released. Usually only Windows drivers are available for these printers. See Section 13.8.1, “Printers without Standard Printer Language Support” (page 199) for more information.

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

<http://www.linuxfoundation.org/OpenPrinting/>

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

<http://pages.cs.wisc.edu/~ghost/>

The Ghostscript Web page.

```
/usr/share/doc/packages/ghostscript-library/catalog-  
log.devices
```

List of included drivers.

13.1 The Workflow of the Printing System

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

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

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

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

13.2 Methods and Protocols for Connecting Printers

There are various possibilities for connecting a printer to the system. The configuration of the CUPS print system does not distinguish between a local printer and a printer connected to the system over the network. For more information about the printer connection, read the article *CUPS in a Nutshell* in the Support Database at http://old-en.opensuse.org/SDB:CUPS_in_a_Nutshell.

WARNING: Changing Cable Connections in a Running System

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

13.3 Installing the Software

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

To configure a PostScript printer, the best approach is to get a suitable PPD file. Many PPD files are available in the package `manufacturer-PPDs`, which is automatically installed within the scope of the standard installation. See Section 13.7.2, “PPD Files in Various Packages” (page 197) and Section 13.8.2, “No Suitable PPD File Available for a PostScript Printer” (page 200).

New PPD files can be stored in the directory `/usr/share/cups/model/` or added to the print system with YaST as described in Section “Adding Drivers with YaST” (page 190). Subsequently, the PPD file can be selected during the printer setup.

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

13.4 Setting Up a Printer

YaST can be used to configure a local printer that is directly connected to your machine (normally with USB or parallel port) and to set up printing with network printers. It is also possible to share printers over the network. Further information about printing (general information, technical details, and troubleshooting) is available in Chapter 13, *Printer Operation* (page 185).

In YaST, click *Hardware > Printer* to start the printer module. By default it opens in the *Printer Configurations* view, displaying a list of all printers that are available and configured. This is especially useful when having access to a lot of printers via the network. From here you can also *Print a Test Page* and configure local printers.

13.4.1 Configuring Local Printers

Usually a local USB printer is automatically detected. There are two possible reasons why a USB printer is not automatically detected:

- The USB printer is switched off.
- The communication between printer and computer is not possible. Check the cable and the plugs to make sure that the printer is properly connected. If this is the case, the problem may not be printer-related, but rather a USB related problem.

Configuring a printer is basically a three-step process. First specify the connection type, then choose a driver, and name the printing queue for this setup.

For many printer models, several drivers are available. When configuring the printer, YaST defaults to the one marked *recommended* as a general rule. Normally it is not necessary to change the driver—the *recommended* one should produce the best results. However, if you want a color printer to print only in black and white, it is most convenient to use a driver that does not support color printing, for example. If you experience performance problems with a PostScript printer when printing graphics, it may help to switch from a PostScript driver to a PCL driver (provided your printer understands PCL).

If no driver for your printer is listed, you can try to select a generic driver with an appropriate standard language from the list. Refer to your printer's documentation to find out which language (the set of commands controlling the printer) your printer understands. If this does not work, refer to Section “Adding Drivers with YaST” (page 190) for another possible solution.

A printer is never used directly, but always through a print queue. This ensures that simultaneous jobs can be queued and processed one after the other. Each printer queue is assigned to a specific driver, and a printer can have multiple queues. This makes it possible to set up a second queue on a color printer that prints black and white only, for example. Refer to Section 13.1, “The Workflow of the Printing System” (page 186) for more information about print queues.

Procedure 13.1 *Adding a New Local Printer*

- 1 Start the YaST printer module with *Hardware > Printer*
- 2 Click *Add* in the *Printer Configurations* screen

- 3 If your printer is already listed under *Specify the Connection*, proceed with the next step. Otherwise, try to *Detect More* or start the *Connection Wizard*.
- 4 Enter the vendor name and the model name into the input box under *Find and Assign a Driver* and click *Search for*.
- 5 Choose the driver marked as recommended that best matches your printer. If no suitable drivers is displayed
 - 5a check your search term
 - 5b broaden your search by clicking *Find More*
 - 5c add a driver as described in Section “Adding Drivers with YaST” (page 190)
- 6 Specify the *Default paper size*
- 7 Enter a unique name for the printer queue in the *Set Arbitrary Name* field.
- 8 The printer is now configured with the default settings and ready to use. Click *OK* to return to the *Printer Configurations* view. The newly configured printer is now visible in the printers list.

Adding Drivers with YaST

If no suitable driver is available in the *Find and Assign a Driver* dialog when adding a new printer, no PPD (PostScript Printer Description) file for your model is available. For more information about PPD files, refer to Section 13.3, “Installing the Software” (page 188).

Get PPD files directly from your printer vendor or from the driver CD of a PostScript printer. For details, see Section 13.8.2, “No Suitable PPD File Available for a PostScript Printer” (page 200). Conversely, you can also find PPD files at <http://www.linuxfoundation.org/en/OpenPrinting/>, the “OpenPrinting.org printer database”. When downloading PPD files from OpenPrinting.org, keep in mind that it always shows the latest Linux support status, which is not necessarily met by openSUSE.

Procedure 13.2 *Adding a PPD file*

- 1 Start the YaST printer module with *Hardware > Printer*

- 2 Click *Add* in the *Printer Configurations* screen
- 3 Click *Driver Packages* in the *Find and Assign a Driver* section
- 4 Enter the full path to the PPD file into the input box under *Make a Printer Description File Available* or choose the file from a dialog box by clicking on *Browse*
- 5 Click *OK* to return to the *Add New Printer Configuration* screen.
- 6 In order to directly use this PPD file, proceed as described in Procedure 13.1, “Adding a New Local Printer” (page 189). Otherwise, click *Cancel*.

Editing a Local Printer Configuration

By editing an existing configuration for a local printer you can not only change basic settings as connection type and driver, but also adjust the default settings for paper size, resolution, media source, etc. You can change the identifier of the printer by altering the printer descriptions.

Procedure 13.3 *Editing a Local Printer*

- 1 Start the YaST printer module with *Hardware > Printer*
- 2 In the *Printer Configurations* screen, choose a local printer from the list and click *Edit*.
- 3 Change the connection type or the driver as described in Procedure 13.1, “Adding a New Local Printer” (page 189). This should only be necessary in case you have problems with the current configuration.
- 4 Make this printer the default by checking *Default Printer*.
- 5 Adjust the default settings by clicking *All Options for the Current Driver*. To change a setting, expand the list of options by clicking the relative + sign. Change the default by clicking on an option. Apply your changes by clicking *OK*

13.4.2 Configuring Printing via the Network with YaST

Network printers are not detected automatically. They must be configured manually using the YaST printer module. Depending on your network setup, you can print to a print server (CUPS, LPD, SMB, or IPX) or directly to a network printer (preferably via TCP). Access the configuration view for network printing by choosing *Printing via Network* from the left pane in the YaST printer module.

Using CUPS

In a Linux environment CUPS is usually used to print via the network. The simplest setup is to only print via a single CUPS server which can directly be accessed by all clients. Printing via more than one CUPS server requires a running local CUPS daemon that communicates with the remote CUPS servers.

Procedure 13.4 *Printing via a Single CUPS server*

- 1 Start the YaST printer module with *Hardware > Printer*
- 2 Launch the *Print via Network* screen from the left pane.
- 3 Check *Do All Your Printing Directly via One Single CUPS Server* and specify the name or IP address of the server.
- 4 Click *Test Server* to make sure you have chosen the correct name or IP address.
- 5 Click OK to return to the *Printer Configurations* screen. All printers available via the CUPS server are now listed.

Procedure 13.5 *Printing via multiple CUPS servers*

- 1 Start the YaST printer module with *Hardware > Printer*
- 2 Launch the *Print via Network* screen from the left pane.
- 3 Check *Accept Printer Announcements from CUPS Servers*
- 4 Specify which servers to use under *General Settings*. You may accept connections from all networks available, from the local network, or from specific hosts. If you choose the latter option, you need to specify the hostnames or IP addresses, as well.
- 5 Confirm by clicking *OK* and then *Yes* when asked to start a local CUPS server. After the server has started you will return to the *Printer Configurations* screen.

Click *Refresh list* to see the printers detected by now. Click this button again, in case more printer are to be available.

Using Print Servers other than CUPS

If your network offers print services via print servers other than CUPS, start the YaST printer module with *Hardware > Printer* and launch the *Print via Network* screen from the left pane. Start the *Connection Wizard* and choose the appropriate *Connection* type. Ask your network administrator for details on configuring a network printer in your environment.

13.4.3 Sharing Printers Over the Network

Printers managed by a local CUPS daemon can be shared over the network and so turn your machine into a CUPS server. Usually you share a printer by enabling CUPS' so-called "browsing mode". If browsing is enabled, the local printer queues are made available on the network for listening to remote CUPS daemons. It is also possible to set up a dedicated CUPS server that manages all printing queues and can directly be accessed by remote clients. In this case it is not necessary to enable browsing.

Procedure 13.6 *Sharing Printers*

- 1** Start the YaST printer module with *Hardware > Printer*
- 2** Launch the *Share Printers* screen from the left pane.
- 3** Select *Allow Remote Access*. For more detailed configuration, additional options are available:
 - Check *For computers within the local network* and enable browsing mode by also checking *Publish printers by default within the local network*.
 - *Add* the network interface to be used by the CUPS server. If you want to share your printers via specified network interfaces, add those in the input box below.
 - In case you like to restrict access to your CUPS server to certain networks or IP addresses, specify these via the two input boxes.
- 4** Click *OK* to restart the CUPS server and return to the *Printer Configurations* screen.

5 Regarding CUPS and firewall settings, see http://en.opensuse.org/SDB:CUPS_and_SANE_Firewall_settings.

13.5 Network Printers

A network printer can support various protocols, some of them even concurrently. Although most of the supported protocols are standardized, some manufacturers modify the standard. Manufacturers then provide drivers for only a few operating systems. Unfortunately, Linux drivers are rarely provided. The current situation is such that you cannot act on the assumption that every protocol works smoothly in Linux. Therefore, you may have to experiment with various options to achieve a functional configuration.

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

`socket`

Socket refers to a connection in which the plain print data is sent directly to a TCP socket. Some of the socket port numbers that are commonly used are 9100 or 35. The device URI (uniform resource identifier) syntax is: `socket://IP.of.the.printer:port`, for example: `socket://192.168.2.202:9100/`.

`LPD` (Line Printer Daemon)

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

`IPP` (Internet Printing Protocol)

`IPP` is a relatively new protocol (1999) based on the `HTTP` protocol. With `IPP`, more job-related data is transmitted than with the other protocols. CUPS uses `IPP` for internal data transmission. The name of the print queue is necessary to configure `IPP` correctly. The port number for `IPP` is 631. Example device URIs are `ipp://192.168.2.202/ps` and `ipp://192.168.2.202/printers/ps`.

SMB (Windows Share)

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

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

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

13.5.1 Configuring CUPS with Command Line Tools

CUPS can be configured with command line tools like `lpinfo`, `lpadmin` and `lpoptions`. You need a device URI consisting of a back-end, such as `parallel`, and parameters. To determine valid device URIs on your system use the command `lpinfo -v | grep ":/":`

```
# lpinfo -v | grep ":/"  
direct usb://ACME/FunPrinter%20XL  
direct parallel:/dev/lp0
```

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

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

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

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

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

The following example configures a network printer:

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

For more options of `lpadmin`, see the man page of `lpadmin(8)`.

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

1 First, list all options:

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

Example:

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

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

2 Change the option with `lpadmin`:

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

3 Check the new setting:

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

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

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

13.6 Printing from the Command Line

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

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

13.7 Special Features in openSUSE

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

13.7.1 CUPS and Firewall

After having performed a default installation of openSUSE, SuSEfirewall2 is active and the network interfaces are configured to be in the `External` Zone which blocks incoming traffic. More information about the SuSEfirewall2 configuration is available in Section “SuSEfirewall2” (Chapter 14, *Masquerading and Firewalls*, ↑Security Guide) and at http://en.opensuse.org/SDB:CUPS_and_SANE_Firewall_settings .

CUPS Client

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

CUPS Server

If the CUPS server is part of a trusted network environment protected by a firewall, the network interface should be configured to be in the `Internal` Zone of the firewall. It is not recommended to set up a CUPS server in an untrusted network environment unless you take care that it is protected by special firewall rules and secure settings in the CUPS configuration.

13.7.2 PPD Files in Various Packages

The YaST printer configuration sets up the queues for CUPS using the PPD files installed in `/usr/share/cups/model` . To find the suitable PPD files for the printer model, YaST compares the vendor and model determined during hardware detection with the vendors and models in all PPD files. For this purpose, the YaST printer configuration generates a database from the vendor and model information extracted from the PPD files.

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

CUPS PPD Files in the `cups` Package

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

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

PPD Files in the `cups-drivers` Package

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

YaST generally prefers a `manufacturer-PPD` file. However, when no suitable `manufacturer-PPD` file exists, a Foomatic PPD file with the entry `*NickName: ... Foomatic ...` (recommended) is selected.

Gutenprint PPD Files in the `gutenprint` Package

Instead of `foomatic-rip`, the CUPS filter `rastertogutenprint` from Gutenprint (formerly known as GIMP-Print) can be used for many non-PostScript printers. This filter and suitable Gutenprint PPD files are available in the `gutenprint` package. The Gutenprint PPD files are located in `/usr/share/cups/model/gutenprint/` and have the entries `*NickName: ... CUPS +Gutenprint` and `*cupsFilter: ... rastertogutenprint`.

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

The `manufacturer-PPDs` package contains PPD files from printer manufacturers that are released under a sufficiently liberal license. PostScript printers should be configured with the suitable PPD file of the printer manufacturer, because this file enables the use of all functions of the PostScript printer. YaST prefers a PPD file from the `manufacturer-PPDs`. YaST cannot use a PPD file from the `manufacturer-PPDs` package if the model name does not match. This may happen if the `manufacturer-PPDs` package contains only one PPD file for similar models, like Funprinter 12xx series. In this case, select the respective PPD file manually in YaST.

13.8 Troubleshooting

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

13.8.1 Printers without Standard Printer Language Support

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

Some GDI printers can be switched to operate either in GDI mode or in one of the standard printer languages. See the manual of the printer whether this is possible. Some models require special Windows software to do the switch (note that the Windows printer driver may always switch the printer back into GDI mode when printing

from Windows). For other GDI printers there are extension modules for a standard printer language available.

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

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

13.8.2 No Suitable PPD File Available for a PostScript Printer

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

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

13.8.3 Parallel Ports

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

- I/O address: 378 (hexadecimal)
- Interrupt: irrelevant

- **Mode:** Normal, SPP, or Output Only
- **DMA:** disabled

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

If interrupt 7 is free, it can be activated with the entry shown in Example 13.1, “`/etc/modprobe.conf`: Interrupt Mode for the First Parallel Port” (page 201). Before activating the interrupt mode, check the file `/proc/interrupts` to see which interrupts are already in use. Only the interrupts currently being used are displayed. This may change depending on which hardware components are active. The interrupt for the parallel port must not be used by any other device. If you are not sure, use the polling mode with `irq=none`.

Example 13.1 */etc/modprobe.conf: Interrupt Mode for the First Parallel Port*

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

13.8.4 Network Printer Connections

Identifying Network Problems

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

Checking the TCP/IP Network

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

Checking a Remote `lpd`

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

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

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

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

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

If `lpd` does not respond, it may not be active or there may be basic network problems. If `lpd` responds, the response should show why printing is not possible on the queue on `host`. If you receive a response like that shown in Example 13.2, “Error Message from `lpd`” (page 202), the problem is caused by the remote `lpd`.

Example 13.2 *Error Message from `lpd`*

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

Checking a Remote `cupsd`

A CUPS network server can broadcast its queues by default every 30 seconds on UDP port 631. Accordingly, the following command can be used to test whether there is a broadcasting CUPS network server in the network. Make sure to stop your local CUPS daemon before executing the command.

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

If a broadcasting CUPS network server exists, the output appears as shown in Example 13.3, “Broadcast from the CUPS Network Server” (page 202).

Example 13.3 *Broadcast from the CUPS Network Server*

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

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

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

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

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

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

Troubleshooting a Network Printer or Print Server Box

Spoolers running in a print server box sometimes cause problems when they have to deal with multiple print jobs. Since this is caused by the spooler in the print server box, there no way to resolve this issue. As a work-around, circumvent the spooler in the print server box by addressing the printer connected to the print server box directly with the TCP socket. See Section 13.5, “Network Printers” (page 194).

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

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

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

Enter a command like

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

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

13.8.5 Defective Printouts without Error Message

For the print system, the print job is completed when the CUPS back-end completes the data transfer to the recipient (printer). If further processing on the recipient fails

(for example, if the printer is not able to print the printer-specific data) the print system does not notice this. If the printer is not able to print the printer-specific data, select a PPD file that is more suitable for the printer.

13.8.6 Disabled Queues

If the data transfer to the recipient fails entirely after several attempts, the CUPS back-end, such as `USB` or `socket`, reports an error to the print system (to `cupsd`). The back-end determines how many unsuccessful attempts are appropriate until the data transfer is reported as impossible. As further attempts would be in vain, `cupsd` disables printing for the respective queue. After eliminating the cause of the problem, the system administrator must reenables printing with the command `cupsenable`.

13.8.7 CUPS Browsing: Deleting Print Jobs

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

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

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

13.8.8 Defective Print Jobs and Data Transfer Errors

If you switch the printer off or shut down the computer during the printing process, print jobs remain in the queue. Printing resumes when the computer (or the printer) is

switched back on. Defective print jobs must be removed from the queue with `cancel`.

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

- 1 To stop printing, remove all paper from ink jet printers or open the paper trays of laser printers. High-quality printers have a button for canceling the current print-out.
- 2 The print job may still be in the queue, because jobs are only removed after they are sent completely to the printer. Use `lpstat -o` or `lpstat -h cups.example.com -o` to check which queue is currently printing. Delete the print job with `cancel queue-jobnumber` or `cancel -h cups.example.com queue-jobnumber`.
- 3 Some data may still be transferred to the printer even though the print job has been deleted from the queue. Check if a CUPS back-end process is still running for the respective queue and terminate it. For example, for a printer connected to the parallel port, the command `fuser -k /dev/lp0` can be used to terminate all processes that are still accessing the printer (more precisely: the parallel port).
- 4 Reset the printer completely by switching it off for some time. Then insert the paper and turn on the printer.

13.8.9 Debugging the CUPS Print System

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

- 1 Set `LogLevel debug` in `/etc/cups/cupsd.conf`.
- 2 Stop `cupsd`.
- 3 Remove `/var/log/cups/error_log*` to avoid having to search through very large log files.
- 4 Start `cupsd`.
- 5 Repeat the action that led to the problem.

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

13.8.10 For More Information

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

Installing and Configuring Fonts for the Graphical User Interface

14

The installation of additional fonts in openSUSE® is very easy. Simply copy the fonts to any directory located in the X11 font path . To the enable use of the fonts, the installation directory should be a subdirectory of the directories configured in `/etc/fonts/fonts.conf` or included into this file with `/etc/fonts/suse-font-dirs.conf` .

The following is an excerpt from `/etc/fonts/fonts.conf` . This file is the standard configuration file that should be appropriate for most configurations. It also defines the included directory `/etc/fonts/conf.d` . In this directory, all files or symbolic links starting with a two digit number are loaded by fontconfig. For a more detailed explanation of this functionality, have a look at `/etc/fonts/conf.d/README` .

```
<!-- Font directory list -->
<dir>/usr/share/fonts</dir>
<dir>/usr/X11R6/lib/X11/fonts</dir>
<dir>/opt/kde3/share/fonts</dir>
<dir>/usr/local/share/fonts</dir>
<dir>~/.fonts</dir>
```

`/etc/fonts/suse-font-dirs.conf` is automatically generated to pull in fonts that ship with (mostly third party) applications like OpenOffice.org, Java or Adobe Acrobat Reader. A typical entry would look like the following:

```
<dir>/usr/lib/Adobe/Reader9/Resource/Font</dir>
<dir>/usr/lib/Adobe/Reader9/Resource/Font/PFM</dir>
```

14.1 Adding Fonts

To install additional fonts systemwide, manually copy the font files to a suitable directory (as root), such as `/usr/share/fonts/truetype` . Alternatively, the task can be performed with the KDE font installer in the KDE Personal Settings. The result is the same.

Instead of copying the actual fonts, you can also create symbolic links. For example, you may want to do this if you have licensed fonts on a mounted Windows partition and want to use them. Subsequently, run `SuSEconfig --module fonts`.

`SuSEconfig --module fonts` executes the script `/usr/sbin/fonts-config`, which handles the font configuration. For more information on this script, refer to its manual page (`man fonts-config`).

The procedure is the same for bitmap fonts, TrueType and OpenType fonts, and Type 1 (PostScript) fonts. All these font types can be installed into any directory known to `fonts-config`.

Upgrading the System and System Changes

You can upgrade an existing system without completely reinstalling it. There are two types of renewing the system or parts of it: *updating individual software packages* and *upgrading the entire system*. Updating individual packages is covered in Chapter 4, *Installing or Removing Software* (page 87) and Chapter 5, *YaST Online Update* (page 111). Two ways to upgrade the system are discussed in the following sections— see Section 15.1.3, “Upgrading with YaST” (page 210) and Section 15.1.4, “Distribution Upgrade with zypper” (page 212).

15.1 Upgrading the System

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

15.1.1 Preparations

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

Before starting your update, make note of the root partition. The command `df /` lists the device name of the root partition. In Example 15.1, “List with `df -h`” (page 210), the root partition to write down is `/dev/sda3` (mounted as `/`).

Example 15.1 *List with `df -h`*

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

15.1.2 Possible Problems

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

Checking `passwd` and `group` in `/etc`

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

PostgreSQL

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

15.1.3 Upgrading with YaST

Following the preparation procedure outlined in Section 15.1.1, “Preparations” (page 209), you can now upgrade your system:

- 1 Boot the system as for the installation, described in Section 1.4, “System Start-Up for Installation” (page 8). In YaST, choose a language and select *Update* in

the *Installation Mode* dialog. Do not select *New Installation*. Also add repositories to make sure to get all available software updated whenever possible. Find more information about installation repositories in Section 1.7.1, “Add-On Products” (page 14).

- 2 YaST determines if there are multiple root partitions. If there is only one, continue with the next step. If there are several, select the right partition and confirm with *Next* (`/dev/sda3` was selected in the example in Section 15.1.1, “Preparations” (page 209)). YaST reads the old `fstab` on this partition to analyze and mount the file systems listed there.

WARNING: Persistent Device Names

All entries in `/etc/fstab` that specify partitions to be mounted using the kernel-device name must be changed to any of the other supported methods prior to performing an update. Kernel-device names are not persistent and are therefore unreliable for use during the update process. This can be done using the YaST System Partitioner by changing the method used in the `fstab` options settings.

- 3 Check the previously used repositories, if there are any. Enable all the repositories you still want to use and from where you want to update third-party software. Click the *Toggle Status* for every list item, if appropriate.
- 4 If you added repositories during the upgrade procedure as recommended above, you now can activate those you are actually interested in.
- 5 In the *Installation Settings* dialog, adjust the settings according to your requirements. Normally, you can leave the default settings untouched. If you intend to enhance your system, however, check the packages and patterns offered in the *Packages* and *Update Options* submenus, or add support for additional languages.

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

- 6 Confirm the upgrade by clicking *Start Update*.

Once the basic upgrade installation is finished, YaST reboots the system. Finally, YaST updates the remaining software, if any and displays the release notes, if wanted.

15.1.4 Distribution Upgrade with zypper

With the `zypper` command line utility you can upgrade to the next version of the distribution. Most importantly, you can initiate the system upgrade process from within the running system.

This feature is attractive for advanced users who want to run remote upgrades or upgrades on many similarly configured systems. Inexperienced users will prefer the upgrade with YaST using a boot medium as described in Section 15.1.3, “Upgrading with YaST” (page 210).

Before Starting the Upgrade with zypper

To avoid unexpected errors during the upgrade process using `zypper`, minimize risky constellations.

Upgrade from the previous version (e.g., 11.3) to this version (11.4)—do not skip any minor version inbetween (this means, do not upgrade from 11.2 or earlier to 11.4 in one go). Make sure all available 11.3 online updates are successfully applied.

Close as many applications and unneeded services as possible and log out all regular users.

Disable third party or openSUSE Build Service repositories before starting the upgrade, or lower the priority of these repositories to make sure packages from the default system repositories will get preference. Enable them again after the upgrade and edit their version string to match the version number of the distribution of the upgraded now running system.

For more information, see http://en.opensuse.org/SDB:System_upgrade.

The Upgrade Procedure

WARNING: Check Your System Backup

Before actually starting the upgrade procedure, check that your system backup is up-to-date and restorable. This is especially important because you must enter many of the following steps manually.

- 1 Run the online update to make sure the software management stack is up-to-date. For more information, see Chapter 5, *YaST Online Update* (page 111).
- 2 Configure the repositories you want to use as an update source. Getting this right is essential. Either use YaST (see Section 4.4, “Managing Software Repositories and Services” (page 101)) or `zypper` (see Section 8.1, “Using Zypper” (page 125)). The name of the repositories as used in the following steps could vary a little bit depending on your customizations.

To view your current repositories enter:

```
zypper lr -u
```

TIP: zypper command names

`zypper` supports long and short command names. For example, you can abbreviate `zypper install` as `zypper in`. In the following text, the short variant is used.

- 2a Increase the version number of the system repositories from 11.3 to 11.4; add the new 11.4 repositories with commands such as:

```
server=http://download.opensuse.org
new_ver=11.4
zypper ar $server/distribution/$new_ver/repo/oss/ openSUSE-
$new_ver-Oss
zypper ar $server/update/$new_ver/ openSUSE-$new_ver-Update
```

And remove the old repositories:

```
old_ver=11.3
zypper rr openSUSE-$old_ver-Oss
zypper rr openSUSE-$old_ver-Update
```

- 2b Disable third party repositories or other openSUSE Build Service repositories, because `zypper dup` is guaranteed to work with the default repositories only (replace `repo-alias` with the name of the repository you want to disable):

```
zypper mr -d repo-alias
```

Alternatively, you can lower the priority of these repositories.

NOTE: Handling of Unresolved Dependencies

`zypper dup` will remove all packages having unresolved dependencies, but it keeps packages of disabled repositories as long as their dependencies are satisfied.

`zypper dup` ensures that all installed packages come from one of the available repositories. It does not consider the version, architecture, or vendor of the installed packages; thus it emulates a fresh installation. Packages that are no longer available in the repositories are considered orphaned. Such packages get uninstalled if their dependencies can not be satisfied. If they can be satisfied, such packages stay installed.

2c Once done, check your repository configuration with:

```
zypper lr -d
```

- 3** Refresh local metadata and repository contents with `zypper ref`.
- 4** Pull in `zypper` and the package management stack from the 11.4 repository with `zypper up zypper`.
- 5** Run the actual distribution upgrade with `zypper dup`. You are asked to confirm the license.
- 6** Perform basic system configuration with `SuSEconfig`.
- 7** Reboot the system with `shutdown -r now`.

15.1.5 Updating Individual Packages

Regardless of your overall updated environment, you can always update individual packages. From this point on, however, it is your responsibility to ensure that your system remains consistent.

Use the YaST software management tool to update packages as described in Chapter 4, *Installing or Removing Software* (page 87). Select components from the YaST package selection list according to your needs. If a newer version of a package exists, the version numbers of the installed and the available versions are listed in blue col-

or in the *Installed (Available)* column. If you select a package essential for the overall operation of the system, YaST issues a warning. Such packages should be updated only in the update mode. For example, many packages contain *shared libraries*. Updating these programs and applications in the running system may lead to system instability.

15.2 Software Changes

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

Problems and special issues of the various versions are published online as they are identified. See the links listed below. Important updates of individual packages can be accessed using the YaST Online Update. For more information, see Chapter 5, *YaST Online Update* (page 111).

Refer to the Product highlights (http://en.opensuse.org/Product_highlights) and the Bugs article in the openSUSE wiki at http://en.opensuse.org/openSUSE:Most_annoying_bugs for information about recent changes and issues.

Part IV. System

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

16

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.

16.1 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 medi-

ates 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).

16.2 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.

16.3 Software Compilation on Biarch Platforms

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

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

The following example refers to an `x86_64` system with `x86` as the second architecture.

1 Use the 32-bit compiler:

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

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

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

3 Set the assembler to generate 32-bit objects:

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

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

```
LDLFLAGS="-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"  
LD_FLAGS="-L/usr/lib;"  
./configure --prefix=/usr --libdir=/usr/lib --x-libraries=/usr/lib  
make  
make install
```

16.4 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 Novell to make sure that the 64-bit version of the kernel-loadable module and the 32-bit compiled version of the kernel API are available for this module.

Booting and Configuring a Linux System

17

Booting a Linux system involves different components. The hardware itself is initialized by the BIOS, which starts the Kernel by means of a boot loader. After this point, the boot process with init and the runlevels is completely controlled by the operating system. The runlevel concept enables you to maintain setups for everyday usage as well as to perform maintenance tasks on the system.

17.1 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** After turning on the computer, the BIOS initializes the screen and keyboard and tests the main memory. Up to this stage, the machine does not access any mass storage media. Subsequently, the information about the current date, time, and the most important peripherals are loaded from the CMOS values. When the first hard disk and its geometry are recognized, the system control passes from the BIOS to the boot loader.
2. **Boot Loader** The first physical 512-byte data sector of the first hard disk is loaded into the main memory and the *boot loader* that resides at the beginning of this sector takes over. The commands executed by the boot loader determine the remaining part of the boot process. Therefore, the first 512 bytes on the first hard disk are referred to as the *Master Boot Record* (MBR). The boot loader then passes control to the actual operating system, in this case, the Linux Kernel. More infor-

mation about GRUB, the Linux boot loader, can be found in Chapter 18, *The Boot Loader GRUB* (page 239).

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 17.1.1, “`initramfs`” (page 224).
4. **`init` on `initramfs`** This program performs all actions needed to mount the proper root file system, like providing Kernel functionality for the needed file system and device drivers for mass storage controllers with `udev`. After the root file system has been found, it is checked for errors and mounted. If this is successful, the `initramfs` is cleaned and the `init` program on the root file system is executed. For more information about `init`, refer to Section 17.1.2, “`init` on `initramfs`” (page 225). Find more information about `udev` in Chapter 20, *Dynamic Kernel Device Management with `udev`* (page 273).
5. **`init`** `init` handles the actual booting of the system through several different levels providing different functionality. `init` is described in Section 17.2, “The `init` Process” (page 227).

17.1.1 `initramfs`

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

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

process, after changing the root file system, it is necessary to regenerate the devices. This is done by `boot.udev` with the command `udevtrigger`.

If you need to change hardware (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 in the same way as with its predecessor, `initrd`—by calling `mkinitrd`. Calling `mkinitrd` without any argument creates an `initramfs`. Calling `mkinitrd -R` creates an `initrd`. In openSUSE®, the modules to load are specified by the variable `INITRD_MODULES` in `/etc/sysconfig/kernel`. After installation, this variable is automatically set to the correct value. The modules are loaded in exactly the order in which they appear in `INITRD_MODULES`. This is only important if you rely on the correct setting of the device files `/dev/sd?`. However, in current systems you also may use the device files below `/dev/disk/` that are sorted in several subdirectories, named by `-id`, `by-path` and `by-uuid`, and always represent the same disk. This is also possible at install time by specifying the respective mount option.

IMPORTANT: Updating `initramfs` or `initrd`

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

17.1.2 init on `initramfs`

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

Loading Kernel Modules

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

Providing Block Special Files

For each loaded module, the Kernel generates device events. `udev` handles these events and generates the required block special files on a RAM file system in /

`dev`. Without those special files, the file system and other devices would not be accessible.

Managing RAID and LVM Setups

If you configured your system to hold the root file system under RAID or LVM, `init` sets up LVM or RAID to enable access to the root file system later. Find information about RAID and LVM in Chapter 3, *Advanced Disk Setup* (page 67).

Managing Network Configuration

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

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

Finding the Installation Medium

As you start the installation process, your machine loads an installation Kernel and a special `initrd` with the YaST installer on the installation medium. The YaST installer, which is run in a RAM file system, needs to have information about the location of the installation medium to access it and install the operating system.

Initiating Hardware Recognition and Loading Appropriate Kernel Modules

As mentioned in Section 17.1.1, “`initramfs`” (page 224), the boot process starts with a minimum set of drivers that can be used with most hardware configurations. `init` starts an initial hardware scanning process that determines the set of drivers suitable for your hardware configuration. The names of the modules needed for the boot process are written to `INITRD_MODULES` in `/etc/sysconfig/kernel`. These names are used to generate a custom `initramfs` that is needed to boot the system. If the modules are not needed for boot but for coldplug, the modules are written to `/etc/sysconfig/hardware/hwconfig-*`. All devices that are described with configuration files in this directory are initialized in the boot process.

Loading the Installation System or Rescue System

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

Starting YaST

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

17.2 The `init` Process

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

`init` is centrally configured in the `/etc/inittab` file where the *runlevels* are defined (see Section 17.2.1, “Runlevels” (page 227)). The file also specifies which services and daemons are available in each of the runlevels. Depending on the entries in `/etc/inittab`, several scripts are run by `init`. By default, the first script that is started after booting is `/etc/init.d/boot`. Once the system initialization phase is finished, the system changes the runlevel to its default runlevel with the `/etc/init.d/rc` script. For reasons of clarity, these scripts, called *init scripts*, all reside in the directory `/etc/init.d` (see Section 17.2.2, “Init Scripts” (page 229)).

The entire process of starting the system and shutting it down is maintained by `init`. From this point of view, the Kernel can be considered a background process to maintain all other processes and adjust CPU time and hardware access according to requests from other programs.

17.2.1 Runlevels

In Linux, *runlevels* define how the system is started and what services are available in the running system. After booting, the system starts as defined in `/etc/inittab` in the line `initdefault`. Usually this is 3 or 5. See Table 17.1, “Available Runlevels” (page 228). As an alternative, the runlevel can be specified at boot time (by adding the runlevel number at the boot prompt, for instance). Any parameters that are not directly evaluated by the Kernel itself are passed to `init`. To boot into runlevel 3, just add the single number 3 to the boot prompt.

Table 17.1 *Available Runlevels*

Runlevel	Description
0	System halt

IMPORTANT: Avoid Runlevel 2 with a Partition Mounted via NFS

You should not use runlevel 2 if your system mounts a partition like `/usr` via NFS. The system might behave unexpectedly if program files or libraries are missing because the NFS service is not available in runlevel 2 (local multiuser mode without remote network).

To change runlevels while the system is running, enter `telinit` and the corresponding number as an argument. Only the system administrator is allowed to do this. The following list summarizes the most important commands in the runlevel area.

`telinit 1` or `shutdown now`

The system changes to *single user mode*. This mode is used for system maintenance and administration tasks.

`telinit 3`

All essential programs and services (including network) are started and regular users are allowed to log in and work with the system without a graphical environment.

`telinit 5`

The graphical environment is enabled. Usually a display manager like XDM, GDM or KDM is started. If autologin is enabled, the local user is logged in to the preselected window manager (GNOME or KDE or any other window manager).

`telinit 0` or `shutdown -h now`

The system halts.

`telinit 6` or `shutdown -r now`

The system halts then reboots.

Runlevel 5 is the default runlevel in all openSUSE standard installations. Users are prompted for login with a graphical interface or the default user is logged in automatically.

WARNING: Errors in `/etc/inittab` May Result in a Faulty System Boot

If `/etc/inittab` is damaged, the system may not boot properly. Therefore, be extremely careful while editing `/etc/inittab`. Always let `init` reread `/etc/inittab` with the command `telinit q` before rebooting the machine.

Generally, two things happen when you change runlevels. First, stop scripts of the current runlevel are launched, closing down some programs essential for the current runlevel. Then start scripts of the new runlevel are started. Here, in most cases, a number of programs are started. For example, the following occurs when changing from runlevel 3 to 5:

1. The administrator (`root`) requests `init` to change to a different runlevel by entering `telinit 5`.
2. `init` checks the current runlevel (`runlevel`) and determines it should start `/etc/init.d/rc` with the new runlevel as a parameter.
3. Now `rc` calls the stop scripts of the current runlevel for which there is no start script in the new runlevel. In this example, these are all the scripts that reside in `/etc/init.d/rc3.d` (the old runlevel was 3) and start with a `K`. The number following `K` specifies the order to run the scripts with the `stop` parameter, because there are some dependencies to consider.
4. The last things to start are the start scripts of the new runlevel. In this example, these are in `/etc/init.d/rc5.d` and begin with an `S`. Again, the number that follows the `S` determines the sequence in which the scripts are started.

When changing into the same runlevel as the current runlevel, `init` only checks `/etc/inittab` for changes and starts the appropriate steps, for example, for starting a `getty` on another interface. The same functionality may be achieved with the command `telinit q`.

17.2.2 Init Scripts

There are two types of scripts in `/etc/init.d` :

Scripts Executed Directly by init

This is the case only during the boot process or if an immediate system shutdown is initiated (power failure or a user pressing Ctrl + Alt + Del). The execution of these scripts is defined in `/etc/inittab`.

Scripts Executed Indirectly by init

These are run when changing the runlevel and always call the master script `/etc/init.d/rc`, which guarantees the correct order of the relevant scripts.

All scripts are located in `/etc/init.d`. Scripts that are run at boot time are called through symbolic links from `/etc/init.d/boot.d`. Scripts for changing the runlevel are called through symbolic links from one of the subdirectories (`/etc/init.d/rc0.d` to `/etc/init.d/rc6.d`). This is just for reasons of clarity and avoids duplicate scripts if they are used in several runlevels. Because every script can be executed as both a start and a stop script, these scripts must understand the parameters `start` and `stop`. The scripts also understand the `restart`, `reload`, `force-reload`, and `status` options. These different options are explained in Table 17.2, “Possible init Script Options” (page 230). Scripts that are run directly by init do not have these links. They are run independently from the runlevel when needed.

Table 17.2 *Possible init Script Options*

Option	Description
<code>start</code>	Start service.

Links in each runlevel-specific subdirectory make it possible to associate scripts with different runlevels. When installing or uninstalling packages, these links are added and removed with the help of the program `insserv` (or using `/usr/lib/lsb/install_initd`, which is a script calling this program). See the `insserv(8)` man page for details.

All of these settings may also be changed with the help of the YaST module. If you need to check the status on the command line, use the tool `chkconfig`, described in the `chkconfig(8)` man page.

A short introduction to the boot and stop scripts launched first or last, respectively, follows as well as an explanation of the maintaining script.

`boot`

Executed while starting the system directly using `init`. It is independent of the chosen runlevel and is only executed once. Here, the `/proc` and `/dev/pts` file systems are mounted and `blogd` (boot logging daemon) is activated. If the system is booted for the first time after an update or an installation, the initial system configuration is started.

The `blogd` daemon is a service started by `boot` and `rc` before any other one. It is stopped after the actions triggered by these scripts (running a number of subscripts, for example, making block special files available) are completed. `blogd` writes any screen output to the log file `/var/log/boot.msg`, but only if and when `/var` is mounted read-write. Otherwise, `blogd` buffers all screen data until `/var` becomes available. Get further information about `blogd` on the `blogd(8)` man page.

The `boot` script is also responsible for starting all the scripts in `/etc/init.d/boot.d` with names that start with `S`. There, the file systems are checked and loop devices are configured if needed. The system time is also set. If an error occurs while automatically checking and repairing the file system, the system administrator can intervene after first entering the root password. The last executed script is `boot.local`.

`boot.local`

Here, enter additional commands to execute at boot before changing into a runlevel. It can be compared to `AUTOEXEC.BAT` on DOS systems.

`halt`

This script is only executed while changing into runlevel 0 or 6. Here, it is executed either as `halt` or as `reboot`. Whether the system shuts down or reboots depends on how `halt` is called. If special commands are needed during the shutdown, add these to the `halt.local` script.

`rc`

This script calls the appropriate stop scripts of the current runlevel and the start scripts of the newly selected runlevel. Like the `/etc/init.d/boot` script, this script is called from `/etc/inittab` with the desired runlevel as parameter.

You can create your own scripts and easily integrate them into the scheme described above. For instructions about formatting, naming and organizing custom scripts, refer to the specifications of the LSB and to the man pages of `init`, `init.d`, `chk-`

config, and insserv. Additionally consult the man pages of startproc and killproc.

WARNING: Faulty init Scripts May Halt Your System

Faulty init scripts may hang your machine up. Edit such scripts with great care and, if possible, subject them to heavy testing in the multiuser environment. Find useful information about init scripts in Section 17.2.1, “Run-levels” (page 227).

To create a custom init script for a given program or service, use the file `/etc/init.d/skeleton` as a template. Save a copy of this file under the new name and edit the relevant program and filenames, paths and other details as needed. You may also need to enhance the script with your own parts, so the correct actions are triggered by the init procedure.

The `INIT INFO` block at the top is a required part of the script and must be edited. See Example 17.1, “A Minimal INIT INFO Block” (page 232).

Example 17.1 A Minimal INIT INFO Block

```
### BEGIN INIT INFO
# Provides:          FOO
# Required-Start:    $syslog $remote_fs
# Required-Stop:     $syslog $remote_fs
# Default-Start:     3 5
# Default-Stop:      0 1 2 6
# Description:       Start FOO to allow XY and provide YZ
### END INIT INFO
```

In the first line of the `INFO` block, after `Provides:`, specify the name of the program or service controlled by this init script. In the `Required-Start:` and `Required-Stop:` lines, specify all services that need to be still running when the service itself is stopped. This information is used later to generate the numbering of script names, as found in the runlevel directories. After `Default-Start:` and `Default-Stop:`, specify the runlevels in which the service should automatically be started or stopped. Finally, for `Description:`, provide a short description of the service in question.

To create the links from the runlevel directories (`/etc/init.d/rc?.d/`) to the corresponding scripts in `/etc/init.d/`, enter the command `insserv new-script-name`. The `insserv` program evaluates the `INIT INFO` header to create the necessary links for start and stop scripts in the runlevel directories (`/etc/init.d/rc?.d/`). The program also takes care of the correct start and stop

order for each runlevel by including the necessary numbers in the names of these links. If you prefer a graphical tool to create such links, use the runlevel editor provided by YaST, as described in Section 17.2.3, “Configuring System Services (Runlevel) with YaST” (page 233).

If a script already present in `/etc/init.d/` should be integrated into the existing runlevel scheme, create the links in the runlevel directories right away with `insserv` or by enabling the corresponding service in the runlevel editor of YaST. Your changes are applied during the next reboot—the new service is started automatically.

Do not set these links manually. If something is wrong in the `INFO` block, problems will arise when `insserv` is run later for some other service. The manually added service will be removed with the next run of `insserv` for this script.

17.2.3 Configuring System Services (Runlevel) with YaST

After starting this YaST module with *YaST > System > System Services (Runlevel)*, it displays an overview listing all the available services and the current status of each service (disabled or enabled). Decide whether to use the module in *Simple Mode* or in *Expert Mode*. The default *Simple Mode* should be sufficient for most purposes. The left column shows the name of the service, the center column indicates its current status and the right column gives a short description. For the selected service, a more detailed description is provided in the lower part of the window. To enable a service, select it in the table then select *Enable*. The same steps apply to disable a service.

For detailed control over the runlevels in which a service is started or stopped or to change the default runlevel, first select *Expert Mode*. The current default runlevel or “initdefault” (the runlevel into which the system boots by default) is displayed at the top. Normally, the default runlevel of a openSUSE system is runlevel 5 (full multiuser mode with network and X). A suitable alternative might be runlevel 3 (full multiuser mode with network).

This YaST dialog allows the selection of one of the runlevels (as listed in Table 17.1, “Available Runlevels” (page 228)) as the new default. Additionally, use the table in this window to enable or disable individual services and daemons. The table lists the services and daemons available, shows whether they are currently enabled on your system and, if so, for which runlevels. After selecting one of the rows with

the mouse, click the check boxes representing the runlevels (*B*, *0*, *1*, *2*, *3*, *5*, *6*, and *S*) to define the runlevels in which the selected service or daemon should be running. Runlevel 4 is undefined to allow creation of a custom runlevel. A brief description of the currently selected service or daemon is provided below the table overview.

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 17.1 *System Services (Runlevel)*



With *Start*, *Stop*, or *Refresh*, decide whether a service should be activated. *Refresh status* checks the current status. *Set or Reset* lets you select whether to apply your changes to the system or to restore the settings that existed before starting the runlevel editor. Selecting *OK* saves the changed settings to disk.

17.3 System Configuration via /etc/sysconfig

The main configuration of openSUSE is controlled by the configuration files in `/etc/sysconfig`. The individual files in `/etc/sysconfig` are only read by the scripts to which they are relevant. This ensures that network settings, for example, only need to be parsed by network-related scripts.

There are two ways to edit the system configuration. Either use the YaST sysconfig Editor or edit the configuration files manually.

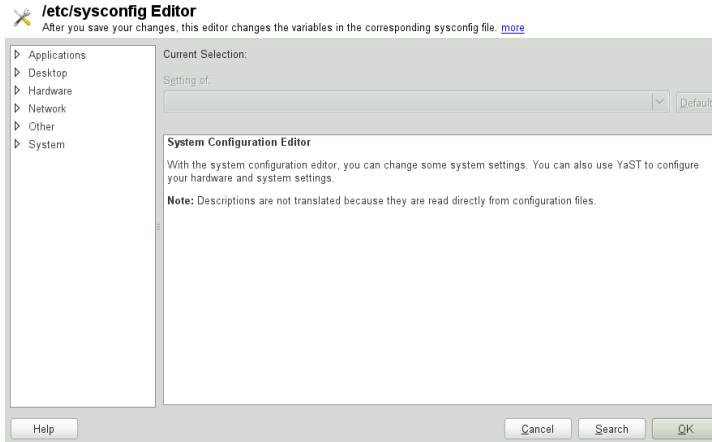
17.3.1 Changing the System Configuration Using the YaST sysconfig Editor

The YaST sysconfig editor provides an easy-to-use front-end for system configuration. Without any knowledge of the actual location of the configuration variable you need to change, you can just use the built-in search function of this module, change the value of the configuration variable as needed and let YaST take care of applying these changes, updating configurations that depend on the values set in `sysconfig` and restarting services.

WARNING: Modifying `/etc/sysconfig/*` Files Can Damage Your Installation

Do not modify the `/etc/sysconfig` files if you lack previous experience and knowledge. It could do considerable damage to your system. The files in `/etc/sysconfig` include a short comment for each variable to explain what effect they actually have.

Figure 17.2 *System Configuration Using the sysconfig Editor*



The YaST sysconfig dialog is split into three parts. The left part of the dialog shows a tree view of all configurable variables. When you select a variable, the right part displays both the current selection and the current setting of this variable. Below, a third window displays a short description of the variable's purpose, possible values, the default value and the actual configuration file from which this variable originates. The dialog also provides information about which configuration script is executed after changing the variable and which new service is started as a result of the change. YaST prompts you to confirm your changes and informs you which scripts will be executed after you leave the dialog by selecting *Finish*. Also select the services and scripts to skip for now, so they are started later. YaST applies all changes automatically and restarts any services involved for your changes to take an effect.

17.3.2 Changing the System Configuration Manually

To manually change the system configuration, proceed as follows

- 1 Become `root`.
- 2 Bring the system into single user mode (runlevel 1) with `telinit 1`.
- 3 Change the configuration files as needed with an editor of your choice.

If you do not use YaST to change the configuration files in `/etc/sysconfig` , make sure that empty variable values are represented by two quotation marks (`KEYTABLE=""`) and that values with blanks in them are enclosed in quotation marks. Values consisting of one word only do not need to be quoted.

- 4 Execute `SuSEconfig` to make sure that the changes take effect.
- 5 Bring your system back to the previous runlevel with a command like `telinit default_runlevel`. Replace `default_runlevel` with the default runlevel of the system. Choose 5 if you want to return to full multiuser with network and X or choose 3 if you prefer to work in full multiuser with network.

This procedure is mainly relevant when changing systemwide settings, such as the network configuration. Small changes should not require going into single user mode, but you may still do so to make absolutely sure that all the programs concerned are correctly restarted.

TIP: Configuring Automated System Configuration

To disable the automated system configuration by `SuSEconfig`, set the variable `ENABLE_SUSECONFIG` in `/etc/sysconfig/suseconfig` to `no`. Do not disable `SuSEconfig` if you want to use the SUSE installation support. It is also possible to disable the autoconfiguration partially.

The Boot Loader 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 17, *Booting and Configuring a Linux System* (page 223). 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).

18.1 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 “Editing Menu Entries during the Boot Procedure” (page 245).

`/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 “Editing Menu Entries during the Boot Procedure” (page 245). 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.

18.1.1 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 18.2, “Configuring the Boot Loader with YaST” (page 250).

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

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

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

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

The device names in GRUB are explained in Section “Naming Conventions for Hard Disks and Partitions” (page 243). This example specifies the first block of the fourth partition of the first hard disk.

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

If the kernel does not have built-in drivers for access to the root partition or a recent Linux system with advanced hotplug features is used, `initrd` must be specified with a separate GRUB command whose only argument is the path to the `initrd` file. Because the loading address of the `initrd` is written into the loaded kernel image, the command `initrd` must follow after the `kernel` command.

The command `root` simplifies the specification of kernel and `initrd` files. The only argument of `root` is a device or a partition. This device is used for all kernel, `initrd`, or other file paths for which no device is explicitly specified until the next `root` command.

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

After writing all menu entries, define one of them as the `default` entry. Otherwise, the first one (entry 0) is used. You can also specify a time-out in seconds after which the default entry should boot. `timeout` and `default` usually precede the menu entries. An example file is described in Section “An Example Menu File” (page 244).

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:

<code>(hd0,0)</code>	first primary partition of the first hard disk
<code>(hd0,1)</code>	second primary partition
<code>(hd0,2)</code>	third primary partition
<code>(hd0,3)</code>	fourth primary partition (usually an extended partition)
<code>(hd0,4)</code>	first logical partition
<code>(hd0,5)</code>	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 18.1.2, “The File `device.map`” (page 246).

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
```

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/message
color white/blue black/light-gray
default 0
timeout 8

title linux
    root (hd0,4)
    kernel /boot/vmlinuz root=/dev/sda7 vga=791 resume=/dev/sda9
    initrd /boot/initrd

title windows
    rootnoverify (hd0,0)
    chainloader +1

title floppy
    rootnoverify (hd0,0)
    chainloader (fd0)+1

title failsafe
    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:

`gfxmenu (hd0,4)/message`

The background image `message` is located in the top directory of the `/dev/sda5` partition.

`color white/blue black/light-gray`

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

default 0

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

timeout 8

After eight seconds without any user input, GRUB automatically boots the default entry. To deactivate automatic boot, delete the `timeout` line. If you set `timeout 0`, GRUB boots the default entry immediately.

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

- The first entry (`title linux`) is responsible for booting openSUSE. The kernel (`linux`) is located in the first logical partition (the boot partition) of the first hard disk. Kernel parameters, such as the root partition and VGA mode, are appended here. The root partition is specified according to the Linux naming convention (`/dev/sda7/`) because this information is read by the kernel and has nothing to do with GRUB. The `initrd` is also located in the first logical partition of the first hard disk.
- The second entry is responsible for loading Windows. Windows is booted from the first partition of the first hard disk (`hd0,0`). The command `chainloader +1` causes GRUB to read and execute the first sector of the specified partition.
- The next entry enables booting from floppy disk without modifying the BIOS settings.
- The boot option `failsafe` starts Linux with a selection of kernel parameters that enables Linux to boot even on problematic systems.

The menu file can be changed whenever necessary. GRUB then uses the modified settings during the next boot. Edit the file permanently using YaST or an editor of your choice. Alternatively, make temporary changes interactively using the `edit` function of GRUB. See Section “Editing Menu Entries during the Boot Procedure” (page 245).

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.

18.1.2 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
```

18.1.3 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.

18.1.4 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 18.6, “Changing the Boot Loader Type” (page 254).

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 18.1 *Screen Resolution and Color Depth Reference*

	640x480	800x600	1024x768	1280x1024	1600x1200
8bit	0x301	0x303	0x305	0x307	0x31C

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.

18.1.5 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:

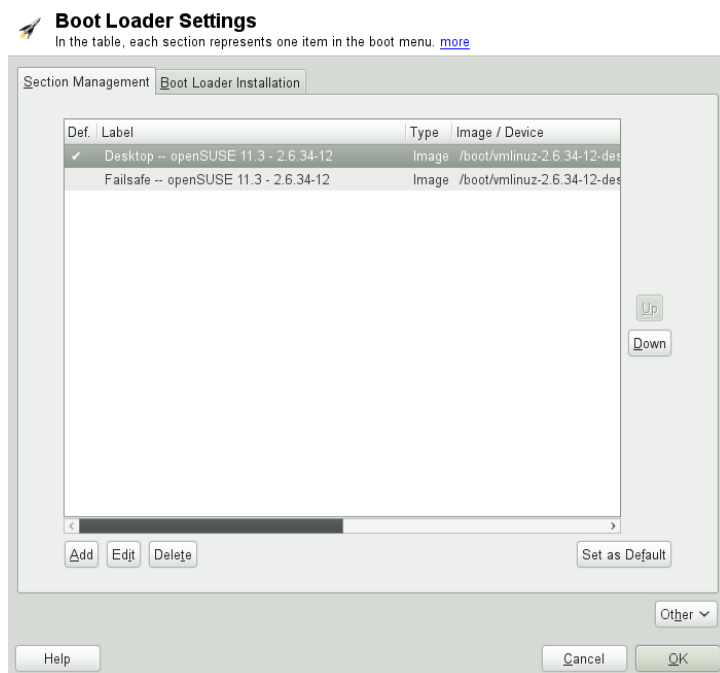
```
Error 32: Must be authenticated
```

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.

18.2 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 18.1, “Boot Loader Settings” (page 250), this shows the current boot loader configuration of your system and allows you to make changes.

Figure 18.1 *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 18.1, “Booting with GRUB” (page 240) 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 18.1, “Bootting with GRUB” (page 240). 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*.

18.2.1 Adjusting the Default Boot Entry

To change the system that is booted by default, proceed as follows:

Procedure 18.1 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.

18.2.2 Modifying the Boot Loader Location

To modify the location of the boot loader, follow these steps:

Procedure 18.2 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.

18.2.3 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 18.3 *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.

18.2.4 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 18.4 *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.

18.2.5 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 18.1.2, “The File device.map” (page 246)). To do so, proceed as follows:

Procedure 18.5 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.

18.2.6 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.

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>

18.2.7 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 18.6 *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.

18.3 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*.

18.4 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 18.7 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 \  
-boot-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.

18.5 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 18, *The Boot Loader GRUB* (page 239) 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.

18.6 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 cov-

ered 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`.

18.7 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.

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).

19.1 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.

19.1.1 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.

19.1.2 The cron Package

If you want to run commands regularly and automatically in the background at pre-defined 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 19.1, “Entry in `/etc/crontab`” (page 262), `root` is entered. Package-specific tables, located in `/etc/cron.d`, have the same format. See the cron man page (`man cron`).

Example 19.1 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 19.2, “`/etc/crontab`: Remove Time Stamp Files” (page 263), which removes the `hourly` one before every full hour, the `daily` one once a day at 2:14 a.m., etc.).

Example 19.2 */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`.

19.1.3 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, e.g. `apache2` (`/etc/logrotate.d/apache2`) and `syslogd` (`/etc/logrotate.d/syslog`).

Example 19.3 *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.

19.1.4 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.

19.1.5 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 19.1, “`ulimit`: Setting Resources for the User” (page 265).

Table 19.1 *ulimit: Setting Resources for the User*

<code>-m</code>	The maximum resident set size
-----------------	-------------------------------

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 19.4 *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.

19.1.6 The free Command

The `free` command is somewhat misleading if your goal is to find out how much RAM is currently being used. That information can be found in `/proc/meminfo`.

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

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

The kernel also contains other caches, such as the *slab cache*, where the caches used for network access are stored. This may explain the differences between the counters in `/proc/meminfo`. Most, but not all, of them can be accessed via `/proc/slabinfo`.

19.1.7 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.

19.1.8 Selecting Man Pages Using the man Command

With `man man_page` you normally display a man page for instant reading. Now, if a man page with the same name exists in different sections, `man` prompts the user, the page from which section shall be made visible. The user is then expected to type the section as the answer.

If you want to return to the previous behavior, set `MAN_POSIXLY_CORRECT=1` in a shell initialization file such as `~/ .bashrc`.

19.1.9 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.

19.2 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`.

19.3 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).

19.4 Language and Country-Specific Settings

The system is, to a very large extent, internationalized and can be flexibly modified for local needs. In other words, internationalization (*I18N*) allows specific localizations (*L10N*). The abbreviations *I18N* and *L10N* are derived from the first and last letters of the words and, in between, the number of letters omitted.

Settings are made with `LC_` variables defined in the file `/etc/sysconfig/language`. This refers not only to *native language support*, but also to the categories *Messages* (Language), *Character Set*, *Sort Order*, *Time and Date*, *Numbers* and *Money*. Each of these categories can be defined directly with its own variable or indirectly with a master variable in the file `language` (see the `locale` man page).

`RC_LC_MESSAGES`, `RC_LC_CTYPE`, `RC_LC_COLLATE`, `RC_LC_TIME`,
`RC_LC_NUMERIC`, `RC_LC_MONETARY`

These variables are passed to the shell without the `RC_` prefix and represent the listed categories. The shell profiles concerned are listed below. The current setting can be shown with the command `locale`.

`RC_LC_ALL`

This variable, if set, overwrites the values of the variables already mentioned.

`RC_LANG`

If none of the previous variables are set, this is the fallback. By default, only `RC_LANG` is set. This makes it easier for users to enter their own values.

`ROOT_USES_LANG`

A `yes` or `no` variable. If set to `no`, `root` always works in the POSIX environment.

The variables can be set with the YaST `sysconfig` editor (see Section 17.3.1, “Changing the System Configuration Using the YaST `sysconfig` Editor” (page 235)). 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>]
```

19.4.1 Some Examples

You should always set the language and country codes together. Language settings follow the standard ISO 639 available at <http://www.evertype.com/standards/iso639/iso639-en.html> and <http://www.loc.gov/standards/iso639-2/>. Country codes are listed in ISO 3166 available at http://www.din.de/gremien/nas/nabd/iso3166ma/codlstp1/en_listp1.html.

It only makes sense to set values for which usable description files can be found in `/usr/lib/locale`. Additional description files can be created from the files in `/usr/share/i18n` using the command `localedef`. The description files are part of the `glibc-i18ndata` package. A description file for `en_US.UTF-8` (for English and United States) can be created with:

```
localedef -i en_US -f UTF-8 en_US.UTF-8
```

```
LANG=en_US.UTF-8
```

This is the default setting if American English is selected during installation. If you selected another language, that language is enabled but still with UTF-8 as the character encoding.

```
LANG=en_US.ISO-8859-1
```

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

```
LANG=en_IE@euro
```

The above example explicitly includes the Euro sign in a language setting. 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.

SuSEconfig reads the variables in `/etc/sysconfig/language` and writes the necessary changes to `/etc/SuSEconfig/profile` and `/etc/SuSE-`


```
config/csh.cshrc  ./etc/SuSEconfig/profile  is read or sourced  
by /etc/profile  ./etc/SuSEconfig/csh.cshrc  is sourced by /  
etc/csh.cshrc    . This makes the settings available systemwide.
```

Users can override the system defaults by editing their `~/ .bashrc` accordingly. For instance, if you do not want to use the systemwide `en_US` for program messages, include `LC_MESSAGES=es_ES` so that messages are displayed in Spanish instead.

19.4.2 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
```

19.4.3 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.

19.4.4 For More Information

- *The GNU C Library Reference Manual*, Chapter “Locales and Internationalization”. It is included in `glibc-info`.
- Markus Kuhn, *UTF-8 and Unicode FAQ for Unix/Linux*, currently at <http://www.cl.cam.ac.uk/~mgk25/unicode.html>.
- *Unicode-Howto*, by Bruno Haible: `/usr/share/doc/howto/en/txt/Unicode-HOWTO.gz` (package `howto`).

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.

20.1 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`.

20.2 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 20.6, “Influencing Kernel Device Event Handling with `udev` Rules” (page 277).

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.

20.3 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.

ry 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`.

20.4 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.

20.5 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`.

20.6 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 20.1 *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/usb1p0`.

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.

20.6.1 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.

20.6.2 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.

20.6.3 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.

20.6.4 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.

20.7 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

```

20.8 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.

20.9 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.

Shell Basics

When working with Linux these days, you can communicate with the system almost without ever requiring a command line interpreter (the shell). After booting your Linux system, you are usually directed to a graphical user interface that guides you through the login process and the following interactions with the operating system. The graphical user interface in Linux (the X Window System or X11) is initially configured during installation. Both KDE and GNOME desktop (and other window managers you can install) use it for interaction with the user.

Nevertheless, it is useful to have some basic knowledge of working with a shell because you might encounter situations where the graphical user interface is not available. For example, if some problem with the X Window System occurs. If you are not familiar with a shell, you might feel a bit uncomfortable at first when entering commands, but the more you get used to it, the more you will realize that the command line is often the quickest and easiest way to perform some daily tasks.

For UNIX or Linux several shells are available which differ slightly in behavior and in the commands they accept. The default shell in openSUSE® is Bash (GNU Bourne-Again Shell).

The following sections will guide you through your first steps with the Bash shell and will show you how to complete some basic tasks via the command line. If you are interested in learning more or rather feel like a shell “power user” already, refer to Chapter 22, *Bash and Bash Scripts* (page 325).

21.1 Starting a Shell

Basically, there are two different ways to start a shell from the graphical user interface which usually shows after you have booted your computer:

- you can leave the graphical user interface or
- you can start a terminal window *within* the graphical user interface.

While the first option is always available, you can only make use of the second option when you are already logged in to a desktop such as KDE or GNOME.

Whichever way you choose, there is always a way back and you can switch back and forth between the shell and the graphical user interface.

If you want to give it a try, press `++F2` to leave the graphical user interface. The graphical user interface disappears and you are taken to a shell which prompts you to log in. Type your username and press `.` Then type your password and press `.` The prompt now changes and shows some useful information as in the following example:

```
❶ ❷ ❸  
tux@linux:~>
```

- ❶ Your login.
- ❷ The hostname of your computer.
- ❸ Path to the current directory. Directly after login, the current directory usually is your home directory, indicated by the `~` symbol (tilde) .

When you are logged in at a remote computer the information provided by the prompt always shows you which system you are currently working on.

When the cursor is located behind this prompt, you can pass commands directly to your computer system. For example, you can now enter `ls -l` to list the contents of the current directory in a detailed format. If this is enough for your first encounter with the shell and you want to go back to the graphical user interface, you should log out from your shell session first. To do so, type `exit` and press `.` Then press `++F7` to switch back to the graphical user interface. You will find your desktop and the applications running on it unchanged.

When you are already logged in to the GNOME or the KDE desktop and want to start a terminal window within the desktop, press `++F2` and enter `konsole` (for KDE) or `gnome-terminal` (for GNOME). This opens a terminal window on your desktop. As you are already logged in to your desktop, the prompt shows information about your system as described above. You can now enter commands and exe-

cute tasks just like in any shell which runs parallel to your desktop. To switch to another application on the desktop just click on the corresponding application window or select it from the taskbar of your panel. To close the terminal window press `+ F4`.

21.2 Entering Commands

As soon as the prompt appears on the shell it is ready to receive and execute commands. A command can consist of several elements. The first element is the actual command, followed by parameters or options. You can type a command and edit it by using the following keys: `, , ,` (Backspace), `,` and `.` You can correct typing errors or add options. The command is not executed until you press `.`

IMPORTANT: No News Is Good News

The shell is not verbose: in contrast to some graphical user interfaces, it usually does not provide confirmation messages when commands have been executed. Messages only appear in case of problems or errors—or if you explicitly ask for them by executing a command with a certain option.

Also keep this in mind for commands to delete objects. Before entering a command like `rm` (without any option) for removing a file, you should know if you really want to get rid of the object: it will be deleted irretrievably, without confirmation.

21.2.1 Using Commands without Options

In Section 21.5.1, “Permissions for User, Group and Others” (page 298) you already got to know one of the most basic commands: `ls`, which used to list the contents of a directory. This command can be used with or without options. Entering the plain `ls` command shows the contents of the current directory:

```
tux@knox:~> ls
bin Desktop Documents public_html tux.txt
tux@knox:~>
```

Files in Linux may have a file extension or a suffix, such as `.txt`, but do not need to have one. This makes it difficult to differentiate between files and folders in this output of the `ls`. By default, the colors in the Bash shell give you a hint: directories are usually shown in blue, files in black.

21.2.2 Using Commands with Options

A better way to get more details about the contents of a directory is using the `ls` command with a string of options. Options modify the way a command works so that you can get it to carry out specific tasks. Options are separated from the command with a blank and are usually prefixed with a hyphen. The `ls -l` command shows the contents of the same directory in full detail (long listing format):

```
tux@knox:~> ls -l
drwxr-xr-x 1 tux users      48 2006-06-23 16:08 bin
drwx---r-- 1 tux users  53279 2006-06-21 13:16 Desktop
drwx----- 1 tux users    280 2006-06-23 16:08 Documents
drwxr-xr-x 1 tux users  70733 2006-06-21 09:35 public_html
-rw-r--r-- 1 tux users  47896 2006-06-21 09:46 tux.txt
tux@knox:~>
```

This output shows the following information about each object:

```
drwxr-xr-x❶ 1❷ tux❸ users❹ 48❺ 2006-06-23 16:08❻ bin❼
```

- ❶ Type of object and access permissions. For further information, refer to Section 21.5.1, “Permissions for User, Group and Others” (page 298).
- ❷ Number of hard links to this file.
- ❸ Owner of the file or directory. For further information, refer to Section 21.5.1, “Permissions for User, Group and Others” (page 298).
- ❹ Group assigned to the file or directory. For further information, refer to Section 21.5.1, “Permissions for User, Group and Others” (page 298).
- ❺ File size in bytes.
- ❻ Date and time of the last change.
- ❼ Name of the object.

Usually, you can combine several options by prefixing only the first option with a hyphen and then write the others consecutively without a blank. For example, if you want to see all files in a directory in long listing format, you can combine the two options `-l` and `-a` (show all files) for the `ls` command. Executing `ls -la` shows also hidden files in the directory, indicated by a dot in front (for example, `.hidden-file`).

The list of contents you get with `ls` is sorted alphabetically by filenames. But like in a graphical file manager, you can also sort the output of `ls -l` according to various criteria such as date, file extension or file size:

- For date and time, use `ls -lt` (displays newest first).

- For extensions, use `ls -lx` (displays files with no extension first).
- For file size, use `ls -lS` (displays largest first).

To revert the order of sorting, add `-r` as an option to your `ls` command. For example, `ls -lr` gives you the contents list sorted in reverse alphabetical order, `ls -ltr` shows the oldest files first. There are lots of other useful options for `ls`. In the following section you will learn how to investigate them.

21.2.3 Getting Help

Nobody is expected to know all options of all commands by heart. If you remember the command name but are not sure about the options or the syntax of the command, choose one of the following possibilities:

`--help` option

If you only want to look up the options of a certain command, try entering the command followed by a space and `--help`. This `--help` option exists for many commands. For example, `ls --help` displays all the options for the `ls` command.

Manual Pages

To learn more about the various commands, you can also use the manual pages. Manual pages also give a short description of what the command does. They can be accessed with `man` followed by the name of the command, for example, `man ls`.

The man pages are displayed directly in the shell. To navigate them, move up and down with `↑` and `↓`. Move between the beginning and the end of a document with `⏪` and `⏩`. End this viewing mode by pressing `Q`. Learn more about the `man` command itself with `man man`.

Info Pages

Info pages usually provide even more information about commands. To view the info page for a certain command, enter `info` followed by the name of the command (for example, `info ls`). You can browse an info page with a viewer directly in the shell and display the different sections, called “nodes.” Use `⏪` to move forward and `⏩` to move backwards. Within a node, you can also browse with `⏪` and `⏩` but only `⏪` and `⏩` will take you also to the previous or subsequent node. Like for the man pages, press `Q` to end the viewing mode.

Note that man pages and info pages do not exist for all commands. Sometimes both are available (usually for key commands), sometimes only a man page or an info page exists, and sometimes neither of them are available.

21.2.4 Bash Shortcut Keys

After having entered several commands, your shell will begin to fill up with all sorts of commands and the corresponding outputs. In the following table, find some useful shortcut keys for navigating and editing in the shell.

Shortcut Key	Function
+ L	Clears the screen and moves the current line to the top of the page.

21.3 Working with Files and Directories

To address a certain file or directory, you must specify the path leading to that directory or file. As you may know from MS DOS or Mac OS already, there are two ways to specify a path:

Absolute Path

Enter the entire path from the root directory to the relevant file or directory.

Relative Path

Enter a path to the relevant file or directory by using the current directory as a starting point. This implies to give the levels you have to move up or down in the file system tree to reach the target directory of file, starting from the current directory.

Paths contain filenames, directories or both, separated by slashes. Absolute paths always start with a slash. Relative paths do not have a slash at the beginning, but can have one or two dots.

When entering commands, you can choose either way to specify a path, depending on your preferences or the amount of typing, both will lead to the same result. To change directories, use the `cd` command and specify the path to the directory.

NOTE: Handling Blanks in Filenames or Directory Names

If a filename or the name of a directory contains a space, either escape the space using a back slash (\) in front of the blank or enclose the filename in single quotes. Otherwise Bash interprets a filename like `My Documents` as the names of two files or directories, `My` and `Documents` in this case.

When specifying paths, the following “shortcuts” can save you a lot of typing:

- The tilde symbol (~) is a shortcut for home directories. For example, to list the contents of your home directory, use `ls ~`. To list the contents of another user's home directory, enter `ls ~username` (or course, this will only work if you have permission to view the contents, see Section 21.5, “File Access Permissions” (page 297)). For example, entering `ls ~tux` would list the contents of the home directory of a user named `tux`. You can use the tilde symbol as shortcut for home directories also if you are working in a network environment where your home directory may not be called `/home` but can be mapped to any directory in the file system.

From anywhere in the file system, you can reach your home directory by entering `cd ~` or by simply entering `cd` without any options.

- When using relative paths, refer to the current directory with a dot (`.`). This is mainly useful for commands such as `cp` or `mv` by which you can copy or move files and directories.
- The next higher level in the tree is represented by two dots (`..`). In order to switch to the parent directory of your current directory, enter `cd ..`, to go up two levels from the current directory enter `cd ../..` etc.

To apply your knowledge, find some examples below. They address basic tasks you may want to execute with files or folders using Bash.

21.3.1 Examples for Working with Files and Directories

Suppose you want to copy a file located somewhere in your home directory to a sub-directory of `/tmp` that you need to create first.

Procedure 21.1 Creating and Changing Directories

From your home directory create a subdirectory in `/tmp`:

1 Enter

```
mkdir /tmp/test
```

`mkdir` stands for “make directory”. This command creates a new directory named `test` in the `/tmp` directory. In this case, you are using an absolute path to create the `test` directory.

2 To check what happened, now enter

```
ls -l /tmp
```

The new directory `test` should appear in the list of contents of the `/tmp` directory.

3 Switch to the newly created directory with

```
cd /tmp/test
```

Procedure 21.2 Creating and Copying Files

Now create a new file in a subdirectory of your home directory and copy it to `/tmp/test`. Use a relative path for this task.

IMPORTANT: Overwriting of Existing Files

Before copying, moving or renaming a file, check if your target directory already contains a file with the same name. If yes, consider changing one of the filenames or use `cp` or `mv` with options like `-i`, which will prompt before overwriting an existing file. Otherwise Bash will overwrite the existing file without confirmation.

1 To list the contents of your home directory, enter

```
ls -l ~
```

It should contain a subdirectory called `Documents` by default. If not, create this subdirectory with the `mkdir` command you already know:

```
mkdir ~/Documents
```

2 To create a new, empty file named `myfile.txt` in the `Documents` directory, enter


```
touch ~/Documents/myfile.txt
```

Usually, the `touch` command updates the modification and access date for an existing file. If you use `touch` with a filename which does not exist in your target directory, it creates a new file.

3 Enter

```
ls -l ~/Documents
```

The new file should appear in the list of contents.

4 To copy the newly created file, enter

```
cp ~/Documents/myfile.txt .
```

Do not forget the dot at the end.

This command tells Bash to go to your home directory and to copy `myfile.txt` from the `Documents` subdirectory to the current directory, `/tmp/test`, without changing the name of the file.

5 Check the result by entering

```
ls -l
```

The file `myfile.txt` should appear in the list of contents for `/tmp/test`.

Procedure 21.3 Renaming and Removing Files or Directories

Now suppose you want to rename `myfile.txt` into `tuxfile.txt`. Finally you decide to remove the renamed file and the `test` subdirectory.

1 To rename the file, enter

```
mv myfile.txt tuxfile.txt
```

2 To check what happened, enter

```
ls -l
```

Instead of `myfile.txt`, `tuxfile.txt` should appear in the list of contents.

`mv` stands for `move` and is used with two options: the first option specifies the source, the second option specifies the target of the operation. You can use `mv` either

- to rename a file or a directory,
 - to move a file or directory to a new location or
 - to do both in one step.
- 3** Coming to the conclusion that you do not need the file any longer, you can delete it by entering
- ```
rm tuxfile.txt
```
- Bash deletes the file without any confirmation.
- 4** Move up one level with `cd ..` and check with
- ```
ls -l test
```
- if the `test` directory is empty now.
- 5** If yes, you can remove the `test` directory by entering
- ```
rmdir test
```

## 21.4 Becoming Root

`root`, also called the superuser, has privileges which authorize him to access all parts of the system and to execute administrative tasks. He or she has the unrestricted capacity to make changes to the system and has unlimited access to all files. Therefore performing some administrative tasks or running certain programs such as YaST requires `root` permissions.

### 21.4.1 Using su

In order to temporarily become `root` in a shell, proceed as follows:

- 1** Enter `su`. You are prompted for the `root` password.
- 2** Enter the password. If you mistyped the `root` password, the shell displays a message. In this case, you have to re-enter `su` before retyping the password. If your password is correct, a hash symbol `#` appears at the end of the prompt, signaling that you are acting as `root` now.

- 3 Execute your task. For example, transfer ownership of a file to a new user which only `root` is allowed to do:

```
chown wilber kde_quick.xml
```

- 4 After having completed your tasks as `root`, switch back to your normal user account. To do so, enter

```
exit
```

The hash symbol disappears and you are acting as “normal” user again.

## 21.4.2 Using `sudo`

Alternatively, you can also use `sudo` (superuser “do”) to execute some tasks which normally are for `roots` only. With `sudo`, administrators can grant certain users `root` privileges for some commands. Depending on the system configuration, users can then run `root` commands by entering their normal password only. Due to a timestamp function, users are only granted a “ticket” for a restricted period of time after having entered their password. The ticket usually expires after a few minutes. In openSUSE, `sudo` requires the `root` password by default (if not configured otherwise by your system administrator).

For users, `sudo` is convenient as it prevents you from switching accounts twice (to `root` and back again). To change the ownership of a file using `sudo`, only one command is necessary instead of three:

```
sudo chown wilber kde_quick.xml
```

After you have entered the password which you are prompted for, the command is executed. If you enter a second `root` command shortly after that, you are not prompted for the password again, because your ticket is still valid. After a certain amount of time, the ticket automatically expires and the password is required again. This also prevents unauthorized persons from gaining `root` privileges in case a user forgets to switch back to his normal user account again and leaves a `root` shell open.

## 21.5 File Access Permissions

In Linux, objects such as files or folders or processes generally belong to the user who created or initiated them. There are some exceptions to this rule. For more infor-

mation about the exceptions, refer to Chapter 10, *Access Control Lists in Linux* (↑Security Guide). The group which is associated with a file or a folder depends on the primary group the user belongs to when creating the object.

When you create a new file or directory, initial access permissions for this object are set according to a predefined scheme. As an owner of a file or directory, you can change the access permissions for this object. For example, you can protect files holding sensitive data against read access by other users and you can authorize the members of your group or other users to write, read, or execute several of your files where appropriate. As `root`, you can also change the ownership of files or folders.

## 21.5.1 Permissions for User, Group and Others

Three permission sets are defined for each file object on a Linux system. These sets include the read, write, and execute permissions for each of three types of users—the owner, the group, and other users.

The following example shows the output of an `ls -l` command in a shell. This command lists the contents of a directory and shows the details for each file and folder in that directory.

**Example 21.1** *Access Permissions For Files and Folders*

```
-rw-r----- 1 tux users 0 2006-06-23 16:08 checklist.txt
-rw-r--r-- 1 tux users 53279 2006-06-21 13:16 gnome_quick.xml
-rw-rw---- 1 tux users 0 2006-06-23 16:08 index.htm
-rw-r--r-- 1 tux users 70733 2006-06-21 09:35 kde-start.xml
-rw-r--r-- 1 tux users 47896 2006-06-21 09:46 kde_quick.xml
drwxr-xr-x 2 tux users 48 2006-06-23 16:09 local
-rwxr--r-- 1 tux users 624398 2006-06-23 15:43 tux.sh
```

As shown in the third column, all objects belong to user `tux`. They are assigned to the group `users` which is the primary group the user `tux` belongs to. To retrieve the access permissions the first column of the list must be examined more closely. Let's have a look at the file `kde-start.xml` :

| Type | User Permissions | Group Permissions | Permissions for Others |
|------|------------------|-------------------|------------------------|
|------|------------------|-------------------|------------------------|

The first column of the list consists of one leading character followed by nine characters grouped in three blocks. The leading character indicates the file type of the object: in this case, the hyphen (-) shows that `kde-start.xml` is a file. If you find the character `d` instead, this shows that the object is a directory, like `local` in Example 21.1, “Access Permissions For Files and Folders” (page 298).

The next three blocks show the access permissions for the owner, the group and other users (from left to right). Each block follows the same pattern: the first position shows read permissions (`r`), the next position shows write permissions (`w`), the last one shows execute permission (`x`). A lack of either permission is indicated by `-`. In our example, the owner of `kde-start.xml` has read and write access to the file but cannot execute it. The `users` group can read the file but cannot write or execute it. The same holds true for the other users as shown in the third block of characters.

## 21.5.2 Files and Folders

Access permissions have a slightly different impact depending on the type of object they apply to: file or directory. The following table shows the details:

**Table 21.1** *Access Permissions For Files And Directories*

| Access Permission | File                              | Folder                                                                                                                                                                                                                                                                                                                        |
|-------------------|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Read (r)          | Users can open and read the file. | Users can view the contents of the directory. Without this permission, users cannot list the contents of this directory with <code>ls -l</code> , for example. However, if they only have execute permission for the directory, they can nevertheless access certain files in this directory if they know of their existence. |

Note that access to a certain file is always dependent on the correct combination of access permissions for the file itself *and* the directory it is located in.

## 21.5.3 Modifying File Permissions

In Linux, objects such as files or folder or processes generally belong to the user who created or initiated them. The group which is associated with a file or a folder depends on the primary group the user belongs to when creating the object. When you create a new file or directory, initial access permissions for this object are set according to a predefined scheme. For further details refer to Section 21.5, “File Access Permissions” (page 297).

As the owner of a file or directory (and, of course, as `root`), you can change the access permissions to this object.

To change object attributes like access permissions of a file or folder, use the `chmod` command followed by the following parameters:

- the users for which to change the permissions,
- the type of access permission you want to remove, set or add and
- the files or folders for which you want to change permissions separated by spaces.

The users for which you can change file access permissions fall into the following categories: the owner of the file (user, `u`), the group that own the file (group, `g`) and the other users (others, `o`). You can add, remove or set one or more of the following permissions: read, write or execute.

As `root`, you can also change the ownership of a file: with the command `chown` (change owner) you can transfer ownership to a new user.

## Examples for Changing Access Permissions and Ownership

The following example shows the output of an `ls -l` command in a shell.

### ***Example 21.2 Access Permissions For Files and Folders***

```
-rw-r----- 1 tux users 0 2006-06-23 16:08 checklist.txt
-rw-r--r-- 1 tux users 53279 2006-06-21 13:16 gnome_quick.xml
-rw-rw---- 1 tux users 0 2006-06-23 16:08 index.htm
-rw-r--r-- 1 tux users 70733 2006-06-21 09:35 kde-start.xml
-rw-r--r-- 1 tux users 47896 2006-06-21 09:46 kde_quick.xml
drwxr-xr-x 2 tux users 48 2006-06-23 16:09 local
-r-xr-xr-x 1 tux users 624398 2006-06-23 15:43 tux.jpg
```

In the example above, user `tux` owns the file `kde-start.xml` and has read and write access to the file but cannot execute it. The `users` group can read the file but cannot write or execute it. The same holds true for the other users as shown by the third block of characters.

### ***Procedure 21.4 Changing Access Permissions***

Suppose you are `tux` and want to modify the access permissions to your files:

- 1 If you want to grant the `users` group also write access to `kde-start.xml`, enter

```
chmod g+w kde-start.xml
```

- 2 To grant the `users` group and other users write access to `kde-start.xml`, enter

```
chmod go+w kde-start.xml
```

- 3 To remove write access for all users, enter

```
chmod -w kde-start.xml
```

If you do not specify any kind of users, the changes apply to all users— the owner of the file, the owning group and the others. Now even the owner `tux` does not have write access to the file without first reestablishing write permissions.

- 4 To prohibit the `users` group and others to change into the directory `local`, enter

```
chmod go-x local
```

- 5 To grant others write permissions for two files, for `kde_quick.xml` and `gnome_quick.xml`, enter

```
chmod o+w kde_quick.xml gnome_quick.xml
```

### ***Procedure 21.5 Changing Ownership***

Suppose you are `tux` and want to transfer the ownership of the file `kde_quick.xml` to an other user, named `wilber`. In this case, proceed as follows:

- 1 Enter the username and password for `root`.

- 2 Enter

```
chown wilber kde_quick.xml
```

### 3 Check what happened with

```
ls -l kde_quick.xml
```

You should get the following output:

```
-rw-r--r-- 1 wilber users 47896 2006-06-21 09:46 kde_quick.xml
```

### 4 If the ownership is set according to your wishes, switch back to your normal user account.

## 21.6 Useful Features of the Shell

As you probably noticed in the examples above, entering commands in Bash can include a lot of typing. In the following, get to know some features of the Bash that can make your work a lot easier and save a lot of typing.

### History

By default, Bash “remembers” commands you have entered. This feature is called *history*. You can browse through commands that have been entered before, select one you want to repeat and then execute it again. To do so, press repeatedly until the desired command appears at the prompt. To move forward through the list of previously entered commands, press `.` For easier repetition of a certain command from Bash history, just type the first letter of the command you want to repeat and press `.`

You can now edit the selected command (for example, change the name of a file or a path), before you execute the command by pressing `.` To edit the command line, just move the cursor to the desired position using the arrow keys and start typing.

You can also search for a certain command in the history. Press `Ctrl + R` to start an incremental search function. showing the following prompt:

```
(reverse-i-search)`':
```

Just type one or several letters from the command you are searching for. Each character you enter narrows down the search. The corresponding search result is shown on the right side of the colon whereas your input appears on the left of the colon. To accept a search result, press `.` The prompt now changes to its normal



appearance and shows the command you chose. You can now edit the command or directly execute it by pressing .

### Completion

Completing a filename or directory name to its full length after typing its first letters is another helpful feature of Bash. To do so, type the first letters then press (Tabulator). If the filename or path can be uniquely identified, it is completed at once and the cursor moves to the end of the filename. You can then enter the next option of the command, if necessary. If the filename or path cannot be uniquely identified (because there are several filenames starting with the same letters), the filename or path is only completed up to the point where it becomes ambiguous again. You can then obtain a list of them by pressing a second time. After this, you can enter the next letters of the file or path then try completion again by pressing . When completing filenames and paths with the help of , you can simultaneously check whether the file or path you want to enter really exists (and you can be sure of getting the spelling right).

### Wild Cards

You can replace one or more characters in a filename with a wild card for path-name expansion. Wild cards are characters that can stand for other characters. There are three different types of these in Bash:

---

| Wild Card | Function |
|-----------|----------|
|-----------|----------|

---

## 21.6.1 Examples For Using History, Completion and Wildcards

The following examples illustrate how to make use of these convenient features of Bash.

### ***Procedure 21.6*** *Using History and Completion*

If you already did the example Section 21.3.1, “Examples for Working with Files and Directories” (page 293) your shell buffer should be filled with commands which you can retrieve using the history function.

- 1 Press repeatedly until `cd ~` appears.
- 2 Press to execute the command and to switch to your home directory.

By default, your home directory contains two subdirectories starting with the same letter, `Documents` and `Desktop`.

- 3** Enter `cd D` and press `.`

Nothing happens since Bash cannot identify to which one of the subdirectories you want to change.

- 4** Press `.` again to see the list of possible choices:

```
tux@knox:~> cd D Desktop/ Documents/ tux@knox:~> cd D
```

- 5** The prompt still shows your initial input. Type the next character of the subdirectory you want to go to and press `.` again.

Bash now completes the path.

- 6** You can now execute the command with `.`

### ***Procedure 21.7 Using Wildcards***

Now suppose that your home directory contains a number of files with various file extensions. It also holds several versions of one file which you saved under different filenames `myfile1.txt`, `myfile2.txt` etc. You want to search for certain files according to their properties.

- 1** First, create some test files in your home directory:

- 1a** Use the `touch` command to create several (empty) files with different file extensions, for example `.pdf`, `.xml` and `.jpg`.

You can do this consecutively (do not forget to use the Bash history function) or with only one `touch` command: simply add several filenames separated by a space.

- 1b** Create at least two files that have the same file extension, for example `.html`.

- 1c** To create several “versions” of one file, enter

```
touch myfile{1..5}.txt
```

This command creates five consecutively numbered files:

```
myfile1.txt,...,myfile5.txt
```

**1d** List the contents of your home directory. It should look similar to this:

```
-rw-r--r-- 1 tux users 0 2006-07-14 13:34 foo.xml
-rw-r--r-- 1 tux users 0 2006-07-14 13:47 home.html
-rw-r--r-- 1 tux users 0 2006-07-14 13:47 index.html
-rw-r--r-- 1 tux users 0 2006-07-14 13:47 toc.html
-rw-r--r-- 1 tux users 0 2006-07-14 13:34 manual.pdf
-rw-r--r-- 1 tux users 0 2006-07-14 13:49 myfile1.txt
-rw-r--r-- 1 tux users 0 2006-07-14 13:49 myfile2.txt
-rw-r--r-- 1 tux users 0 2006-07-14 13:49 myfile3.txt
-rw-r--r-- 1 tux users 0 2006-07-14 13:49 myfile4.txt
-rw-r--r-- 1 tux users 0 2006-07-14 13:49 myfile5.txt
-rw-r--r-- 1 tux users 0 2006-07-14 13:32 tux.png
```

**2** With the help of wild cards, select certain subsets of the files according to various criteria:

**2a** To list all files with the `.html` extension, enter

```
ls -l *.html
```

**2b** To list all “versions” of `myfile.txt` , enter

```
ls -l myfile?.txt
```

Note that you can only use the `?` wild card here because the numbering of the files is single-digit. As soon as you have a file named `myfile10.txt` you must to use the `*` wild card to view all versions of `myfile.txt` (or add another question mark, so your string looks like `myfile???.txt`).

**2c** To remove, for example, version 1-3 and version 5 of `myfile.txt` , enter

```
rm myfile[1-3,5].txt
```

**2d** Check the result with

```
ls -l
```

Of all `myfile.txt` versions only `myfile4.txt` should be left.

You can also combine several wild cards in one command. In the example above, `rm myfile[1-3,5].*` would lead to the same result as `rm myfile[1-3,5].txt` because there are only files with the extension `.txt` available.

---

**NOTE: Using Wildcards in rm Commands**

Wildcards in a `rm` command can be very useful but also dangerous: you might delete more files from your directory than intended. To see which files would be affected by the `rm`, run your wildcard string with `ls` instead of `rm` first.

---

## 21.7 Editing Texts

In order to edit files from the command line, you will need to know the `vi` editor. `vi` is a default editor which can be found on nearly every UNIX/Linux system. It can run several operating modes in which the keys you press have different functions. This does not make it very easy for beginners, but you should know at least the most basic operations with `vi`. There may be situations where no other editor than `vi` is available.

Basically, `vi` makes use of three operating modes:

*command* mode

In this mode, `vi` accepts certain key combinations as commands. Simple tasks such as searching words or deleting a line can be executed.

*insert* mode

In this mode, you can write normal text.

*extended* mode

In this mode, also known as colon mode (as you have to enter a colon to switch to this mode), `vi` can execute also more complex tasks such as searching and replacing text.

In the following (very simple) example, you will learn how to open and edit a file with `vi`, how to save your changes and quit `vi`.

### 21.7.1 Example: Editing with vi

---

**NOTE: Display of Keys**

In the following, find several commands that you can enter in `vi` by just pressing keys. These appear in uppercase as on a keyboard. If you need

to enter a key in uppercase, this is stated explicitly by showing a key combination including the `key`.

---

- 1 To create and open a new file with `vi`, enter

```
vi textfile.txt
```

By default, `vi` opens in *command* mode in which you cannot enter text.

- 2 Press `I` to switch to insert mode. The bottom line changes and indicates that you now can insert text.
- 3 Write some sentences. If you want to insert a new line, first press `Esc` to switch back to command mode. Press `O` to insert a new line and to switch to insert mode again.
- 4 In the insert mode, you can edit the text with the arrow keys and with `.`
- 5 To leave `vi`, press `Esc` to switch to command mode again. Then press `:` which takes you to the extended mode. The bottom line now shows a colon.
- 6 To leave `vi` and save your changes, type `wq` (`w` for write; `q` for quit) and press `.`. If you want to save the file under a different name, type `w filename` and press `.`.

To leave `vi` without saving, type `q!` instead and press `.`

## 21.8 Searching for Files or Contents

Bash offers you several commands to search for files and to search for the contents of files:

`locate`

This utility is only available if you have installed the `findutils-locate` package. With this command you can find out in which directory a specified file is located. If desired, use wild cards to specify filenames. The program is very quick, because it uses a database specifically created for the purpose (rather than searching through the entire file system). This very fact, however, also results in a major drawback: `locate` is unable to find any files created after the latest update of its database. The database can be generated by `root` running `updatedb`.

`find`

With `find`, search for a file in a given directory. The first argument specifies the directory in which to start the search. The option `-name` must be followed by a search string, which may also include wild cards. Unlike `locate`, which uses a database, `find` scans the actual directory.

`grep`

The `grep` command finds a specific search string in the specified text files. If the search string is found, the command displays the line in which `search-string` was found, along with the filename. If desired, use wild cards to specify filenames.

## 21.8.1 Examples for Searching

The KDE and GNOME desktops store user-specific application data in hidden directories, for example `.kde` and `.gnome`.

- 1 To locate these directories on your computer, enter

```
locate .kde
```

if you have installed KDE desktop or

```
locate .gnome
```

if you have installed GNOME desktop.

You will see that `locate` displays all file names in the database that contain the string `.kde` or `.gnome` anywhere. To learn how to modify this behavior refer to the man page of `locate`.

- 2 To search your home directory for all occurrences of filenames that contain the file extension `.txt`, use

```
find ~ -name '*.txt' -print
```

- 3 To search a directory (in this case, your home directory) for all occurrences of files which contain, for example, the word `music`, enter

```
grep music ~/*
```

Note that `grep` is case-sensitive— unless you use it with the `-i` option. With the command above you will not find any files containing `Music`.

If you want to use a search string which consists of more than one word, enclose the string in double quotation marks, for example:

```
grep "music is great" ~/*
```

## 21.9 Viewing Text Files

When searching for the contents of a file with `grep`, the output gives you the line in which the `searchstring` was found along with the filename. Often this contextual information is still not enough information to decide whether you want to open and edit this file. Bash offers you several commands to have a quick look at the contents of a text file directly in the shell, without opening an editor.

`head`

With `head` you can view the first lines of a text file. If you do not specify the command any further, `head` shows the first 10 lines of a text file.

`tail`

The `tail` command is the counterpart of `head`. If you use `tail` without any further options it displays the last 10 lines of a text file. This can be very useful to view log files of your system, where the most recent messages or log entries are usually found at the end of the file.

`less`

With `less`, display the whole contents of a text file. To move up and down half a page use `and` `.` Use `to` scroll down one page. `to` takes you to the beginning, and `to` to the end of the document. To end the viewing mode, press `Q`.

`more`

Instead of `less`, you can also use the older program `more`. It has basically the same function—however, it is less convenient because it does not allow you to scroll backwards. Use `to` to move forward. When you reach the end of the document, the viewer closes automatically.

`cat`

The `cat` command displays the contents of a file, printing the entire contents to the screen without interruption. As `cat` does not allow you to scroll it is not very useful as viewer but it is rather often used in combination with other commands.

# 21.10 Redirection and Pipes

Sometimes it would be useful if you could write the output of a command to a file for further editing or if you could combine several commands, using the output of one command as the input for the next one. The shell offers this function by means of redirection or pipes.

Normally, the standard output in the shell is your screen (or an open shell window) and the standard input is the keyboard. With the help of certain symbols you can redirect the input or the output to another object, such as a file or another command.

## Redirection

With `>` you can forward the output of a command to a file (output redirection), with `<` you can use a file as input for a command (input redirection).

## Pipe

By means of a pipe symbol `|` you can also redirect the output: with a pipe, you can combine several commands, using the output of one command as input for the next command. In contrast to the other redirection symbols `>` and `<`, the use of the pipe is not constrained to files.

## 21.10.1 Examples for Redirection and Pipe

- 1 To write the output of a command like `ls` to a file, enter

```
ls -l > filelist.txt
```

This creates a file named `filelist.txt` that contains the list of contents of your current directory as generated by the `ls` command.

However, if a file named `filelist.txt` already exists, this command overwrites the existing file. To prevent this, use `>>` instead of `>`. Entering

```
ls -l >> filelist.txt
```

simply appends the output of the `ls` command to an already existing file named `filelist.txt`. If the file does not exist, it is created.

- 2 Redirections also works the other way round. Instead of using the standard input from the keyboard for a command, you can use a file as input:



```
sort < filelist.txt
```

This will force the `sort` command to get its input from the contents of `filelist.txt`. The result is shown on the screen. Of course, you can also write the result into another file, using a combination of redirections:

```
sort < filelist.txt > sorted_filelist.txt
```

- 3** If a command generates a lengthy output, like `ls -l` may do, it may be useful to pipe the output to a viewer like `less` to be able to scroll through the pages. To do so, enter

```
ls -l | less
```

The list of contents of the current directory is shown in `less`.

The pipe is also often used in combination with the `grep` command in order to search for a certain string in the output of another command. For example, if you want to view a list of files in a directory which are owned by the user `tux`, enter

```
ls -l | grep tux
```

## 21.11 Starting Programs and Handling Processes

As you have seen in Section 21.7, “Editing Texts” (page 306), programs can be started from the shell. Applications with a graphical user interface need the X Window System and can only be started from a terminal window within a graphical user interface. For example, if you want to open a file named `vacation.pdf` in your home directory from a terminal window in KDE or GNOME, simply run `okular ~/vacation.pdf` (or `evince ~/vacation.pdf`) to start a PDF viewer displaying your file.

When looking at the terminal window again you will realize that the command line is blocked as long as the PDF viewer is open, meaning that your prompt is not available. To change this, press `+Z` to suspend the process and enter `bg` to send the process to the background. Now you can still have a look at `vacation.pdf` while your prompt is available for further commands. An easier way to achieve this is by sending a process to the background directly when starting it. To do so, add an ampersand at the end of the command:

```
okular ~/vacation.pdf &
```

If you have started several background processes (also named jobs) from the same shell, the `jobs` command gives you an overview of the jobs. It also shows the job number in brackets and their status:

```
tux@linux:~> jobs
[1] Running okular book.opensuse.startup-xep.pdf &
[2]- Running okular book.opensuse.reference-xep.pdf &
[3]+ Stopped man jobs
```

To bring a job to the foreground again, enter `fg job_number`.

Whereas `jobs` only shows the background processes started from a specific shell, the `ps` command (run without options) shows a list of all your processes—those you started. Find an example output below:

```
tux@linux:~> ps
PID TTY TIME CMD
15500 pts/1 00:00:00 bash
28214 pts/1 00:00:00 okular
30187 pts/1 00:00:00 kwrite
30280 pts/1 00:00:00 ps
```

In case a program cannot be terminated in the normal way, use the `kill` command to stop the process (or processes) belonging to that program. To do so, specify the process ID (PID) shown by the output of `ps`. For example, to shut down the KWrite editor in the example above, enter

```
kill 30187
```

This sends a *TERM* signal that instructs the program to shut itself down.

Alternatively, if the program or process you want to terminate is a background job and is shown by the `jobs` command, you can also use the `kill` command in combination with the job number to terminate this process. When identifying the job with the job number, you must prefix the number with a percent character (%):

```
kill %job_number
```

If `kill` does not help—as is sometimes the case for “runaway” programs—try

```
kill -9 PID
```

This sends a *KILL* signal instead of a *TERM* signal, bringing the specified process to an end in most cases.

This section is intended to introduce the most basic set of commands for handling jobs and processes. Find an overview for system administrators in Section “Processes” (Chapter 2, *System Monitoring Utilities*, ↑System Analysis and Tuning Guide).

# 21.12 Important Linux Commands

This section gives insight into the most important commands. There are many more commands than listed in this chapter. Along with the individual commands, parameters are listed and, where appropriate, a typical sample application is introduced. To learn more about the various commands, use the manual pages, accessed with `man` followed by the name of the command, for example, `man ls`.

Man pages are displayed directly in the shell. To navigate them, move up and down with `↑` and `↓`. Move between the beginning and the end of a document with `⏪` and `⏩`. End this viewing mode by pressing `Q`. Learn more about the `man` command itself with `man man`.

In the following overview, the individual command elements are written in different typefaces. The actual command and its mandatory options are always printed as `command option`. Specifications or parameters that are not required are placed in `[square brackets]`.

Adjust the settings to your needs. It makes no sense to write `ls file` if no file named `file` actually exists. You can usually combine several parameters, for example, by writing `ls -la` instead of `ls -l -a`.

## 21.12.1 File Commands

The following section lists the most important commands for file management. It covers everything from general file administration to the manipulation of file system ACLs.

### File Administration

`ls [options] [files]`

If you run `ls` without any additional parameters, the program lists the contents of the current directory in short form.

`-l`

Detailed list

`-a`

Displays hidden files

`cp [options] source target`

**Copies** source to target.

**-i**

Waits for confirmation, if necessary, before an existing target is overwritten

**-r**

Copies recursively (includes subdirectories)

`mv [options] source target`

**Moves** source to target then deletes the original source.

**-b**

Creates a backup copy of the source before moving

**-i**

Waits for confirmation, if necessary, before an existing target file is overwritten

`rm [options] files`

**Removes** the specified files from the file system. Directories are not removed by `rm` unless the option `-r` is used.

**-r**

Deletes any existing subdirectories

**-i**

Waits for confirmation before deleting each file

`ln [options] source target`

**Creates** an internal link from source to target. Normally, such a link points directly to source on the same file system. However, if `ln` is executed with the `-s` option, it creates a symbolic link that only points to the directory in which source is located, enabling linking across file systems.

**-s**

Creates a symbolic link

`cd [options] [directory]`

**Changes** the current directory. `cd` without any parameters changes to the user's home directory.

`mkdir [options] directory`

Creates a new directory.

`rmdir [options] directory`

Deletes the specified directory if it is already empty.

`chown [options] username[:[group]] files`

Transfers ownership of a file to the user with the specified username.

`-R`

Changes files and directories in all subdirectories

`chgrp [options] groupname files`

Transfers the group ownership of a given file to the group with the specified group name. The file owner can change group ownership only if a member of both the current and the new group.

`chmod [options] mode files`

Changes the access permissions.

The mode parameter has three parts: group, access, and access type. group accepts the following characters:

u

User

g

Group

o

Others

For access, grant access with + and deny it with -.

The access type is controlled by the following options:

r

Read

w

Write

x

Execute—executing files or changing to the directory

s

Setuid bit—the application or program is started as if it were started by the owner of the file

As an alternative, a numeric code can be used. The four digits of this code are composed of the sum of the values 4, 2, and 1—the decimal result of a binary mask. The first digit sets the set user ID (SUID) (4), the set group ID (2), and the sticky (1) bits. The second digit defines the permissions of the owner of the file. The third digit defines the permissions of the group members and the last digit sets the permissions for all other users. The read permission is set with 4, the write permission with 2, and the permission for executing a file is set with 1. The owner of a file would usually receive a 6 or a 7 for executable files.

`gzip [parameters] files`

This program compresses the contents of files using complex mathematical algorithms. Files compressed in this way are given the extension `.gz` and need to be uncompressed before they can be used. To compress several files or even entire directories, use the `tar` command.

`-d`

Decompresses the packed `gzip` files so they return to their original size and can be processed normally (like the command `gunzip`)

`tar options archive files`

`tar` puts one or more files into an archive. Compression is optional. `tar` is a quite complex command with a number of options available. The most frequently used options are:

`-f`

Writes the output to a file and not to the screen as is usually the case

`-c`

Creates a new `tar` archive

`-r`

Adds files to an existing archive

`-t`

Outputs the contents of an archive

- u  
Adds files, but only if they are newer than the files already contained in the archive
- x  
Unpacks files from an archive (*extraction*)
- z  
Packs the resulting archive with `gzip`
- j  
Compresses the resulting archive with `bzip2`
- v  
Lists files processed

The archive files created by `tar` end with `.tar`. If the tar archive was also compressed using `gzip`, the ending is `.tgz` or `.tar.gz`. If it was compressed using `bzip2`, the ending is `.tar.bz2`.

#### `locate` patterns

This command is only available if you have installed the `findutils-locate` package. The `locate` command can find in which directory a specified file is located. If desired, use wild cards to specify filenames. The program is very fast, because it uses a database specifically created for the purpose (rather than searching through the entire file system). This very fact, however, also results in a major drawback: `locate` is unable to find any files created after the latest update of its database. The database can be generated by `root` with `updatedb`.

#### `updatedb` [options]

This command performs an update of the database used by `locate`. To include files in all existing directories, run the program as `root`. It also makes sense to place it in the background by appending an ampersand (&), so you can immediately continue working on the same command line (`updatedb &`). This command usually runs as a daily cron job (see `cron.daily`).

#### `find` [options]

With `find`, search for a file in a given directory. The first argument specifies the directory in which to start the search. The option `-name` must be followed by a search string, which may also include wild cards. Unlike `locate`, which uses a database, `find` scans the actual directory.

## Commands to Access File Contents

`file [options] [files]`

With `file`, detect the contents of the specified files.

`-Z`

Tries to look inside compressed files

`cat [options] files`

The `cat` command displays the contents of a file, printing the entire contents to the screen without interruption.

`-n`

Numbers the output on the left margin

`less [options] files`

This command can be used to browse the contents of the specified file. Scroll half a screen page up or down with **PgUp** and **PgDn** or a full screen page down with **Space**. Jump to the beginning or end of a file using **Home** and **End**. Press **Q** to exit the program.

`grep [options] searchstring files`

The `grep` command finds a specific search string in the specified files. If the search string is found, the command displays the line in which `searchstring` was found along with the filename.

`-i`

Ignores case

`-H`

Only displays the names of the relevant files, but not the text lines

`-n`

Additionally displays the numbers of the lines in which it found a hit

`-l`

Only lists the files in which `searchstring` does not occur

`diff [options] file1 file2`

The `diff` command compares the contents of any two files. The output produced by the program lists all lines that do not match. This is frequently used by



programmers who need only to send their program alterations and not the entire source code.

`-q`

Only reports whether the two files differ

`-u`

Produces a “unified” diff, which makes the output more readable

## File Systems

`mount [options] [device] mountpoint`

This command can be used to mount any data media, such as hard disks, CD-ROM drives, and other drives, to a directory of the Linux file system.

`-r`

Mount read-only

`-t filesystem`

Specify the file system, commonly `ext2` for Linux hard disks, `msdos` for MS-DOS media, `vfat` for the Windows file system, and `iso9660` for CDs

For hard disks not defined in the file `/etc/fstab`, the device type must also be specified. In this case, only `root` can mount it. If the file system needs to also be mounted by other users, enter the option `user` in the appropriate line in the `/etc/fstab` file (separated by commas) and save this change. Further information is available in the `mount(1)` man page.

`umount [options] mountpoint`

This command unmounts a mounted drive from the file system. To prevent data loss, run this command before taking a removable data medium from its drive. Normally, only `root` is allowed to run the commands `mount` and `umount`. To enable other users to run these commands, edit the `/etc/fstab` file to specify the option `user` for the relevant drive.

## 21.12.2 System Commands

The following section lists a few of the most important commands needed for retrieving system information and controlling processes and the network.

## System Information

`df [options] [directory]`

The `df` (disk free) command, when used without any options, displays information about the total disk space, the disk space currently in use, and the free space on all the mounted drives. If a directory is specified, the information is limited to the drive on which that directory is located.

`-h`

Shows the number of occupied blocks in gigabytes, megabytes, or kilobytes—in human-readable format

`-T`

Type of file system (ext2, nfs, etc.)

`du [options] [path]`

This command, when executed without any parameters, shows the total disk space occupied by files and subdirectories in the current directory.

`-a`

Displays the size of each individual file

`-h`

Output in human-readable form

`-s`

Displays only the calculated total size

`free [options]`

The command `free` displays information about RAM and swap space usage, showing the total and the used amount in both categories. See Section 19.1.6, “The `free` Command” (page 265) for more information.

`-b`

Output in bytes

`-k`

Output in kilobytes

`-m`

Output in megabytes

`date [options]`

This simple program displays the current system time. If run as `root`, it can also be used to change the system time. Details about the program are available in the `date(1)` man page.

## Processes

`top [options]`

`top` provides a quick overview of the currently running processes. Press `H` to access a page that briefly explains the main options for customizing the program.

`ps [options] [process_ID]`

If run without any options, this command displays a table of all your own programs or processes—those you started. The options for this command are not preceded by hyphen.

`aux`

Displays a detailed list of all processes, independent of the owner

`kill [options] process_ID`

Unfortunately, sometimes a program cannot be terminated in the normal way. In most cases, you should still be able to stop such a runaway program by executing the `kill` command, specifying the respective process ID (see `top` and `ps`). `kill` sends a *TERM* signal that instructs the program to shut itself down. If this does not help, the following parameter can be used:

`-9`

Sends a *KILL* signal instead of a *TERM* signal, bringing the specified process to an end in almost all cases

`killall [options] processname`

This command is similar to `kill`, but uses the process name (instead of the process ID) as an argument, killing all processes with that name.

## Network

`ping [options] hostname_or_IP address`

The `ping` command is the standard tool for testing the basic functionality of TCP/IP networks. It sends a small data packet to the destination host, requesting

an immediate reply. If this works, `ping` displays a message to that effect, which indicates that the network link is basically functioning.

`-cnumber`

Determines the total number of packages to send and ends after they have been dispatched (by default, there is no limitation set)

`-f`

*flood ping*: sends as many data packages as possible; a popular means, reserved for `root`, to test networks

`-ivalue`

Specifies the interval between two data packages in seconds (default: one second)

`host [options] hostname [server]`

The domain name system resolves domain names to IP addresses. With this tool, send queries to name servers (DNS servers).

`ssh [options] [user@]hostname [command]`

SSH is actually an Internet protocol that enables you to work on remote hosts across a network. SSH is also the name of a Linux program that uses this protocol to enable operations on remote computers.

## Miscellaneous

`passwd [options] [username]`

Users may change their own passwords at any time using this command. The administrator `root` can use the command to change the password of any user on the system.

`su [options] [username]`

The `su` command makes it possible to log in under a different username from a running session. Specify a username and the corresponding password. The password is not required from `root`, because `root` is authorized to assume the identity of any user. When using the command without specifying a username, you are prompted for the `root` password and change to the superuser (`root`). Use `su -` to start a login shell for a different user.

`halt [options]`

To avoid loss of data, you should always use this program to shut down your system.

`reboot [options]`

Does the same as `halt` except the system performs an immediate reboot.

`clear`

This command cleans up the visible area of the console. It has no options.

## 21.12.3 For More Information

There are many more commands than listed in this chapter. For information about other commands or more detailed information, the O'Reilly publication *Linux in a Nutshell* is recommended.



## Bash and Bash Scripts

These days many people use computers with a graphical user interface (GUI) like KDE or GNOME. Although they offer lots of features, their use is limited when it comes to the execution of automatical tasks. Shells are a good addition to GUIs and this chapter gives you an overview of some aspects of shells, in this case Bash.

### 22.1 What is “The Shell”?

Traditionally, *the* shell is Bash (Bourne again Shell). When this chapter speaks about “the shell” it means Bash. There are actually more available shells than Bash (ash, csh, ksh, zsh, ...), each employing different features and characteristics. If you need further information about other shells, search for *shell* in YaST.

#### 22.1.1 Knowing The Bash Configuration Files

A shell can be invoked as an:

1. interactive login shell. This is used when logging in to a machine, invoking Bash with the `--login` option or when logging in to a remote machine with SSH.
2. “ordinary” interactive shell. This is normally the case when starting xterm, konsole, gnome-terminal or similar tools.

3. non-interactive shell. This is used when invoking a shell script in the command-line.

Depending on which type of shell you use, different configuration files are being read. The following tables show the login and non-login shell configuration files.

**Table 22.1** *Bash Configuration Files for Login Shells*

| File                      | Description                                                                                     |
|---------------------------|-------------------------------------------------------------------------------------------------|
| <code>/etc/profile</code> | Do not modify this file, otherwise your modifications can be destroyed during your next update! |

**Table 22.2** *Bash Configuration Files for Non-Login Shells*

|                               |                                                                                                 |
|-------------------------------|-------------------------------------------------------------------------------------------------|
| <code>/etc/bash.bashrc</code> | Do not modify this file, otherwise your modifications can be destroyed during your next update! |
|-------------------------------|-------------------------------------------------------------------------------------------------|

Additionally, Bash uses some more files:

**Table 22.3** *Special Files for Bash*

| File                         | Description                                          |
|------------------------------|------------------------------------------------------|
| <code>~/.bash_history</code> | Contains a list of all commands you have been typing |

## 22.1.2 The Directory Structure

The following table provides a short overview of the most important higher-level directories that you find on a Linux system. Find more detailed information about the directories and important subdirectories in the following list.



**Table 22.4** *Overview of a Standard Directory Tree*

| Directory | Contents                                                 |
|-----------|----------------------------------------------------------|
| /         | Root directory—the starting point of the directory tree. |

The following list provides more detailed information and gives some examples of which files and subdirectories can be found in the directories:

`/bin`

Contains the basic shell commands that may be used both by `root` and by other users. These commands include `ls`, `mkdir`, `cp`, `mv`, `rm` and `rmdir`. `/bin` also contains `Bash`, the default shell in openSUSE.

`/boot`

Contains data required for booting, such as the boot loader, the kernel, and other data that is used before the kernel begins executing user-mode programs.

`/dev`

Holds device files that represent hardware components.

`/etc`

Contains local configuration files that control the operation of programs like the X Window System. The `/etc/init.d` subdirectory contains scripts that are executed during the boot process.

`/home/username`

Holds the private data of every user who has an account on the system. The files located here can only be modified by their owner or by the system administrator. By default, your e-mail directory and personal desktop configuration are located here in the form of hidden files and directories. KDE users find the personal configuration data for their desktop in `.kde4` and GNOME users find it in `.gconf`.

---

#### **NOTE: Home Directory in a Network Environment**

If you are working in a network environment, your home directory may be mapped to a directory in the file system other than `/home`.

---

`/lib`

Contains the essential shared libraries needed to boot the system and to run the commands in the root file system. The Windows equivalent for shared libraries are DLL files.

`/media`

Contains mount points for removable media, such as CD-ROMs, USB sticks and digital cameras (if they use USB). `/media` generally holds any type of drive except the hard drive of your system. As soon as your removable medium has been inserted or connected to the system and has been mounted, you can access it from here.

`/mnt`

This directory provides a mount point for a temporarily mounted file system. `root` may mount file systems here.

`/opt`

Reserved for the installation of third-party software. Optional software and larger add-on program packages can be found here.

`/root`

Home directory for the `root` user. The personal data of `root` is located here.

`/sbin`

As the `s` indicates, this directory holds utilities for the superuser. `/sbin` contains the binaries essential for booting, restoring and recovering the system in addition to the binaries in `/bin`.

`/srv`

Holds data for services provided by the system, such as FTP and HTTP.

`/tmp`

This directory is used by programs that require temporary storage of files.

---

### **IMPORTANT: Cleaning up `/tmp` at Boot Time**

Data stored in `/tmp` are not guaranteed to survive a system reboot. It depends, for example, on settings in `/etc/sysconfig/cron` .

---

`/usr`

`/usr` has nothing to do with users, but is the acronym for UNIX system resources. The data in `/usr` is static, read-only data that can be shared among var-

ious hosts compliant with the Filesystem Hierarchy Standard (FHS). This directory contains all application programs and establishes a secondary hierarchy in the file system. KDE4 and GNOME are also located here. `/usr` holds a number of subdirectories, such as `/usr/bin` , `/usr/sbin` , `/usr/local` , and `/usr/share/doc` .

`/usr/bin`

Contains generally accessible programs.

`/usr/sbin`

Contains system daemons and programs reserved for the system administrator that are not essential for booting the system.

`/usr/local`

In this directory the system administrator can install local, distribution-independent extensions.

`/usr/share/doc`

Holds various documentation files and the release notes for your system. In the `manual` subdirectory find an online version of this manual. If more than one language is installed, this directory may contain versions of the manuals for different languages.

Under `packages` find the documentation included in the software packages installed on your system. For every package, a subdirectory `/usr/share/doc/packages/ packagename` is created that often holds README files for the package and sometimes examples, configuration files or additional scripts.

If HOWTOs are installed on your system `/usr/share/doc` also holds the `howto` subdirectory in which to find additional documentation on many tasks related to the setup and operation of Linux software.

`/var`

Whereas `/usr` holds static, read-only data, `/var` is for data which is written during system operation and thus is variable data, such as log files or spooling data. For an overview of the most important log files you can find under `/var/log/` , refer to Table “Log Files” (↑Start-Up).

`/windows`

Only available if you have both Microsoft Windows and Linux installed on your system. Contains the Windows data available on the Windows partition of your

system. Whether you can edit the data in this directory depends on the file system your Windows partition uses. If it is FAT32, you can open and edit the files in this directory. For NTFS, openSUSE also includes write access support. However, the driver for the NTFS-3g file system has limited functionality. Learn more in Section 37.4, “Accessing Files on Different OS on the Same Computer” (page 589).

## 22.2 Writing Shell Scripts

Shell scripts are a convenient way of doing all sorts of tasks: collecting data, searching for a word or phrase in a text and many other useful things. The following example shows a small shell script that prints a text:

### **Example 22.1** *A Shell Script Printing a Text*

```
#!/bin/sh ❶
Output the following line: ❷
echo "Hello World" ❸
```

- ❶ The first line begins with the *Shebang* characters (`#!`) which is an indicator that this file is a script. The script is executed with the specified interpreter after the Shebang, in this case `/bin/sh`.
- ❷ The second line is a comment beginning with the hash sign. It is recommended to comment difficult lines to remember what they do.
- ❸ The third line uses the built-in command `echo` to print the corresponding text.

Before you can run this script you need some prerequisites:

1. Every script should contain a Shebang line (this is already the case with our example above). If a script does not have this line, you have to call the interpreter manually.
2. Save the script to a directory where the shell can find it. The search path in a shell is determined by the environment variable `PATH`. Usually a normal user does not have write access to `/usr/bin`. Therefore, it is recommended to save your scripts in the users' directory `~/bin/`. The above example gets the name `hello.sh`.
3. The script needs executable permissions. Set the permissions with the following command:

```
chmod +x ~/bin/hello.sh
```

If you have fulfilled all of the above prerequisites, you can execute the script in the following ways:

1. **As Absolute Path** The script can be executed with an absolute path. In our case, it is `~/bin/hello.sh`.
2. **Everywhere** If the `PATH` environment variable contains the directory where the script is located, you can execute the script just with `hello.sh`.

## 22.3 Redirecting Command Events

Each command can use three channels, either for input or output:

- **Standard Output** This is the default output channel. Whenever a command prints something, it uses the standard output channel.
- **Standard Input** If a command needs input from users or other commands, it uses this channel.
- **Standard Error** Commands use this channel for error reporting.

To redirect these channels, there are the following possibilities:

Command > File

Saves the output of the command into a file, an existing file will be deleted. For example, the `ls` command writes its output into the file `listing.txt` :

```
ls > listing.txt
```

Command >> File

Appends the output of the command to a file. For example, the `ls` command appends its output to the file `listing.txt` :

```
ls >> listing.txt
```

Command < File

Reads the file as input for the given command. For example, the `read` command reads in the content of the file to the variable `a`:

```
read a < foo
```

Command1 | Command2

Redirects the output of the left command as input for the right command. For example, the `cat` command outputs the content of the `/proc/cpuinfo` file.

This output is used by `grep` to filter only those lines which contain `cpu`:

```
cat /proc/cpuinfo | grep cpu
```

Every channel has a *file descriptor*: 0 (zero) for standard input, 1 for standard output and 2 for standard error. It is allowed to insert this file descriptor before a `<` or `>` character. For example, the following line searches for a file starting with `foo`, but suppresses its errors by redirecting it to `/dev/null`, the garbage bin:

```
find / -name "foo*" 2>/dev/null
```

## 22.4 Using Aliases

An alias is a shortcut definition of one or more commands. This is useful if you have commands which are hard to remember or with lots of options. The syntax for an alias is:

```
alias NAME=DEFINITION
```

For example, the following line defines an alias `lt` which outputs a long listing (option `-l`), sorts it by modification time (`-t`) and prints it in reverse order while sorting (`-r`):

```
alias lt='ls -ltr'
```

To view all alias definitions, use `alias`. Remove your alias with `unalias`.

## 22.5 Using Variables in Bash

A shell variable can be global or local. Global variables, or environment variables, can be accessed in all shells. In contrast, local variables are visible in the current shell only.

To view all environment variables, use the `printenv` command. To see the value of a variable, insert the name of your variable as an argument:

```
printenv PATH
```

A variable, be it global or local, can also be viewed with `echo`:

```
echo $PATH
```

To set a local variable, use a variable name followed by the equal sign, followed by the value:

```
PROJECT="SLED"
```

Do not insert spaces around the equal sign, otherwise you get an error. To set an environment variable, use `export`:

```
export NAME="tux"
```

To remove a variable, use `unset`:

```
unset NAME
```

The following table contains some common environment variables which can be used in you shell scripts:

**Table 22.5** *Useful Environment Variables*

---

|      |                                        |
|------|----------------------------------------|
| HOME | the home directory of the current user |
|------|----------------------------------------|

---

## 22.5.1 Using Argument Variables

For example, if you have the script `foo.sh` you can execute it like this:

```
foo.sh "Tux Penguin" 2000
```

To access all the arguments which are passed to your script, you need positional parameters. These is `$1` for the first argument, `$2` for the second, and so on. You can have up to nine parameters. To get the script name, use `$0`.

The following script `foo.sh` prints all arguments from 1 to 4:

```
#!/bin/sh
echo \"$1\" \"$2\" \"$3\" \"$4\"
```

If you execute this script with the above arguments, you get:

```
"Tux Penguin" "2000" "" ""
```

## 22.5.2 Using Variable Substitution

Variable substitutions apply a pattern to the content of a variable either from the left or right side. The following list contains the possible syntax forms:

`${VAR#pattern}`

removes the shortest possible match from the left:

```
file=/home/tux/book/book.tar.bz2
echo ${file#*/}
home/tux/book/book.tar.bz2
```

`${VAR##pattern}`

removes the longest possible match from the left:

```
file=/home/tux/book/book.tar.bz2
echo ${file##*/}
book.tar.bz2
```

`${VAR%pattern}`

removes the shortest possible match from the right:

```
file=/home/tux/book/book.tar.bz2
echo ${file%.*}
/home/tux/book/book.tar
```

`${VAR%%pattern}`

removes the longest possible match from the right:

```
file=/home/tux/book/book.tar.bz2
echo ${file%%.*}
/home/tux/book/book
```

`${VAR/pattern_1/pattern_2}`

substitutes the content of *VAR* from the *pattern\_1* with *pattern\_2*:

```
file=/home/tux/book/book.tar.bz2
echo ${file/tux/wilber}
/home/wilber/book/book.tar.bz2
```

## 22.6 Grouping And Combining Commands

Shells allow you to concatenate and group commands for conditional execution. Each command returns an exit code which determines the success or failure of its operation. If it is 0 (zero), the command was successful. Everything else marks an error which is specific to the command.

The following list shows, how commands can be grouped:

`Command1 ; Command2`

executes the commands in sequential order. The exit code is not checked. The following line displays the content of the file with `cat` and then prints its file properties with `ls` regardless of their exit codes:



```
cat filelist.txt ; ls -l filelist.txt
```

Command1 && Command2

runs the right command, if the left command was successful (logical AND). The following line displays the content of the file and prints its file properties only, when the previous command was successful (compare it with the previous entry in this list):

```
cat filelist.txt && ls -l filelist.txt
```

Command1 || Command2

runs the right command, when the left command has failed (logical OR). The following line creates only a directory in `/home/wilber/bar` when the creation of the directory in `/home/tux/foo` has failed:

```
mkdir /home/tux/foo || mkdir /home/wilber/bar
```

```
funcname() { ... }
```

creates a shell function. You can use the positional parameters to access its arguments. The following line defines the function `hello` to print a short message:

```
hello() { echo "Hello $1"; }
```

Call this function like this:

```
hello Tux
```

which prints:

```
Hello Tux
```

## 22.7 Working with Common Flow Constructs

To control the flow of your script, a shell has `while`, `if`, `for` and `case` constructs.

### 22.7.1 The if Control Command

The `if` command is used to check expressions. For example, the following code tests whether the current user is Tux:

```
if test $USER = "tux" ; then
 echo "Hello Tux."
else
```

```
 echo "You are not Tux."
fi
```

The test expression can be as complex or simple as possible. The following expression checks if the file `foo.txt` exists:

```
if test -e /tmp/foo.txt ; then
 echo "Found foo.txt"
fi
```

The test expression can also be abbreviated in angled brackets:

```
if [-e /tmp/foo.txt] ; then
 echo "Found foo.txt"
fi
```

Find more useful expressions at <http://bash.cyberciti.biz/guide/If..else..fi>.

## 22.7.2 Creating Loops With the For Command

The `for` loop allows you to execute commands to a list of entries. For example, the following code prints some information about PNG files in the current directory:

```
for i in *.png;
do
 ls -l $i
done
```

## 22.8 For More Information

Important information about Bash is provided in the man pages `man bash`. More about this topic can be found in the following list:

- <http://tldp.org/LDP/Bash-Beginners-Guide/html/index.html> —Bash Guide for Beginners
- <http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html>—BASH Programming - Introduction HOW-TO
- <http://tldp.org/LDP/abs/html/index.html> —Advanced Bash-Scripting Guide

- <http://www.grymoire.com/Unix/Sh.html>  
Shell

—Sh - the Bourne



## **Part V. 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 23.1, “Several Protocols in the TCP/IP Protocol Family” (page 341), 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>.

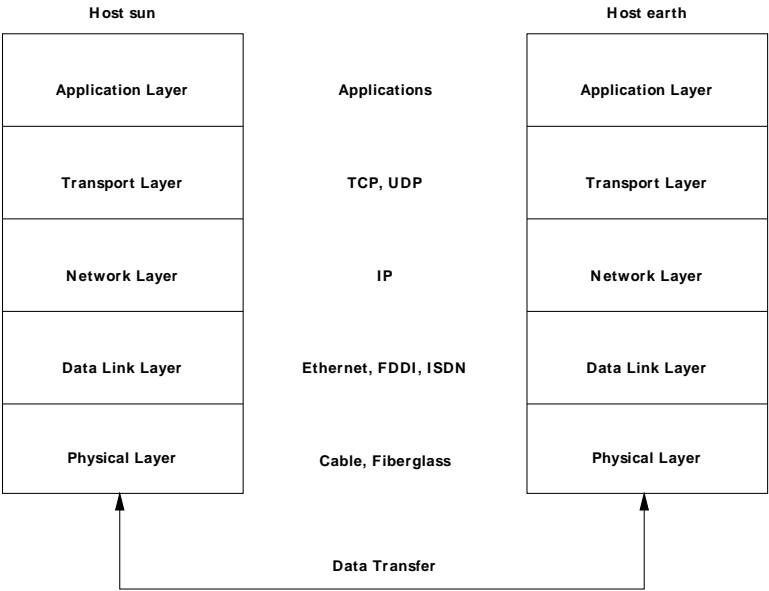
**Table 23.1** *Several Protocols in the TCP/IP Protocol Family*

| Protocol | Description                                                                                                         |
|----------|---------------------------------------------------------------------------------------------------------------------|
| TCP      | Transmission Control Protocol: a connection-oriented secure protocol. The data to transmit is first sent by the ap- |

| Protocol | Description                                                                                                                                                                                                                                                                                                                                                                |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          | 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. |

As shown in Figure 23.1, “Simplified Layer Model for TCP/IP” (page 342), 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 23.1** *Simplified Layer Model for TCP/IP*



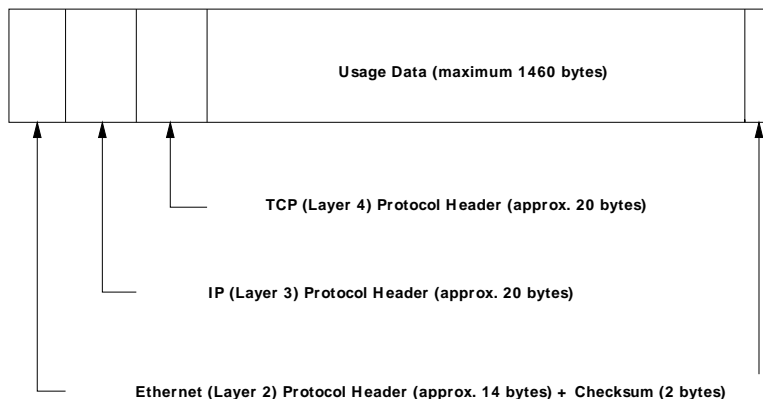


The diagram provides one or two examples for each layer. The layers are ordered according to *abstraction levels*. The lowest layer is very close to the hardware. The uppermost layer, however, is almost a complete abstraction from the hardware. Every layer has its own special function. The special functions 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 23.2, “TCP/IP Ethernet Packet” (page 343). The proof sum is located at the end of the packet, not at the beginning. This simplifies things for the network hardware.

**Figure 23.2** 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 ul-

timately 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.

## 23.1 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 23.2, “IPv6—The Next Generation Internet” (page 346).

### 23.1.1 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 23.1, “Writing IP Addresses” (page 344).

#### **Example 23.1** *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.

### 23.1.2 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 23.2, “Linking IP Addresses to the Netmask” (page 345). 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 23.2, “Linking IP Addresses to the Netmask” (page 345) the first net with 24 bits could also be written as 192.168.0.0/24.

### **Example 23.2** *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 23.2** *Specific Addresses*

| Address Type         | Description                                                                                                                                                                                              |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Base Network Address | This is the netmask AND any address in the network, as shown in Example 23.2, “Linking IP Addresses to the Netmask” (page 345) under <code>Result</code> . This address cannot be assigned to any hosts. |

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 23.3, “Private IP Address Domains” (page 346).

**Table 23.3** *Private IP Address Domains*

| Network/Netmask    | Domain   |
|--------------------|----------|
| 10.0.0.0/255.0.0.0 | 10.x.x.x |

## 23.2 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.

## 23.2.1 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 23.2.2, “Address Types and Structure” (page 349).

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 main-

tain 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 23.2.3, “Coexistence of IPv4 and IPv6” (page 353). Also, systems can rely on a *dual stack IP* technique to support both protocols at the same time, meaning that they have two network stacks that are completely separate, such that there is no interference between the two protocol versions.

### Custom Tailored Services through Multicasting

With IPv4, some services, such as SMB, need to broadcast their packets to all hosts in the local network. IPv6 allows a much more fine-grained approach by enabling servers to address hosts through *multicasting*—by addressing a number of hosts as parts of a group (which is different from addressing all hosts through *broadcasting* or each host individually through *unicasting*). Which hosts are addressed as a group may depend on the concrete application. There are some

predefined groups to address all name servers (the *all name servers multicast group*), for example, or all routers (the *all routers multicast group*).

## 23.2.2 Address Types and Structure

As mentioned, the current IP protocol is lacking in two important aspects: there is an increasing shortage of IP addresses and configuring the network and maintaining the routing tables is becoming a more complex and burdensome task. IPv6 solves the first problem by expanding the address space to 128 bits. The second one is countered by introducing a hierarchical address structure, combined with sophisticated techniques to allocate network addresses, as well as *multihoming* (the ability to assign several addresses to one device, giving access to several networks).

When dealing with IPv6, it is useful to know about three different types of addresses:

### Unicast

Addresses of this type are associated with exactly one network interface. Packets with such an address are delivered to only one destination. Accordingly, unicast addresses are used to transfer packets to individual hosts on the local network or the Internet.

### Multicast

Addresses of this type relate to a group of network interfaces. Packets with such an address are delivered to all destinations that belong to the group. Multicast addresses are mainly used by certain network services to communicate with certain groups of hosts in a well-directed manner.

### Anycast

Addresses of this type are related to a group of interfaces. Packets with such an address are delivered to the member of the group that is closest to the sender, according to the principles of the underlying routing protocol. Anycast addresses are used to make it easier for hosts to find out about servers offering certain services in the given network area. All servers of the same type have the same anycast address. Whenever a host requests a service, it receives a reply from the server with the closest location, as determined by the routing protocol. If this server should fail for some reason, the protocol automatically selects the second closest server, then the third one, and so forth.

An IPv6 address is made up of eight four-digit fields, each representing 16 bits, written in hexadecimal notation. They are 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 23.3, “Sample IPv6 Address” (page 350), where all three lines represent the same address.

**Example 23.3** *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 23.4, “IPv6 Address Specifying the Prefix Length” (page 350), 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 23.4** *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 23.4, “Various IPv6 Prefixes” (page 350).

**Table 23.4** *Various IPv6 Prefixes*

| Prefix (hex) | Definition                                                                                                                                                                                                                                                                                       |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 00           | IPv4 addresses and IPv4 over IPv6 compatibility addresses. These are used to maintain compatibility with IPv4. Their use still requires a router able to translate IPv6 packets into IPv4 packets. Several special addresses, such as the one for the loopback device, have this prefix as well. |



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 23.2.3, “Coexistence of IPv4 and IPv6” (page 353)) to allow IPv4 and IPv6 hosts to communicate with others operating in a pure IPv4 environment.

#### IPv4 Addresses Mapped to IPv6

This type of address specifies a pure IPv4 address in IPv6 notation.

## Local Addresses

There are two address types for local use:

### link-local

This type of address can only be used in the local subnetwork. Packets with a source or target address of this type should not be routed to the Internet or other subnetworks. These addresses contain a special prefix ( $\text{fe80}::/10$ ) and the interface ID of the network card, with the middle part consisting of 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 auto-configuration* 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 pack-

ets destined to the care-of address are directly transferred to the mobile host without any special detours.

## 23.2.3 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 23.2.2, “Address Types and Structure” (page 349)).

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 communica-

tion 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.

## 23.2.4 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 “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.

## 23.2.5 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).

## 23.3 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 “Configuring Hostname and DNS” (page 365). Setting up your own name server is described in Chapter 25, *The Domain Name System* (page 399).

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 name-server 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> .

---

## **23.4 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 23.6, “Configuring a Network Connection Manually” (page 379).

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.

## 23.4.1 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 “Configuring Global Networking Options” (page 358).

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 “Configuring an Undetected Network Card” (page 364). If you want to change the configuration of an already configured card, see Section “Changing the Configuration of a Network Card” (page 359).

The *Hostname/DNS* tab allows to set the hostname of the machine and name the servers to be used. For more information, see Section “Configuring Hostname and DNS” (page 365).

The *Routing* tab is used for the configuration of routing. See Section “Configuring Routing” (page 367) for more information.

**Figure 23.3** *Configuring Network Settings*



## 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 35, *Using NetworkManager* (page 557).

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 autoload 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*.

## 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 34.5, “Configuration with YaST” (page 543).

### 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 set-up 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 “Configuring Hostname and DNS” (page 365). To configure a gateway, proceed as described in Section “Configuring Routing” (page 367).

## 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 14, *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*.

## 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 “Changing the Configuration of a Network Card” (page 359).
- 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*.

## 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*.

## 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 `multi-pathd` do.

---

- 4 If the system is a router, enable the *IP Forwarding* option in the *Network Settings*.
- 5 To activate the configuration, click *OK*.

## 23.4.2 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*.

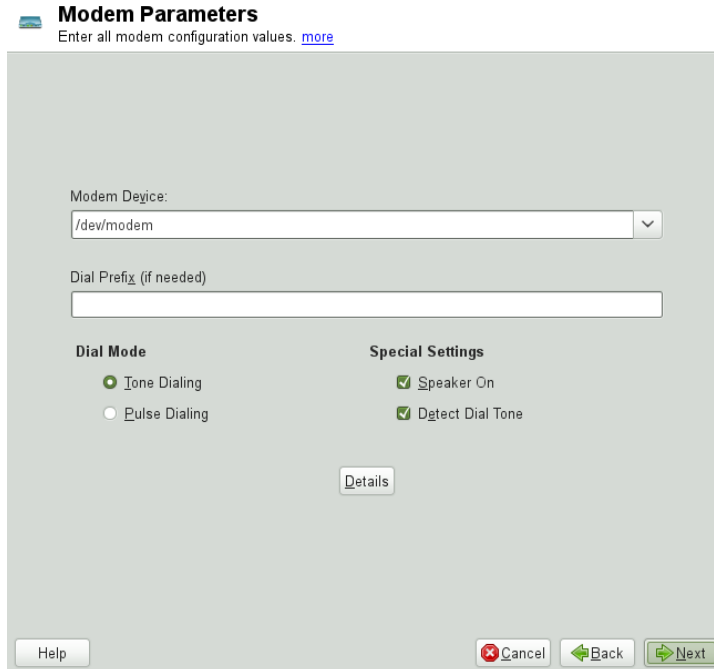
---

### TIP: CDMA and GPRS Modems

Configure supported CDMA and GPRS modems with the YaST *Modem* module just as you would configure regular modems.

---

**Figure 23.4** *Modem Configuration*



**Modem Parameters**  
Enter all modem configuration values. [more](#)

Modem Device:

Dial Prefix (if needed)

**Dial Mode**

☒ Tone Dialing  
☐ Pulse Dialing

**Special Settings**

☒ Speaker On  
☒ Detect Dial Tone

If you are behind a private branch exchange (PBX), you may need to enter a dial prefix. This is often a zero. Consult the instructions that came with the PBX to find out. Also select whether to use tone or pulse dialing, whether the speaker should be on and whether the modem should wait until it detects a dial tone. The last option should not be enabled if the modem is connected to an exchange.

Under *Details*, set the baud rate and the modem initialization strings. Only change these settings if your modem was not detected automatically or if it requires special settings for data transmission to work. This is mainly the case with ISDN terminal adapters. Leave this dialog by clicking *OK*. To delegate control over the modem to the normal user without root permissions, activate *Enable Device Control for Non-root User via KInternet*. In this way, a user without administrator permissions can activate or deactivate an interface. Under *Dial Prefix Regular Expression*, specify a regular expression. The *Dial Prefix* in KInternet, which can be modified by the normal user, must match this regular expression. If this field is left empty, the user cannot set a different *Dial Prefix* without administrator permissions.

In the next dialog, select the ISP. To choose from a predefined list of ISPs operating in your country, select *Country*. Alternatively, click *New* to open a dialog in which to provide the data for your ISP. This includes a name for the dial-up connection and ISP as well as the login and password provided by your ISP. Enable *Always Ask for Password* to be prompted for the password each time you connect.

In the last dialog, specify additional connection options:

#### *Dial on Demand*

If you enable *Dial on Demand*, set at least one name server. Use this feature only if your Internet connection is inexpensive, because there are programs that periodically request data from the Internet.

#### *Modify DNS when Connected*

This option is enabled by default, with the effect that the name server address is updated each time you connect to the Internet.

#### *Automatically Retrieve DNS*

If the provider does not transmit its domain name server after connecting, disable this option and enter the DNS data manually.

#### *Automatically Reconnect*

If this options is enabled, the connection is automatically reestablished after failure.

#### *Ignore Prompts*

This option disables the detection of any prompts from the dial-up server. If the connection build-up is slow or does not work at all, try this option.

### *External Firewall Interface*

Selecting this option activates the 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*.

## **23.4.3 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 23.5** 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:

---

**ISDN Protocol**

☒ Euro-ISDN (EDSSI)  
☐ ITR6  
☐ Leased Line  
☐ NI1

Country:

Code:

Area Code:

Dial Prefix:

☒ Start ISDN Log

Activate device:

In the next dialog, shown in Figure 23.5, “ISDN Configuration” (page 371), 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 Hotplug*, used for PCMCIA or USB devices, loads the driver after the device is plugged in. When finished with these settings, select *OK*.

In the next dialog, specify the interface type for your ISDN card and add ISPs to an existing interface. Interfaces may be either the *SyncPPP* or the *RawIP* type, but most ISPs operate in the *SyncPPP* mode, which is described below.

**Figure 23.6** 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 1TR6 protocol for internal calls. Their MSN is called EAZ and usually corresponds to the direct-dial number. For the configuration under Linux, it should be sufficient to enter the last digit of the EAZ. As a last resort, try each of the digits from 1 to 9.

For the connection to be terminated just before the next charge unit is due, enable *ChargeHUP*. However, remember that may not work with every ISP. You can also enable channel bundling (multilink PPP) by selecting the corresponding option. Finally, you can enable 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*.

## 23.4.4 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.

## 23.4.5 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 23.4.1, “Configuring the Network Card with YaST” (page 357)). 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 23.7, “DSL Configuration” (page 376)), 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 23.7** 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.

# 23.5 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.

## 23.5.1 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 35.2, “Enabling NetworkManager” (page 558) 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 35, *Using NetworkManager* (page 557).

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 within your desktop environment using an applet. NetworkManager also makes it possible to change and configure wireless card connections without requiring `root` privileges. For this reason, NetworkManager is the ideal solution for a mobile workstation.

Traditional configuration with ifup also provides some ways to switch, stop or start the connection with or without user intervention, like user-managed devices. 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), dial-up and wired networks using DHCP and static configuration. They also support connection through VPN.

NetworkManager tries to keep your computer connected at all times using the best connection available. If 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.

## 23.5.2 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 can be found 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 35.3, “Configuring Network Connections” (page 558).

## 23.5.3 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. For an introduction to PolicyKit and details on how to use it, refer to Chapter 9, *PolicyKit* (↑Security Guide).

The following table gives an overview of the PolicyKit identifiers related to NetworkManager:

**Table 23.5** *PolicyKit Identifiers for NetworkManager*

| Identifier                                            | Description                         |
|-------------------------------------------------------|-------------------------------------|
| org.freedesktop.NetworkManager.enable-disable-network | Enable or disable system networking |

## 23.6 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 23.6, “Manual Network Configuration Scripts” (page 380) summarizes the most important scripts involved in the network configuration.

**Table 23.6** *Manual Network Configuration Scripts*

| Command                                                         | Function                                                                                                                                                                    |
|-----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ifup</code> , <code>ifdown</code> , <code>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. |

For more information about `udev` and persistent device names, see Chapter 20, *Dynamic Kernel Device Management with `udev`* (page 273).

## 23.6.1 Configuration Files

This section provides an overview of the network configuration files and explains their purpose and the format used.

### **`/etc/sysconfig/network/ifcfg-*`**

These files contain the configurations for network interfaces. They include information such as the start mode and the IP address. Possible parameters are described in the manual page of `ifup`. Additionally, 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 “`/etc/sysconfig/network/config`, `/etc/sysconfig/network/dhcp`, and `/etc/sysconfig/network/wireless`” (page 380).

### **`/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.

## **`/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
```

## **/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 23.5, “`/etc/resolv.conf`” (page 383) 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 using 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 23.5** */etc/resolv.conf*

```
Our domain
search example.com
#
We use dns.example.com (192.168.1.116) as nameserver
nameserver 192.168.1.116
```

## **/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`.

## **/etc/hosts**

In this file, shown in Example 23.6, “`/etc/hosts` ” (page 384), 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 23.6** */etc/hosts*

```
127.0.0.1 localhost
192.168.2.100 jupiter.example.com jupiter
192.168.2.101 venus.example.com venus
```

## **/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 23.7, “`/etc/networks` ” (page 385).

### Example 23.7 /etc/networks

```
loopback 127.0.0.0
localnet 192.168.0.0
```

## /etc/host.conf

Name resolution—the translation of host and network names via the *resolver* library—is controlled by this file. This file is only used for programs linked to libc4 or libc5. For current glibc programs, refer to the settings in `/etc/nsswitch.conf`. A parameter must always stand alone in its own line. Comments are preceded by a `#` sign. Table 23.7, “Parameters for `/etc/host.conf`” (page 385) shows the parameters available. A sample `/etc/host.conf` is shown in Example 23.8, “`/etc/host.conf`” (page 385).

**Table 23.7** 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): |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|

---

### Example 23.8 /etc/host.conf

```
We have named running
order hosts bind
Allow multiple address
multi on
```

## /etc/nsswitch.conf

The introduction of the GNU C Library 2.0 was accompanied by the introduction of the *Name Service Switch* (NSS). Refer to the `nsswitch.conf(5)` man page and *The GNU C Library Reference Manual* for details.

The order for queries is defined in the file `/etc/nsswitch.conf`. A sample `nsswitch.conf` is shown in Example 23.9, “`/etc/nsswitch.conf`” (page 386). 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 23.9** */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 23.8, “Databases Available via */etc/nsswitch.conf*” (page 386). The configuration options for NSS databases are listed in Table 23.9, “Configuration Options for NSS “Databases”” (page 386).

**Table 23.8** *Databases Available via /etc/nsswitch.conf*

|         |                                                           |
|---------|-----------------------------------------------------------|
| aliases | Mail aliases implemented by send-mail; see man 5 aliases. |
|---------|-----------------------------------------------------------|

**Table 23.9** *Configuration Options for NSS “Databases”*

|       |                                                  |
|-------|--------------------------------------------------|
| files | directly access files, for example, /etc/aliases |
|-------|--------------------------------------------------|

**/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 `nsd` 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 `nsd` with the command `rcnsd restart`.

## **/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).

## **23.6.2 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.

## **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`.

# 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 23.10, “Output of the Command `ping`” (page 389), 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 23.10** *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
```

---

## 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 23.11, “Output of the `ifconfig` Command” (page 391), `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 23.11** *Output of the ifconfig Command*

```
eth0 Link encap:Ethernet HWaddr 00:08:74:98:ED:51
 inet6 addr: fe80::208:74ff:fe98:ed51/64 Scope:Link
 UP BROADCAST MULTICAST MTU:1500 Metric:1
 RX packets:634735 errors:0 dropped:0 overruns:4 frame:0
 TX packets:154779 errors:0 dropped:0 overruns:0 carrier:1
 collisions:0 txqueuelen:1000
 RX bytes:162531992 (155.0 Mb) TX bytes:49575995 (47.2 Mb)
 Interrupt:11 Base address:0xec80

lo Link encap:Local Loopback
 inet addr:127.0.0.1 Mask:255.0.0.0
 inet6 addr: ::1/128 Scope:Host
 UP LOOPBACK RUNNING MTU:16436 Metric:1
 RX packets:8559 errors:0 dropped:0 overruns:0 frame:0
 TX packets:8559 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:0
 RX bytes:533234 (520.7 Kb) TX bytes:533234 (520.7 Kb)

wlan1 Link encap:Ethernet HWaddr 00:0E:2E:52:3B:1D
 inet addr:192.168.2.4 Bcast:192.168.2.255 Mask:255.255.255.0
 inet6 addr: fe80::20e:2eff:fe52:3b1d/64 Scope:Link
 UP BROADCAST NOTRAILERS RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:50828 errors:0 dropped:0 overruns:0 frame:0
 TX packets:43770 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:45978185 (43.8 Mb) TX bytes:7526693 (7.1 MB)
```

For more options and information about using `ifconfig`, enter `ifconfig -h` or see the `ifconfig (8) man` page.

## **Configuring Routing with route**

`route` is a program for manipulating the IP routing table. You can use it to view your routing configuration and 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 23.12** *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.

## 23.6.3 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 23.10, “Some Start-Up Scripts for Network Programs” (page 392).

**Table 23.10** *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. |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|

## 23.7 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 connec-

tion 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.

## 23.7.1 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.

## 23.7.2 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).

---

## 24.1 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.

## 24.2 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.

## 24.3 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.

## 24.4 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 43). 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.

## 24.5 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`.

## 24.6 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 host-name `jupiter`, for example. Before setting up your own name server, read the general information about DNS in Section 23.3, “Name Resolution” (page 355). The following configuration examples refer to BIND.

## 25.1 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.

## 25.2 Installation

To install a DNS server, start YaST and select *Software > Software Management*. Choose *Filter > Patterns* and select *DHCP and DNS Server*. Confirm the installation of the dependent packages to finish the installation process.

## 25.3 Configuration with YaST

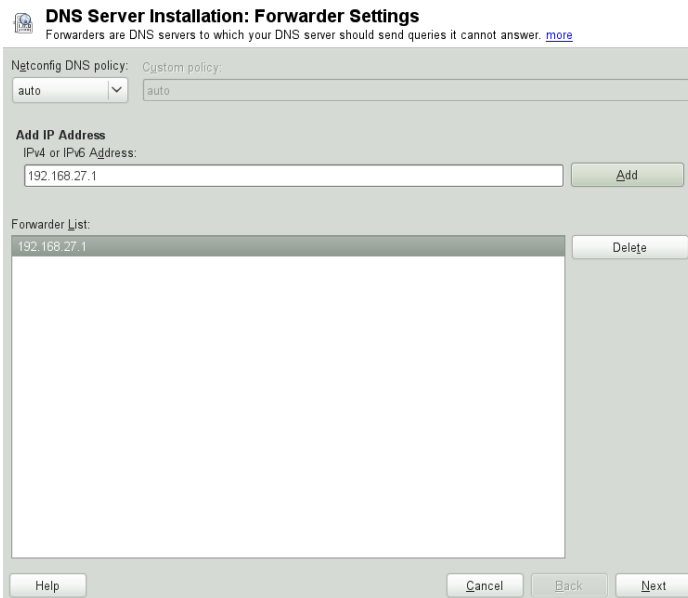
Use the YaST DNS module to configure a DNS server for the local network. When starting the module for the first time, a wizard starts, prompting you to make a few decisions concerning administration of the server. Completing this initial setup produces a basic server configuration. Use the expert mode to deal with more advanced configuration tasks, such as setting up ACLs, logging, TSIG keys, and other options.

## 25.3.1 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 25.1, “DNS Server Installation: Forwarder Settings” (page 401), 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`.

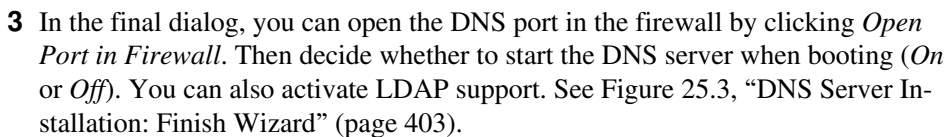
**Figure 25.1** DNS Server Installation: Forwarder Settings



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 25.6, “Zone Files” (page 416). 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 25.2** *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 403

## 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`.

## Basic Options

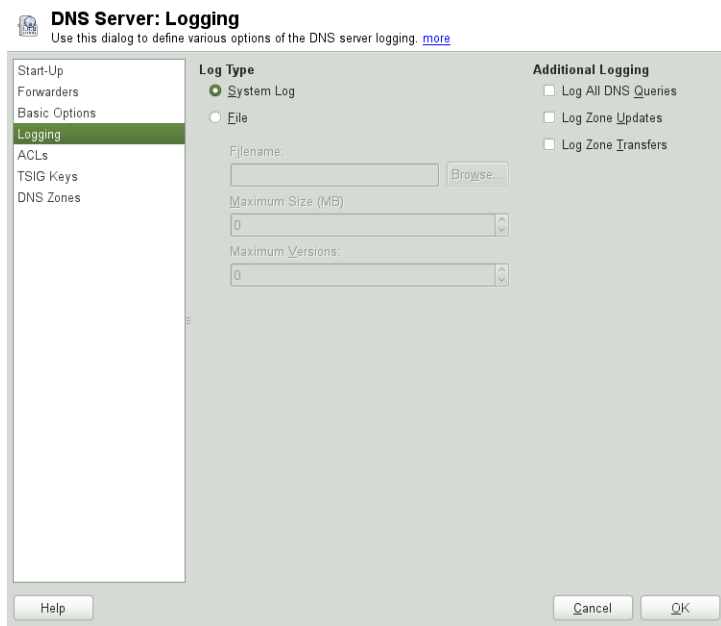
In this section, set basic server options. From the *Option* menu, select the desired item then specify the value in the corresponding entry field. Include the new entry by selecting *Add*.

## Logging

To set what the DNS server should log and how, select *Logging*. Under *Log Type*, specify where the DNS server should write the log data. Use the systemwide log file `/var/log/messages` by selecting *System Log* or specify a different file by selecting *File*. In the latter case, additionally specify a name, the maximum file size in megabytes and the number of 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 25.4, “DNS Server: Logging” (page 405).

**Figure 25.4** DNS Server: Logging



## 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 netmask) 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.

## TSIG Keys

The main purpose of TSIGs (transaction signatures) is to secure communications between DHCP and DNS servers. They are described in Section 25.8, “Secure Transactions” (page 420).

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*.

## 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.

## 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`.

## DNS Zones (Editing a Master Zone)

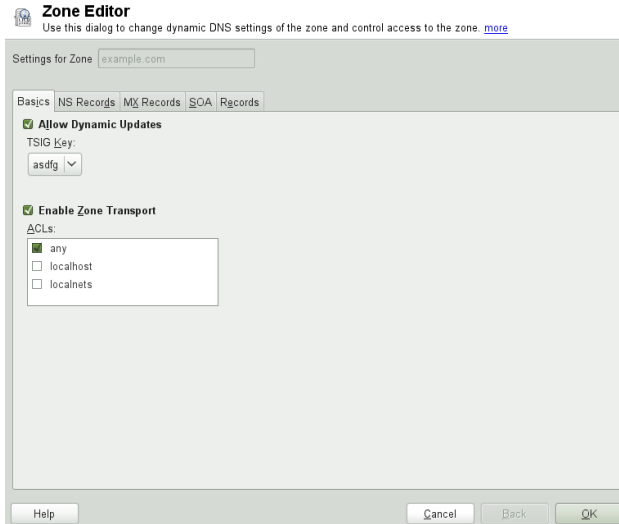
To edit a master zone, select *DNS Zones*, select the master zone from the table, and click *Edit*. The dialog consists of several pages: *Basics* (the one opened first), *NS Records*, *MX Records*, *SOA*, and *Records*.

The basic dialog, shown in Figure 25.5, “DNS Server: Zone Editor (Basics)” (page 407), 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 25.5** *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 25.6, “DNS Server: Zone Editor (NS Records)” (page 408).


**Figure 25.6** DNS Server: Zone Editor (NS Records)

The screenshot shows the 'Zone Editor' window for 'example.com'. It has tabs for 'Basics', 'NS Records', 'MX Records', 'SOA', and 'Records'. The 'NS Records' tab is active. Below the tabs, there is a 'Name Server to Add:' label, a text input field, and an 'Add' button. Below that is a 'Name Server List:' label, a large empty list box, and a 'Delete' button. At the bottom are 'Help', 'Cancel', 'Back', and 'OK' buttons.

### 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 25.7, “DNS Server: Zone Editor (MX Records)” (page 409).

**Figure 25.7** DNS Server: Zone Editor (MX Records)

**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 25.6, “The /var/lib/named/example.com.zone File” (page 417).

**Figure 25.8** 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:

### 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.

---

## 25.4 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 25.1, “Forwarding

Options in `named.conf`” (page 412) are just examples. Adjust these entries to your own setup.

**Example 25.1** *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`.

## 25.5 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 25.2, “A Basic `/etc/named.conf`” (page 413).

### Example 25.2 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";
};
```

## 25.5.1 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`.

## 25.5.2 Logging

What, how, and where logging takes place can be extensively configured in BIND. Normally, the default settings should be sufficient. Example 25.3, “Entry to Disable Logging” (page 415), shows the simplest form of such an entry and completely suppresses any logging.

### **Example 25.3** *Entry to Disable Logging*

```
logging {
 category default { null; };
};
```

## 25.5.3 Zone Entries

### **Example 25.4** *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 25.4, “Zone Entry for example.com” (page 415). 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 25.5, “Zone Entry for example.net” (page 415).

### **Example 25.5** *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.

## 25.6 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 25.6, “The `/var/lib/named/example.com.zone` File” (page 417).

### **Example 25.6** *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 in-

clude 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 address 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 25.7, “Reverse Lookup” (page 419).

**Example 25.7 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.

## 25.7 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 25.8, “Secure Transactions” (page 420).

## 25.8 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`.

## 25.9 DNS Security

DNSSEC, or DNS security, is described in RFC 2535. The tools available for DNSSEC are discussed in the BIND Manual.

A zone considered secure must have one or several zone keys associated with it. These are generated with `dnssec-keygen`, just like the host keys. The DSA encryption algorithm is currently used to generate these keys. The public keys generated should be included in the corresponding zone file with an `$INCLUDE` rule.

With the command `dnssec-makekeyset`, all keys generated are packaged into one set, which must then be transferred to the parent zone in a secure manner. On the parent, the set is signed with `dnssec-signkey`. The files generated by this command are then used to sign the zones with `dnssec-signzone`, which in turn generates the files to include for each zone in `/etc/named.conf`.

## 25.10 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 to 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.

## 26.1 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.

### 26.1.1 Initial Configuration (Wizard)

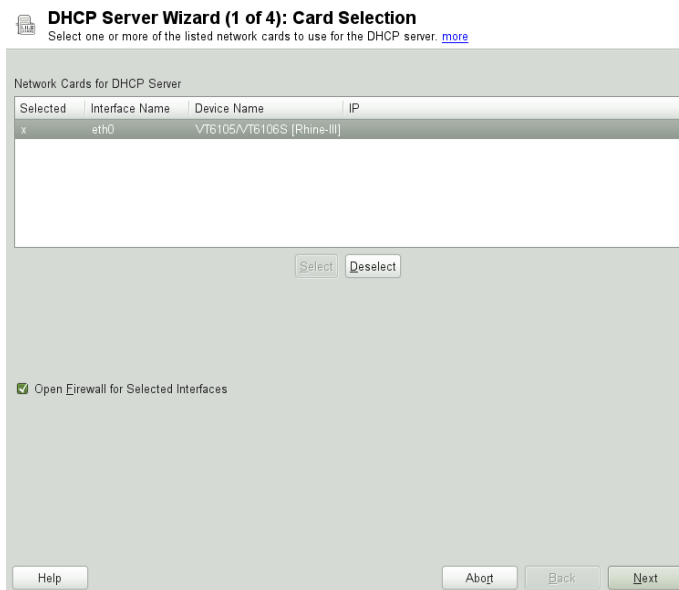
When the module is started for the first time, a wizard starts, prompting you to make a few basic decisions concerning server administration. Completing this initial set-

up 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 26.1, “DHCP Server: Card Selection” (page 425).


**Figure 26.1** 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 26.2, “DHCP Server: Global Settings” (page 426).

**Figure 26.2** *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 26.3, “DHCP Server: Dynamic DHCP” (page 427).

**Figure 26.3** *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     | 192.168.2.128    |

☐ Allow Dynamic BOOTP

**Lease Time**

| Default: | Units: | Maximum: | Units: |
|----------|--------|----------|--------|
| 4        | Hours  | 2        | 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 26.4, “DHCP Server: Start-Up” (page 427).

**Figure 26.4** *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](#)

## 26.2 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`.

## 26.3 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 26.1, “The Configuration File `/etc/dhcpd.conf`” (page 428).

### **Example 26.1** *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 25, *The Domain Name System* (page 399).
- 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 26.1, “The Configuration File `/etc/dhcpd.conf`” (page 428), 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 envi-

ronment so the daemon can find them. Normally, there is no need to worry about this because the command `redhcpd start` automatically copies the files.

## 26.3.1 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 26.2, “Additions to the Configuration File” (page 430), are added to the configuration file of Example 26.1, “The Configuration File `/etc/dhcpd.conf`” (page 428), the DHCP daemon always assigns the same set of data to the corresponding client.

### *Example 26.2 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 host-name `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.

## 26.3.2 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`.

## 26.4 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

# 27

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.

## 27.1 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.

## 27.1.1 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 27.1** 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 menu 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' (selected) with address '1.opensuse.pool.ntp.org'. At the bottom are buttons for 'Add', 'Edit', 'Delete', 'Display Log...', 'Help', 'Cancel', and 'OK'.

| Synchronization Type              | Address                 |
|-----------------------------------|-------------------------|
| Undisciplined Local Clock (LOCAL) |                         |
| Server                            | 1.opensuse.pool.ntp.org |

### *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 19.1.2, “The cron Package” (page 262).

### *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.

## 27.1.2 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 27.2** *YaST: NTP Server*



### Server

In the pull-down *Select* list (see Figure 27.2, “YaST: NTP Server” (page 435), 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*.

Another dialog enables you to select an NTP server. Activate *Use for Initial Synchronization* to trigger the synchronization of the time information between the server and the client when the system is booted. *Options* allows you to specify additional options for `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 27.3, “Advanced NTP Configuration: Security Settings” (page 437)). 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`.

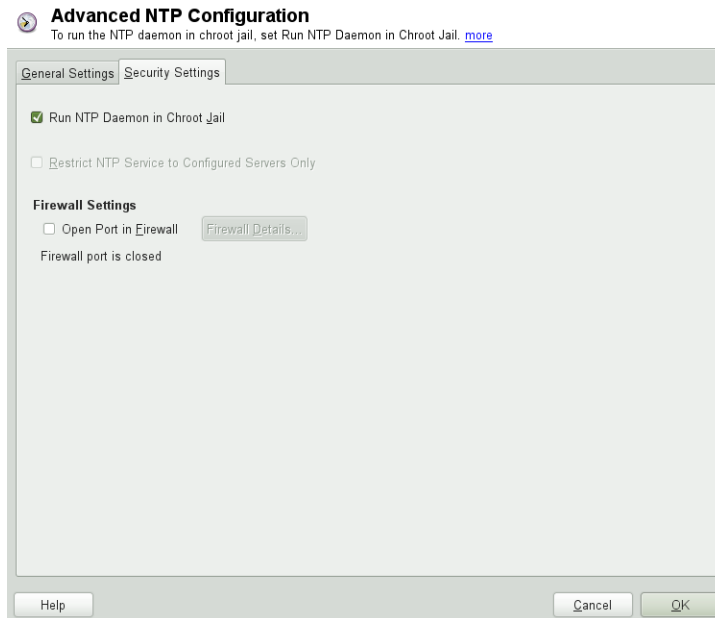
## Outgoing Broadcast

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 27.3** *Advanced NTP Configuration: Security Settings*



In the *Security Settings* tab (see Figure 27.3, “Advanced NTP Configuration: Security Settings” (page 437)), 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 re-

strictions 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.

## 27.2 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`.



## 27.3 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.

## 27.4 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/driver NN.html` (where *NN* is the number of the driver) provides information about the particular type of clock. For example, the “type 8” clock (radio clock over serial interface) requires an additional mode that specifies the clock more precisely. The Conrad DCF77 receiver module, for example, has mode 5. To use this clock as a preferred reference, specify the keyword `prefer`. The complete `server` line for a Conrad DCF77 receiver module would be:

```
server 127.127.8.0 mode 5 prefer
```

Other clocks follow the same pattern. Following the installation of the `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

# 28

Distributing and sharing file systems over a network is a common task in corporate environments. The proven NFS system works together with NIS, the yellow pages protocol. For a more secure protocol that works together with LDAP and may also be kerberized, 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.

Like NIS, NFS is a client/server system. However, a machine can be both—it can supply file systems over the network (export) and mount file systems from other hosts (import).

---

## **IMPORTANT: Need for DNS**

In principle, all exports can be made using IP addresses only. To avoid time-outs, you need a working DNS system. DNS is necessary at least for logging purposes, because the mountd daemon does reverse lookups.

---

## 28.1 Installing the Required Software

To configure your host as an NFS client, you do not need to install additional software. All needed packages are installed by default.

NFS server software is not part of the default installation. To install the NFS server software, start YaST and select *Software > Software Management*. Now choose *Filter > Patterns* and select *File Server* or use the *Search* option and search for *NFS Server*. Confirm the installation of the packages to finish the installation process.

## 28.2 Importing File Systems with YaST

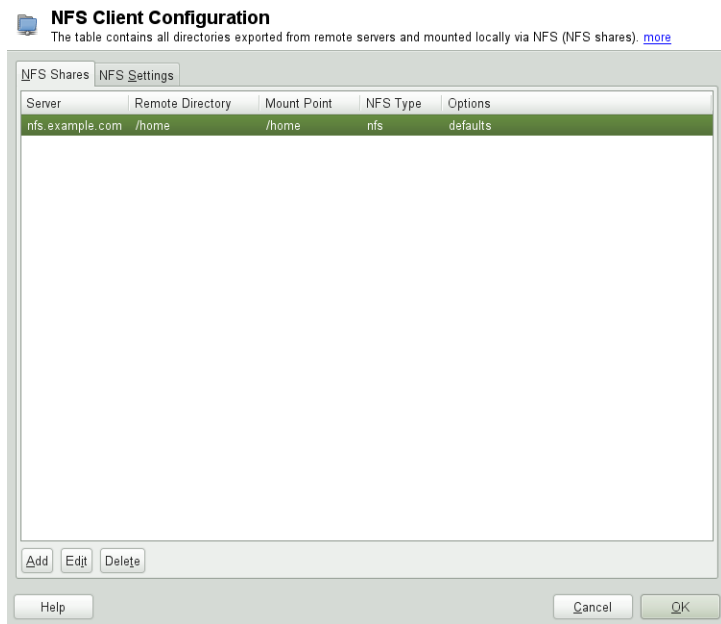
Authorized users can mount NFS directories from an NFS server into the local file tree using the YaST NFS client module. Click on *Add* and enter the hostname of the NFS server, the directory to import, and the mount point at which to mount this directory locally. The changes will take effect after *Finish* is clicked 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 enabled, 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 28.1, “NFS Client Configuration with YaST” (page 443).

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 28.1** *NFS Client Configuration with YaST*



## 28.3 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 `rcrpcbind 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
```

### 28.3.1 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 `rcautoofs start` as root. In this example, `/nfs-mounts/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 `autoofs` is running, the automounter must be restarted for the changes to take effect with `rcautoofs restart`.

## 28.3.2 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:

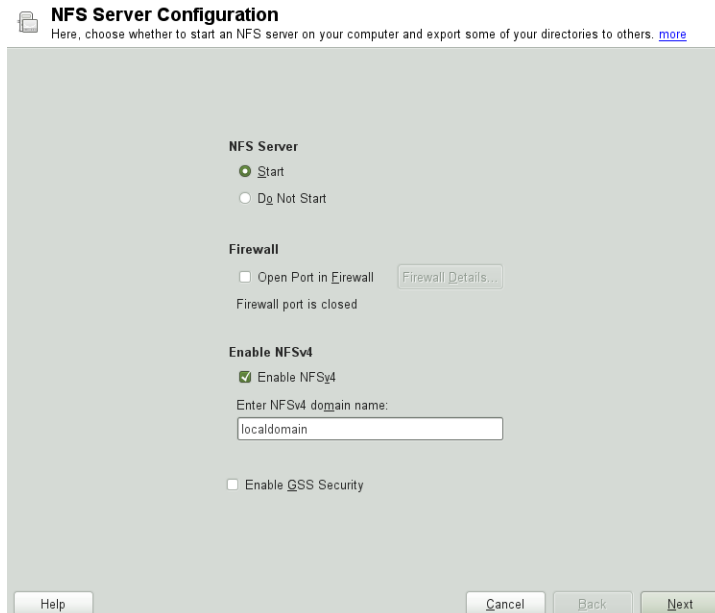
```
mount /local/path
```

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.

## 28.4 Exporting File Systems with YaST

With YaST, turn a host in your network into an NFS server—a server that exports directories and files to all hosts granted access to it. This could be done to provide applications to all members of a group without installing the applications locally on each and every host. To install such a server, start YaST and select *Network Services* > *NFS Server*; see Figure 28.2, “NFS Server Configuration Tool” (page 445).

**Figure 28.2** *NFS Server Configuration Tool*

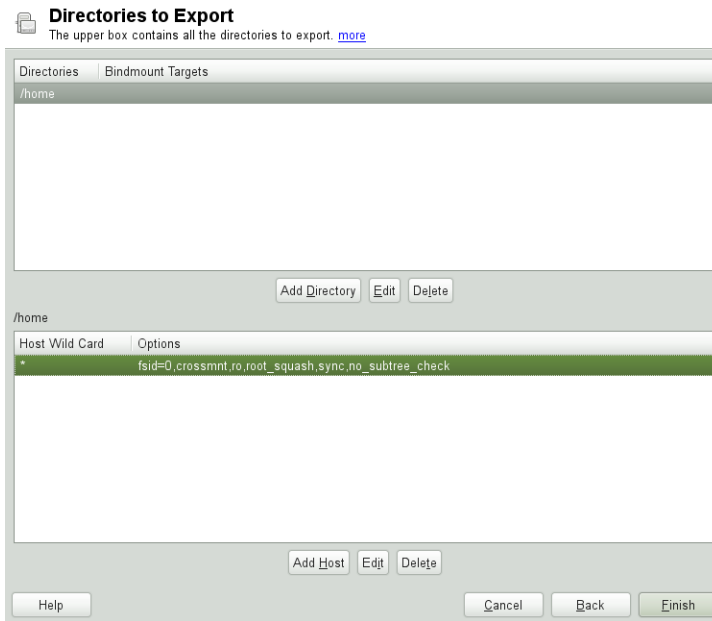


Then activate *Start* and enter the *NFSv4 Domain Name*.

Click *Enable GSS Security* if you need secure access to the server. A prerequisite for this is to have Kerberos installed on your domain and to have both the server and the clients kerberized. Click *Next*.

In the upper text field, enter the directories to export. Below, enter the hosts that should have access to them. This dialog is shown in Figure 28.3, “Configuring an NFS Server with YaST” (page 446). 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. Click *Finish* to complete the configuration.

**Figure 28.3** *Configuring an NFS Server with YaST*



---

### IMPORTANT: Automatic Firewall Configuration

If a firewall is active on your system (SuSEfirewall2), YaST adapts its configuration for the NFS server by enabling the `nfs` service when *Open Ports in Firewall* is selected.

---

## 28.4.1 Exporting for NFSv4 Clients

Activate *Enable NFSv4* to support NFSv4 clients. Clients with NFSv3 can still access the server's exported directories if they are exported appropriately. This is explained in detail in Section 28.4.3, “Coexisting v3 and v4 Exports” (page 449).

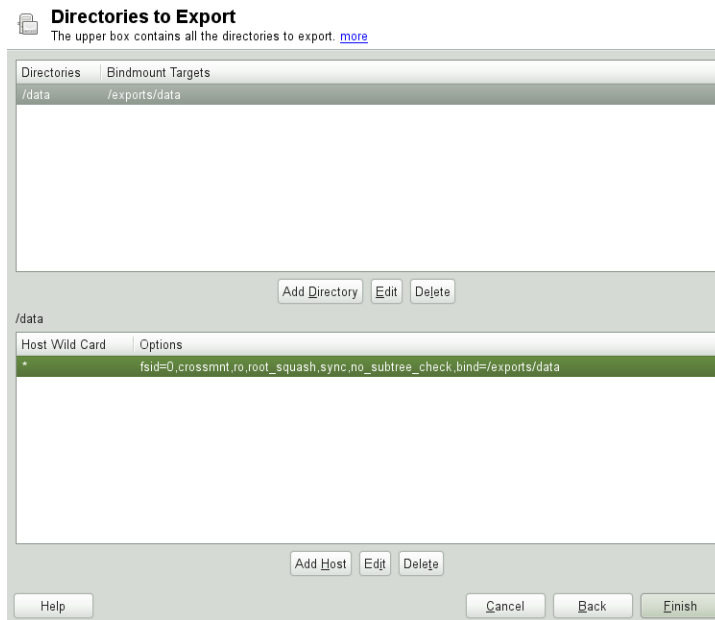
After activating NFSv4, enter an appropriate domain name. Make sure the name is the same as the one in the `/etc/ldapd.conf` file of any NFSv4 client that accesses this particular server. This parameter is for the `ldapd` service that is required for NFSv4 support (on both server and client). Leave it as `localdomain` (the default) if you do not have special requirements. For more information, see the links in Section 28.7, “For More Information” (page 453).



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 28.4** *Exporting Directories with NFSv4*



In the lower half of the dialog, enter the client (wild card) and export options for a particular directory. After adding a directory in the upper half, another dialog for entering the client and option information pops up automatically. After that, to add a new client (client set), click *Add Host*.

In the small dialog that opens, enter the host wild card. There are four possible types of host wild cards that can be set for each host: a single host (name or IP address),

netgroups, wild cards (such as `*` indicating all machines can access the server), and IP networks. Then, in *Options*, include `fsid=0` in the comma-separated list of options to configure the directory as pseudo root. If this directory 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 subdirectory of `/exports`. Any change in the option `bind=/target/path`, whether addition, deletion, or change in value, is reflected in *Bindmount Targets*. This column is not 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.

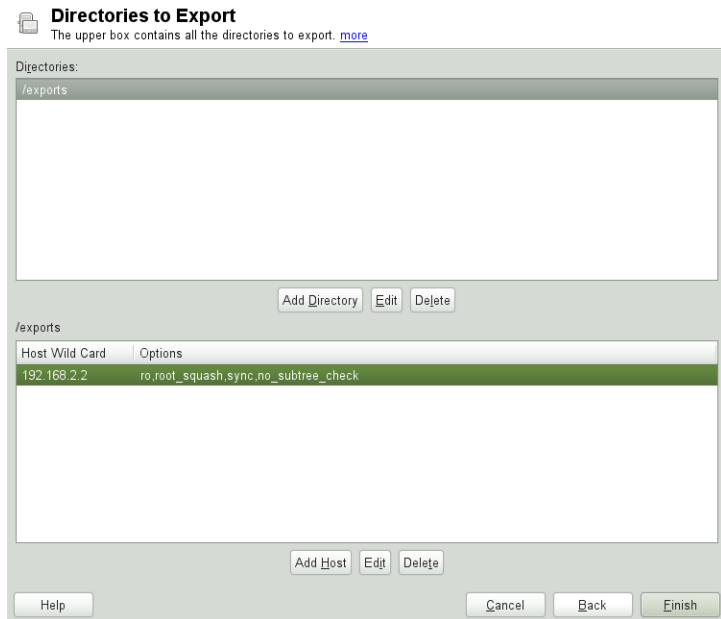
## 28.4.2 NFSv3 and NFSv2 Exports

Make sure that *Enable NFSv4* is not checked in the initial dialog before clicking *Next*.

The next dialog has two parts. In the upper text field, enter the directories to export. Below, enter the hosts that should have access to them. There are four types of host wild cards that can be set for each host: a single host (name or IP address), netgroups, wild cards (such as `*` indicating all machines can access the server), and IP networks.

This dialog is shown in Figure 28.5, “Exporting Directories with NFSv2 and v3” (page 449). Find a more thorough explanation of these options in `man exports`. Click *Finish* to complete the configuration.

**Figure 28.5** *Exporting Directories with NFSv2 and v3*



## 28.4.3 Coexisting v3 and v4 Exports

Both, NFSv3 and NFSv4 exports can coexist on a server. After enabling the support for NFSv4 in the initial configuration dialog, those exports for which `fsid=0` and `bind=/target/path` are not included in the option list are considered v3 exports. Consider the example in Figure 28.3, “Configuring an NFS Server with YaST” (page 446). 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.

---

# 28.5 Exporting File Systems Manually

The configuration files for the NFS export service are `/etc/exports` and `/etc/sysconfig/nfs`. In addition to these files, `/etc/idmapd.conf` is needed for the NFSv4 server configuration. To start or restart the services, run the 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 `rcr-pcbind restart`.

## 28.5.1 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 in that it is the root of the filesystem that is exported, sometime 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. Importantly, the parameter `NFS4_SUPPORT` must be set 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-sserver on`.

## 28.5.2 Exporting File Systems with NFSv2 and NFSv3

This section is specific to NFSv3 and NFSv2 exports. Refer to Section 28.4.1, “Exporting for NFSv4 Clients” (page 446) 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`.

## 28.6 NFS with Kerberos

To use Kerberos authentication for NFS, GSS security must be enabled. To do so, select *Enable GSS Security* in the initial YaST 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 28.7, “For More Information” (page 453).

## 28.7 For More Information

As well as the man pages of `exports`, `nfs`, and `mount`, information about configuring an NFS server and client is available in `/usr/share/doc/packages/nfsidmap/README`. Online documentation can be found at the following Web documents:

- Find the detailed technical documentation online at SourceForge [<http://nfs.sourceforge.net/>].
- For instructions for setting up kerberized NFS, refer to NFS Version 4 Open Source Reference Implementation [<http://www.citi.umich.edu/projects/nfsv4/linux/krb5-setup.html>].
- If you have 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.

## 29.1 Terminology

The following are some terms used in Samba documentation and in the YaST module.

### SMB protocol

Samba uses the SMB (server message block) protocol that is based on the Net-BIOS 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.

## 29.2 Installing a Samba Server

To install a Samba server, start YaST and select *Software > Software Management*. Choose *Filter > Patterns* and select *File Server*. Confirm the installation of the required packages to finish the installation process.

## 29.3 Starting and Stopping Samba

You can start or stop the Samba server automatically (during boot) or manually. Starting and stopping policy is a part of the YaST Samba server configuration described in Section 29.4.1, “Configuring a Samba Server with YaST” (page 457).

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.

## 29.4 Configuring a Samba Server

A Samba server in openSUSE® can be configured in two different ways: with YaST or manually. Manual configuration offers a higher level of detail, but lacks the convenience of the YaST GUI.

### 29.4.1 Configuring a Samba Server with YaST

To configure a Samba server, start YaST and select *Network Services > Samba Server*.

## 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 Server 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 CD (PDC) and click *Next*.

### Start-Up

Select whether you want to start Samba *During Boot* or *Manually* and click *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.

## 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 “Initial Samba Configuration” (page 458). 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 29.3, “Starting and Stopping Samba” (page 457).

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 *In-active*. 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).

## 29.4.2 Web Administration with SWAT

An alternative tool for Samba server administration is SWAT (Samba Web Administration Tool). It provides a simple Web interface with which to configure the Samba server. To use SWAT, open <http://localhost:901> in a Web browser and log in as user `root`. If you do not have a special Samba root account, use the system `root` account.

---

### NOTE: Activating SWAT

After Samba server installation, SWAT is not activated. To activate it, open *Network Services > Network Services (xinetd)* in YaST, enable the network services configuration, select *swat* from the table, and click *Toggle Status (On or Off)*.

---

## 29.4.3 Configuring the Server Manually

If you intend to use Samba as a server, install `samba`. The main configuration file of Samba is `/etc/samba/smb.conf`. This file can be divided into two logical parts. The `[global]` section contains the central and global settings. The `[share]` sections contain the individual file and printer shares. By means of this approach, details regarding the shares can be set differently or globally in the `[global]` section, which enhances the structural transparency of the configuration file.

### The global Section

The following parameters of the `[global]` section need some adjustment to match the requirements of your network setup so other machines can access your Samba server via SMB in a Windows environment.

```
workgroup = TUX-NET
```

This line assigns the Samba server to a workgroup. Replace `TUX-NET` with an appropriate workgroup of your networking environment. Your Samba server appears under its DNS name unless this name has been assigned to 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 29.7, “For More Information” (page 466).

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.

## Shares

The following examples illustrate how a CD-ROM drive and the user directories (homes) are made available to the SMB clients.

```
[cdrom]
```

To avoid having the CD-ROM drive accidentally made available, these lines are deactivated with comment marks (semicolons in this case). Remove the semicolons in the first column to share the CD-ROM drive with Samba.

### **Example 29.1** *A CD-ROM Share (deactivated)*

```
[cdrom]
; comment = Linux CD-ROM
; path = /media/cdrom
; locking = No
```

[cdrom] and comment

The [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 29.2** *[homes] Share*

```
[homes]
comment = Home Directories
valid users = %S
browseable = No
read only = No
create mask = 0640
directory mask = 0750
```

[homes]

As long as there is no other share using the share name of the user connecting to the SMB server, a share is dynamically generated using the [homes] share directives. The resulting name of the share is the username.

```
valid users = %S
```

%S is replaced with the concrete name of the share as soon as a connection has been successfully established. For a [homes] share, this is always the



username. As a consequence, access rights to a user's share are restricted exclusively to 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`.

## 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`.

## 29.5 Configuring Clients

Clients can only access the Samba server via TCP/IP. NetBEUI and NetBIOS via IPX cannot be used with Samba.

### 29.5.1 Configuring a Samba Client with YaST

Configure a Samba client to access resources (files or printers) on the Samba 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 automati-

cally 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.

## 29.6 Samba as Login Server

In networks where predominantly Windows clients are found, it is often preferable that users may only register with a valid account and password. In a Windows-based network, this task is handled by a primary domain controller (PDC). You can use a Windows NT server configured as PDC, but this task can also be done with a Samba server. The entries that must be made in the `[global]` section of `smb.conf` are shown in Example 29.3, “Global Section in `smb.conf`” (page 465).

### **Example 29.3** *Global Section in `smb.conf`*

```
[global]
 workgroup = TUX-NET
 domain logons = Yes
 domain master = Yes
```

If encrypted passwords are used for verification purposes 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`.

## 29.7 For More Information

Detailed Samba information is available in the digital documentation. Enter `apropos samba` at the command line to display some manual pages or just browse the `/usr/share/doc/packages/samba` directory if Samba documentation is installed for more online documentation and examples. Find a commented example configuration (`smb.conf.SUSE`) in the `examples` subdirectory.

The Samba 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.net-craft.com/>. Apache, developed by the Apache Software Foundation (<http://www.apache.org/>), is available for most operating systems. openSUSE® includes Apache version 2.2. In this chapter, learn how to install, configure and set up a Web server; how to use SSL, CGI, and additional modules; and how to troubleshoot Apache.

## 30.1 Quick Start

With the help of this section, quickly set up and start Apache. You must be `root` to install and configure Apache.

### 30.1.1 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 23, *Basic Networking* (page 341).
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 27, *Time Synchronization with NTP* (page 433) 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 14, *Masquerading and Firewalls*, ↑Security Guide) for details.

## 30.1.2 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 30.1*** *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 30.4, “Installing, Activating, and Configuring Modules” (page 486) for more information about modules.

## 30.1.3 Start

You can make Apache start automatically at boot time or you can start it manually.

### ***Procedure 30.2*** *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.

For more information about the runlevels in openSUSE and a description of the YaST runlevel editor, refer to Section 17.2.3, “Configuring System Services (Runlevel) with YaST” (page 233).

To manually start Apache using the shell, run `rcapache2 start`.

#### **Procedure 30.3** *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 30.8, “Troubleshooting” (page 505).

Now that the Web server is running, you can add your own documents, adjust the configuration according to your needs, or add functionality by installing modules.

## 30.2 Configuring Apache

openSUSE offers two configuration options:

- Configuring Apache Manually (page 473)
- Configuring Apache with YaST (page 477)

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 30.3, “Starting and Stopping Apache” (page 484).

If you configure Apache with YaST, this can be taken care of automatically if you set *HTTP Service* to *Enabled* as described in Section “HTTP Server Configuration” (page 482).

---

## 30.2.1 Apache Configuration Files

This section gives an overview of the Apache configuration files. If you decide to use YaST for configuration, you will 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 470)
- `/etc/apache2/` (page 470)

### **`/etc/sysconfig/apache2`**

`/etc/sysconfig/apache2` controls some global settings of Apache, like modules to load, additional configuration files to include, flags with which the server should be started, and flags that should be added to the command line. Every configuration option in this file is extensively documented and therefore not mentioned here. For a general-purpose Web server, the settings in `/etc/sysconfig/apache2` should be sufficient for any configuration needs.

### **`/etc/apache2/`**

`/etc/apache2/` hosts all configuration files for Apache. In the following, the purpose of each file is explained. Each file includes several configuration options (also referred to as *directives*). Every configuration option in these files is extensively documented and therefore not mentioned here.

The Apache configuration files are organized as follows:

```
/etc/apache2/
|
|- charset.conv
|- conf.d/
| |
| |- *.conf
```



```

|
|- default-server.conf
|- errors.conf
|- httpd.conf
|- listen.conf
|- magic
|- mime.types
|- mod_*.conf
|- server-tuning.conf
|- ssl.*
|- ssl-global.conf
|- sysconfig.d
| |
| |- global.conf
| |- include.conf
| |- loadmodule.conf . .
|
|- uid.conf
|- vhosts.d
| |- *.conf

```

## ***Apache Configuration Files in /etc/apache2/***

`charset.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 474).

#### `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 30.4, “Installing, Activating, and Configuring Modules” (page 486) for details. Note that configuration files for optional modules reside in the directory `conf.d`.

#### `server-tuning.conf`

Contains configuration directives for the different MPMs (see Section 30.4.4, “Multiprocessing Modules” (page 491)) as well as general configuration options that control Apache's performance. Properly test your Web server when making changes here.

#### `ssl-global.conf` and `ssl.*`

Global SSL configuration and SSL certificate data. Refer to Section 30.6, “Setting Up a Secure Web Server with SSL” (page 497) 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 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 “Virtual Host Configuration” (page 473) for details.

## 30.2.2 Configuring Apache Manually

Configuring Apache manually involves editing plain text configuration files as user `root`.

### 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 480) 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 configuration 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 30.1*** *Variations of Name-Based VirtualHost Entries*

```
NameVirtualHost IP-address[:Port]
NameVirtualHost 192.168.3.100:80
NameVirtualHost 192.168.3.100
NameVirtualHost *:80
NameVirtualHost *
NameVirtualHost [2002:c0a8:364::]:80
```

The opening `VirtualHost` tag takes the IP address (or fully qualified domain name) previously declared with the `NameVirtualHost` as an argument in a name-based virtual host configuration. A port number previously declared with the `NameVirtualHost` directive is optional.

The wild card `*` is also allowed as a substitute for the IP address. This syntax is only valid in combination with the wild card usage in `NameVirtualHost *`. When using IPv6 addresses, the address must be included in square brackets.

### ***Example 30.2 Name-Based VirtualHost Directives***

```
<VirtualHost 192.168.3.100:80>
...
</VirtualHost>

<VirtualHost 192.168.3.100>
...
</VirtualHost>

<VirtualHost *:80>
...
</VirtualHost>

<VirtualHost *>
...
</VirtualHost>

<VirtualHost [2002:c0a8:364::]>
...
</VirtualHost>
```

## **IP-Based Virtual Hosts**

This alternative virtual host configuration requires the setup of multiple IPs for a machine. One instance of Apache hosts several domains, each of which is assigned a different IP.

The physical server must have one IP address for each IP-based virtual host. If the machine does not have multiple network cards, virtual network interfaces (IP aliasing) can also be used.

The following example shows Apache running on a machine with the IP `192.168.3.100`, hosting two domains on the additional IPs `192.168.3.101` and `192.168.3.102`. A separate `VirtualHost` block is needed for every virtual server.

### **Example 30.3** *IP-Based VirtualHost Directives*

```
<VirtualHost 192.168.3.101>
...
</VirtualHost>

<VirtualHost 192.168.3.102>
...
</VirtualHost>
```

Here, `VirtualHost` directives are only specified for interfaces other than `192.168.3.100`. When a `Listen` directive is also configured for `192.168.3.100`, a separate IP-based virtual host must be created to answer HTTP requests to that interface—otherwise the directives found in the default server configuration (`/etc/apache2/default-server.conf`) are applied.

## **Basic Virtual Host Configuration**

At least the following directives should be present in each virtual host configuration in order to set up a virtual host. See `/etc/apache2/vhosts.d/vhost.template` for more options.

`ServerName`

The fully qualified domain name under which the host should be addressed.

`DocumentRoot`

Path to the directory from which Apache should serve files for this host. For security reasons, access to the entire file system is forbidden by default, so you must explicitly unlock this directory within a `Directory` container.

`ServerAdmin`

E-mail address of the server administrator. This address is, for example, shown on error pages Apache creates.

`ErrorLog`

The error log file for this virtual host. Although it is not necessary to create separate error log files for each virtual host, it is common practice to do so, because it makes 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, be-

cause 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/docs">
 Order allow,deny
 Allow from all
</Directory>
```

The complete configuration file looks like this:

### **Example 30.4** *Basic VirtualHost Configuration*

```
<VirtualHost 192.168.3.100>
 ServerName www.example.com
 DocumentRoot /srv/www/www.example.com/docs
 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/docs">
 Order allow,deny
 Allow from all
 </Directory>
</VirtualHost>
```

## **30.2.3 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 “HTTP Server Configuration” (page 482).

### **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 483). Click *Next* to advance to the next dialog.


## Default Host

This option pertains to the default Web server. As explained in Section “Virtual Host Configuration” (page 473), Apache can serve multiple virtual hosts from a single physical machine. The first declared virtual host in the configuration file is commonly referred to as the *default host*. Each virtual host inherits the default host's configuration.

To edit the host settings (also called *directives*), choose the appropriate entry in the table then click *Edit*. To add new directives, click *Add*. To delete a directive, select it and click *Delete*.



**Figure 30.1** HTTP Server Wizard: Default Host

 **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               | /srv/www/htdocs                         |
| Directory                   | /srv/www/htdocs/...                     |
| Alias                       | /icons/ /usr/share/apache2/icons/       |
| Directory                   | /usr/share/apache2/icons/...            |
| ScriptAlias                 | /cgi-bin/ /srv/www/cgi-bin/             |
| Directory                   | /srv/www/cgi-bin/...                    |
| mod_userdir.c               |                                         |
| Include                     | /etc/apache2/conf.d/*.conf              |
| Include                     | /etc/apache2/conf.d/apache2-manual.conf |
| Server Name                 | archimedes                              |
| Server Administrator E-Mail | root@archimedes                         |

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 “Virtual Host Configuration” (page 473)). 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 30.6.2, “Configuring Apache with SSL” (page 502) for details on SSL and certificates. With the *Directory Index* option, you can specify which file to display when the client requests a directory (by default, `index.html` ). Add one or more filenames (space-separated) if you want to change this. With *Enable Public HTML*, the content of the users public directories (`~ user/public_html/` ) is made available on the server under `http://www.example.com/~user`.

---

### **IMPORTANT: Creating Virtual Hosts**

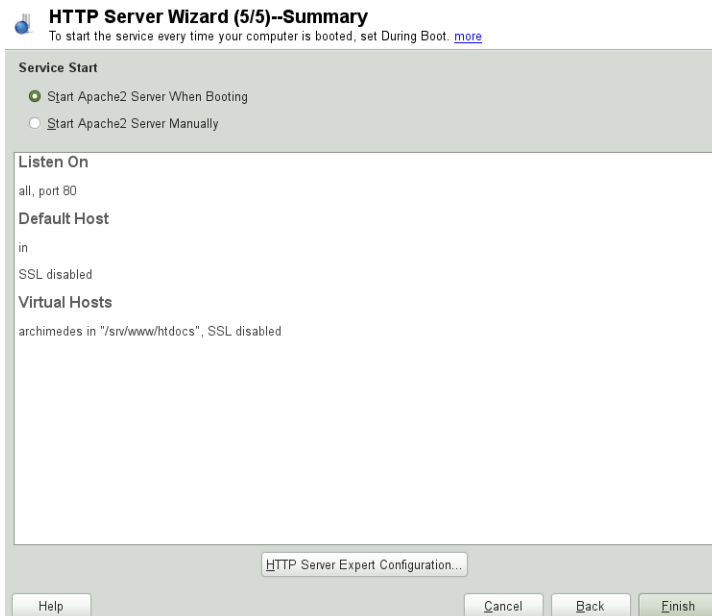
It is not possible to add virtual hosts at will. If using name-based virtual hosts, each hostname must be resolved on the network. If using IP-based virtual hosts, you can assign only one host to each IP address available.

---

## **Summary**

This is the final step of the wizard. Here, determine how and when the Apache server is started: when booting or manually. Also see a short summary of the configuration made so far. If you are satisfied with your settings, click *Finish* to complete configuration. If you want to change something, click *Back* until you have reached the desired dialog. Clicking *HTTP Server Expert Configuration* opens the dialog described in Section “HTTP Server Configuration” (page 482).

**Figure 30.2** *HTTP Server Wizard: Summary*



## HTTP Server Configuration

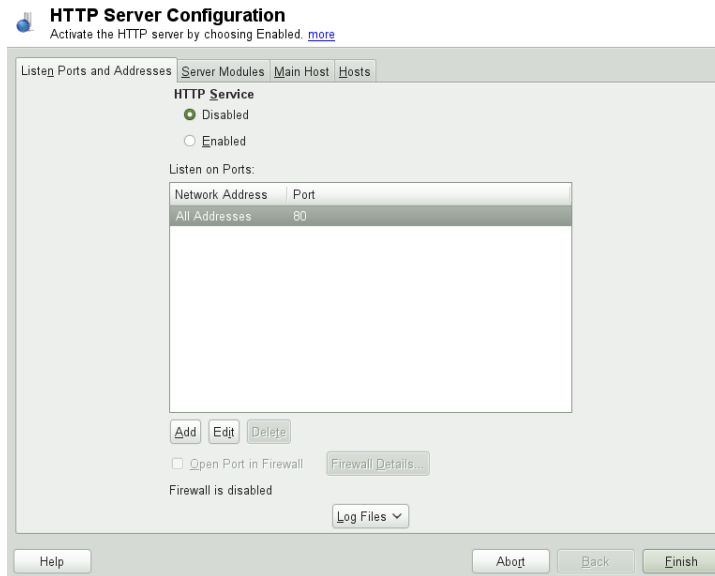
The *HTTP Server Configuration* dialog also lets you make even more adjustments to the configuration than the wizard (which only runs if you configure your Web server for the first time). It consists of four tabs described in the following. No configuration option you change here is effective immediately—you always must confirm your changes with *Finish* to make them effective. Clicking *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 30.3, “Starting and Stopping Apache” (page 484). These commands are effective immediately and their log messages are also displayed immediately.

**Figure 30.3** *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 30.4, “Installing, Activating, and Configuring Modules” (page 486).

**Figure 30.4** HTTP Server Configuration: Server Modules



## Main Host or Hosts

These dialogs are identical to the ones already described. Refer to Section “Default Host” (page 478) and Section “Virtual Hosts” (page 480).

## 30.3 Starting and Stopping Apache

If configured with YaST as described in Section 30.2.3, “Configuring Apache with YaST” (page 477), 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`. For general information about init scripts, refer to Section 17.2.2, “Init Scripts” (page 229). 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 30.6, “Setting Up a Secure Web Server with SSL” (page 497).

`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.

---

## 30.4 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 30.4.2, “Activation and Deactivation” (page 488) 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.

## 30.4.1 Module Installation

If you have done a default installation as described in Section 30.1.2, “Installation” (page 468), 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.

## 30.4.2 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 “HTTP Server Wizard” (page 477). All other modules can be enabled or disabled as described in Section “Server Modules” (page 483).

If you prefer to activate or deactivate the modules manually, use the commands `a2enmod mod_foo` or `a2dismod mod_foo`, respectively. `a2enmod -l` outputs a list of all currently active modules.

---

### IMPORTANT: Including Configuration Files for External Modules

If you have activated external modules manually, make sure to load their configuration files in all virtual host configurations. Configuration files for external modules are located under `/etc/apache2/conf.d/` and are not loaded by default. If you need the same modules on each virtual host, you can include `*.conf` from this directory. Otherwise include individual files. See `/etc/apache2/vhost.d/vhost.template` for examples.

---

## 30.4.3 Base and Extension Modules

All base and extension modules are described in detail in the Apache documentation. Only a brief description of the most important modules is available here. Refer to <http://httpd.apache.org/docs/2.2/mod/> to learn details about each module.

### `mod_actions`

Provides methods to execute a script whenever a certain MIME type (such as `application/pdf`), a file with a specific extension (like `.rpm`), or a certain request method (such as `GET`) is requested. This module is enabled by default.

### `mod_alias`

Provides `Alias` and `Redirect` directives with which you can map a URL to a specific directory (`Alias`) or redirect a requested URL to another location. This module is enabled by default.

## mod\_auth\*

The authentication modules provide different authentication methods: basic authentication with `mod_auth_basic` or digest authentication with `mod_auth_digest`. Digest authentication in Apache 2.2 is considered experimental.

`mod_auth_basic` and `mod_auth_digest` must be combined with an authentication provider module, `mod_authn_*` (for example, `mod_authn_file` for text file-based authentication) and with an authorization module `mod_authz_*` (for example, `mod_authz_user` for user authorization).

More information about this topic is available in the *Authentication HOWTO* at <http://httpd.apache.org/docs/2.2/howto/auth.html>.

## mod\_autoindex

Autoindex generates directory listings when no index file (for example, `index.html`) is present. The look and feel of these indexes is configurable. This module is enabled by default. However, directory listings are disabled by default via the `Options` directive—overwrite this setting in your virtual host configuration. The default configuration file for this module is located at `/etc/apache2/mod_autoindex-defaults.conf`.

## mod\_cgi

`mod_cgi` is needed to execute CGI scripts. This module is enabled by default.

## mod\_deflate

Using this module, Apache can be configured to compress given file types on the fly before delivering them.

## mod\_dir

`mod_dir` provides the `DirectoryIndex` directive with which you can configure which files are automatically delivered when a directory is requested (`index.html` by default). It also provides an automatic redirect to the correct URL when a directory request does not contain a trailing slash. This module is enabled by default.

## mod\_env

Controls the environment that is passed to CGI scripts or SSI pages. Environment variables can be set or unset or passed from the shell that invoked the `httpd` process. This module is enabled by default.

### `mod_expires`

With `mod_expires`, you can control how often proxy and browser caches refresh your documents by sending an `Expires` header. This module is enabled by default.

### `mod_include`

`mod_include` lets you use Server Side Includes (SSI), which provide a basic functionality to generate HTML pages dynamically. This module is enabled by default.

### `mod_info`

Provides a comprehensive overview of the server configuration under `http://localhost/server-info/`. For security reasons, you should always limit access to this URL. By default only `localhost` is allowed to access this URL. `mod_info` is configured at `/etc/apache2/mod_info.conf` .

### `mod_log_config`

With this module, you can configure the 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 30.6, “Setting Up a Secure Web Server with SSL” (page 497) for details. This module is enabled by default.

#### `mod_status`

Provides information on server activity and performance under `http://localhost/server-status/`. For security reasons, you should always limit access to this URL. By default, only `localhost` is allowed to access this URL. `mod_status` is configured at `/etc/apache2/mod_status.conf`

#### `mod_suexec`

`mod_suexec` lets you run CGI scripts under a different user and group. This module is enabled by default.

#### `mod_userdir`

Enables user-specific directories available under `~ user/`. The `UserDir` directive must be specified in the configuration. This module is enabled by default.

## 30.4.4 Multiprocessing Modules

openSUSE provides two different multiprocessing modules (MPMs) for use with Apache:

- Prefork MPM (page 491)
- Section “Worker MPM” (page 492)

### 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.

---

## 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.

---

## 30.4.5 External Modules

Find a list of all external modules shipped with openSUSE [here](#). Find the module's documentation in the listed directory.

`mod-apparmor`

Adds support to Apache to provide Novell AppArmor confinement to individual CGI scripts handled by modules like `mod_php5` and `mod_perl`.

Package Name: `apache2-mod_apparmor`

More Information: Part “Confining Privileges with Novell 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`

## 30.4.6 Compilation

Apache can be extended by advanced users by writing custom modules. To develop modules for Apache or compile third-party modules, the package `apache2-devel` is required along with the corresponding development tools. `apache2-devel` also contains the `apxs2` tools, which are necessary for compiling additional modules for Apache.

`apxs2` enables the compilation and installation of modules from source code (including the required changes to the configuration files), which creates *dynamic shared objects* (DSOs) that can be loaded into Apache at runtime.

The `apxs2` binaries are located under `/usr/sbin` :

- `/usr/sbin/apxs2` —suitable for building an extension module that works with any MPM. The installation location is `/usr/lib/apache2` .
- `/usr/sbin/apxs2-prefork` —suitable for prefork MPM modules. The installation location is `/usr/lib/apache2-prefork` .
- `/usr/sbin/apxs2-worker` —suitable for worker MPM modules. 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.

## 30.5 Getting CGI Scripts to Work

Apache's Common Gateway Interface (CGI) lets you create dynamic content with programs or scripts usually referred to as CGI scripts. CGI scripts can be written in any programming language. Usually, script languages such as Perl or PHP are used.

To enable Apache to deliver content created by CGI scripts, `mod_cgi` needs to be activated. `mod_alias` is also needed. Both modules are enabled by default. Refer



to Section 30.4.2, “Activation and Deactivation” (page 488) for details on activating modules.

---

### **WARNING: CGI Security**

Allowing the server to execute CGI scripts is a potential security hole. Refer to Section 30.7, “Avoiding Security Problems” (page 503) for additional information.

---

## **30.5.1 Apache Configuration**

In openSUSE, the execution of CGI scripts is only allowed in the directory `/srv/www/cgi-bin/`. This location is already configured to execute CGI scripts. If you have created a virtual host configuration (see Section “Virtual Host Configuration” (page 473)) and want to place your scripts in a host-specific directory, you must unlock and configure this directory.

### ***Example 30.5 VirtualHost CGI Configuration***

```
ScriptAlias /cgi-bin/ "/srv/www/www.example.com/cgi-bin/"❶

<Directory "/srv/www/www.example.com/cgi-bin/">
 Options +ExecCGI❷
 AddHandler cgi-script .cgi .pl❸
 Order allow,deny❹
 Allow from all
</Directory>
```

- ❶ Tells Apache to handle all files within this directory as CGI scripts.
- ❷ Enables CGI script execution
- ❸ Tells the server to treat files with the extensions `.pl` and `.cgi` as CGI scripts. Adjust according to your needs.
- ❹ The `Order` and `Allow` directives control the default access state and the order in which `allow` and `deny` directives are evaluated. In this case “allow” statements are evaluated before “deny” statements and universal access is enabled.

## **30.5.2 Running an Example Script**

CGI programming differs from “regular” programming in that the CGI programs and scripts must be preceded by a MIME-Type header such as `Content-type:`

text/html. This header is sent to the client, so it understands what kind of content it receives. Secondly, the script's output must be something the client, usually a Web browser, understands—HTML in most cases or plain text or images, for example.

A simple test script available under `/usr/share/doc/packages/apache2/test-cgi` is part of the Apache package. It outputs the content of some environment variables as plain text. Copy this script to either `/srv/www/cgi-bin/` or the script directory of your virtual host (`/srv/www/www.example.com/cgi-bin/`) and name it `test.cgi`.

Files accessible by the Web server should be owned by the user `root`. For additional information see Section 30.7, “Avoiding Security Problems” (page 503). Because the Web server runs with a different user, the CGI scripts must be world-executable and world-readable. Change into the CGI directory and use the command `chmod 755 test.cgi` to apply the proper permissions.

Now call `http://localhost/cgi-bin/test.cgi` or `http://www.example.com/cgi-bin/test.cgi`. You should see the “CGI/1.0 test script report”.

## 30.5.3 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.

## 30.6 Setting Up a Secure Web Server with SSL

Whenever sensitive data, such as credit card information, is transferred between Web server and client, it 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/TLS, 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/TLS 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`.

---

### 30.6.1 Creating an SSL Certificate

In order to use SSL/TLS 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](http://httpd.apache.org/docs/2.2/ssl/ssl_intro.html) .

---

## Creating a “Dummy” Certificate

Generating a dummy certificate is simple. Just call the script `/usr/bin/gensslcert`. It creates or overwrites the 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.

---

# Creating a Self-Signed Certificate

If you are setting up a secure Web server for an Intranet or for a defined circle of users, it might be sufficient if you sign a certificate with your own certificate authority (CA).

Creating a self-signed certificate is an interactive nine-step process. Change into the directory `/usr/share/doc/packages/apache2` and run the following command: `./mkcert.sh make --no-print-directory /usr/bin/openssl /usr/sbin/ custom`. Do not attempt to run this command from outside this directory. The program provides a series of prompts, some of which require user input.

## ***Procedure 30.4 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` . Other-

wise 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.

---

## **Getting an Officially Signed Certificate**

There are a number of official certificate authorities that sign your certificates. The certificate is signed by a trustworthy third party, so can be fully trusted. Publicly operating secure Web servers usually have got an officially signed certificate.

The best-known official CAs are Thawte (<http://www.thawte.com/>) or Verisign (<http://www.verisign.com>). These and other CAs are already compiled into all browsers, so certificates signed by these certificate authorities are automatically accepted by the browser.

When requesting an officially signed certificate, you do not send a certificate to the CA. Instead, issue a Certificate Signing Request (CSR). To create a CSR, call the script `/usr/share/ssl/misc/CA.sh -newreq`.

First the script asks for a password with which the CSR should be encrypted. Then you are asked to enter a distinguished name. This requires you to answer a few questions, such as country name or organization name. Enter valid data—everything you enter here later shows up in the certificate and is checked. You do not need to answer every question. If one does not apply to you or you want to leave it blank, use “.”.

Common name is the name of the CA itself—choose a significant name, such as *My company* CA. Last, a challenge password and an alternative company name must be entered.

Find the CSR in the directory from which you called the script. The file is named `newreq.pem` .

## 30.6.2 Configuring Apache with SSL

The default port for SSL and TLS requests on the Web server side is 443. There is no conflict between a “regular” Apache listening on port 80 and an SSL/TLS-enabled Apache listening on port 443. In fact, HTTP and HTTPS can be run with the same Apache instance. Usually separate virtual hosts are used to dispatch requests to port 80 and port 443 to separate virtual servers.

---

### IMPORTANT: Firewall Configuration

Do not forget to open the firewall for SSL-enabled Apache on port 443. This can be done with YaST as described in Section “Configuring the Firewall with YaST” (Chapter 14, *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 “Virtual Host Configuration” (page 473) for the general virtual host configuration.

To get started, copy the template to `/etc/apache2/vhosts.d/mySSL-host.conf` and edit it. Adjusting the values for the following directives should be sufficient:



- DocumentRoot
- ServerName
- ServerAdmin
- ErrorLog
- TransferLog

---

**IMPORTANT: Name-Based Virtual Hosts and SSL**

It is not possible to run multiple SSL-enabled virtual hosts on a server with only one IP address. Users connecting to such a setup receive a warning message stating that the certificate does not match the server name every time they visit the URL. A separate IP address or port is necessary for every SSL-enabled domain to achieve communication based on a valid SSL certificate.

---

## 30.7 Avoiding Security Problems

A Web server exposed to the public Internet requires an ongoing administrative effort. It is inevitable that security issues appear, both related to the software and to accidental misconfiguration. Here are some tips for how to deal with them.

### 30.7.1 Up-to-Date Software

If there are vulnerabilities found in the Apache software, a security advisory will be issued by SUSE. It contains instructions for fixing the vulnerabilities, which in turn should be applied 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](http://www.novell.com/linux/security/suse_security.xml)

## 30.7.2 DocumentRoot Permissions

By default in openSUSE, the `DocumentRoot` directory `/srv/www/htdocs` and the CGI directory `/srv/www/cgi-bin` belong to the user and group `root`. You should not change these permissions. If the directories 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`.

## 30.7.3 File System Access

By default, access to the whole file system is denied in `/etc/apache2/httpd.conf`. You should never overwrite these directives, but specifically enable access to all directories Apache should be able to read. For details, see Section “Basic Virtual Host Configuration” (page 476). In doing so, ensure that no critical files, such as password or system configuration files, can be read from the outside.

## 30.7.4 CGI Scripts

Interactive scripts in Perl, PHP, SSI, or any other programming language can essentially run arbitrary commands and therefore present a general security issue. Scripts that will be executed from the server should only be installed from sources the server administrator trusts—allowing users to run their own scripts is generally not a good idea. It is also recommended to do security audits for all scripts.

To make the administration of scripts as easy as possible, it is common practice to limit the execution of CGI scripts to specific directories instead of globally allowing them. The directives `ScriptAlias` and `Option ExecCGI` are used for configuration. The openSUSE default configuration does not allow execution of CGI scripts from everywhere.

All CGI scripts run as the same user, so different scripts can potentially conflict with each other. The module `suEXEC` lets you run CGI scripts under a different user and group.

## 30.7.5 User Directories

When enabling user directories (with `mod_userdir` or `mod_rewrite`) you should strongly consider not allowing `.htaccess` files, which would allow users to overwrite security settings. At least you should limit the user's engagement by using the directive `AllowOverride`. In openSUSE, `.htaccess` files are enabled by default, but the user is not allowed to overwrite any `Option` directives when using `mod_userdir` (see the `/etc/apache2/mod_userdir.conf` configuration file).

## 30.8 Troubleshooting

If Apache does not start, the Web page is not accessible, or users cannot connect to the Web server, it is important to find the cause of the problem. Here are some typical places to look for error explanations and important things to check:

### 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 30.3, “Starting and Stopping Apache” (page 484)). It is verbose about errors, and it even provides tips and hints for solving 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 that is 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 30.2.3, “Configur-

ing Apache with YaST” (page 477)). 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](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](http://comp.infosystems.www.servers.unix).

## 30.9 For More Information

The package `apache2-doc` contains the complete Apache manual in various localizations for local installation and reference. It is not installed by default—the quickest way to install it is to use the command `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.*`.

### 30.9.1 Apache 2.2

For a list of new features in Apache 2.2, refer to [http://httpd.apache.org/docs/2.2/new\\_features\\_2\\_2.html](http://httpd.apache.org/docs/2.2/new_features_2_2.html). Information about upgrading from version 2.0 to 2.2 is available at <http://httpd.apache.org/docs-2.2/upgrading.html>.

### 30.9.2 Apache Modules

More information about external Apache modules that are briefly described in Section 30.4.5, “External Modules” (page 492) is available at the following locations:

`mod-apparmor`

<http://en.opensuse.org/SDB:AppArmor>

`mod_mono`

[http://www.mono-project.com/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](http://www.php.net/manual/en/install.unix.apache2.php)

mod\_python

<http://www.modpython.org/>

mod\_tidy

<http://mod-tidy.sourceforge.net/>

## 30.9.3 Development

More information about developing Apache modules or about getting involved in the Apache Web server project are available at the following locations:

Apache Developer Information

<http://httpd.apache.org/dev/>

Apache Developer Documentation

<http://httpd.apache.org/docs/2.2/developer/>

Writing Apache Modules with Perl and C

<http://www.modperl.com/>

## 30.9.4 Miscellaneous Sources

If you experience difficulties specific to Apache in openSUSE, take a look at the openSUSE wiki at <http://old-en.opensuse.org/Apache> . The history of Apache is provided at [http://httpd.apache.org/ABOUT\\_APACHE.html](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.
- 3 In the *Start-Up* dialog, configure the options for starting of the FTP server. For more information, see Section 31.1, “Starting the FTP server” (page 510).

In the *General* dialog, configure FTP directories, welcome message, file creation masks and various other parameters. For more information, see Section 31.2, “FTP General Settings” (page 511).

In the *Performance* dialog, set the parameters that affect the load on the FTP server. For more information, see Section 31.3, “FTP Performance Settings” (page 512).

In the *Authentication* dialog, set whether the FTP server should be available for anonymous and/or authenticated users. For more information, see Section 31.4, “Authentication” (page 512).

In the *Expert Settings* dialog, configure the operation mode of the FTP server, SSL connections and firewall settings. For more information, see Section 31.5, “Expert Settings” (page 513).

- 4 Press *Finish* to save the configurations.

## 31.1 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*.

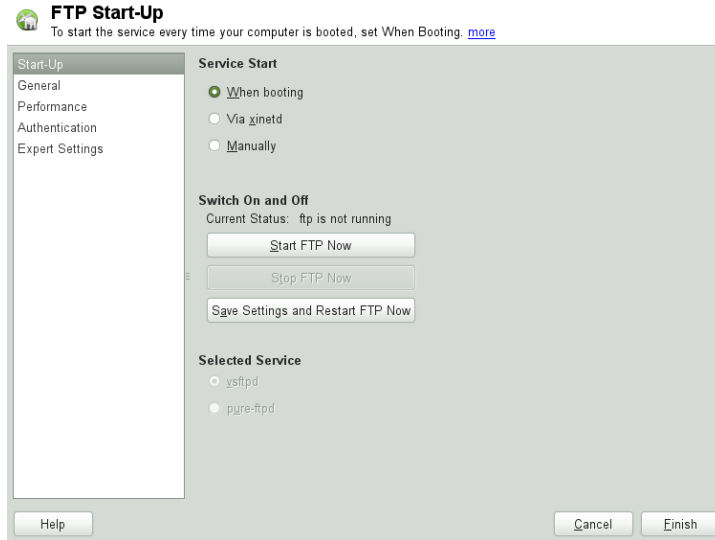
The current status of the FTP server is shown in the *Switch On and Off* frame of the *FTP Start-Up* dialog. Start the FTP server by clicking *Start FTP Now*. To stop the server, click *Stop FTP Now*. After having changed the settings of the server click *Save Settings and Restart FTP Now*. Your configurations 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 be-



tween 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 31.1** *FTP Server Configuration — Start-Up*



## 31.2 FTP General Settings

In the *General Settings* frame of the *FTP General Settings* dialog you can set the *Welcome message* which is shown after connecting to the FTP server.

If you check the *Chroot Everyone* option, all local users will be placed in a chroot jail in their home directory after login. This option has security implications, especially if the users have upload permission or shell access, so be careful enabling this option.

If you check the *Verbose Logging* option, all FTP requests and responses are logged.

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.

---

## 31.3 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.

## 31.4 Authentication

In the *Enable/Disable Anonymous and Local Users* frame of the *Authentication* dialog, you are able to set which users are allowed to access your FTP server. You can 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.

---

## 31.5 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.

## 31.6 For more information

For more information about FTP servers read the manual pages of vsftpd and vsftpd.conf.



## **Part VI. Mobility**



# Mobile Computing with Linux

# 32

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.

## 32.1 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 36, *Using Tablet PCs* (page 575).

---

## 32.1.1 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.

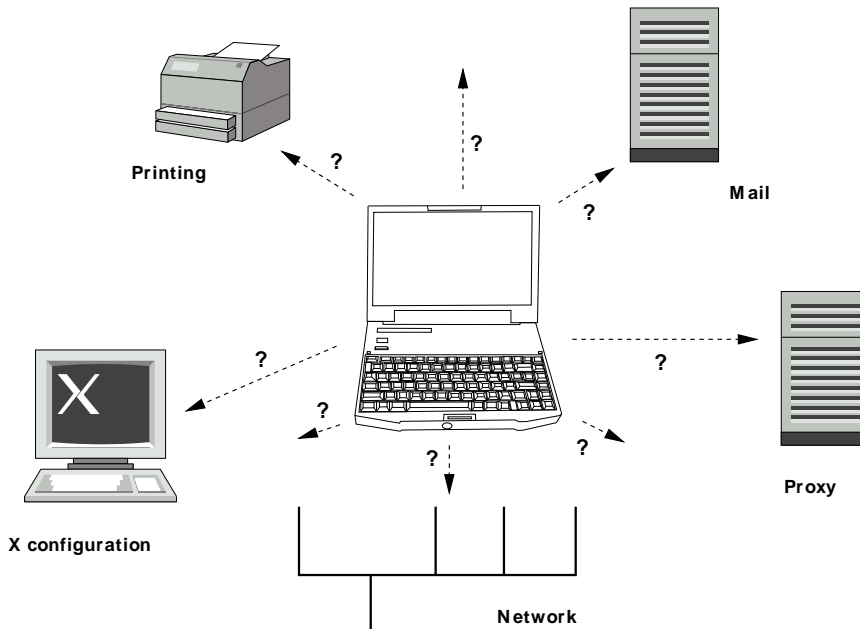
Detailed background information about power management in openSUSE is provided in Chapter 33, *Power Management* (page 527).

## 32.1.2 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 32.1** *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, the 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 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 35.4, “Using KNetworkManager” (page 562) and Section 35.5, “Using GNOME NetworkManager Applet” (page 564).

**Table 32.1** *Use Cases for NetworkManager*

| My computer... | Use NetworkManager |
|----------------|--------------------|
| is a laptop    | Yes                |

Use the YaST tools to configure networking whenever NetworkManager should not handle network configuration.

## SLP

The service location protocol (SLP) simplifies the connection of a laptop to an existing network. Without SLP, the administrator of a laptop usually requires detailed knowledge of the services available in a network. SLP broadcasts the availability of a certain type of service to all clients in a local network. Applications that support SLP can process the information dispatched by SLP and be configured automatically. SLP can even be used for the installation of a system, sparing the effort of searching for a suitable installation source. Find detailed information about SLP in Chapter 24, *SLP Services in the Network* (page 395).

# 32.1.3 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.

## System Monitoring

Two KDE system monitoring tools are provided by openSUSE:

## KPowersave

KPowersave is an applet that displays the state of the rechargeable battery in the control panel. The icon adjusts to represent the type of power supply. When working on AC power, a small plug icon is displayed. When working on batteries, the icon changes to a battery.

Right-click the KPowersave tray icon to access options to configure KPowersave behavior. You can choose one of four listed schemes, according to 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. You can also tell the system what to do if, for example, you close the laptop lid or press the power button.

## KSysguard

KSysguard is an independent application that gathers all measurable parameters of the system into one monitoring environment. KSysguard has monitors for ACPI (battery status), CPU load, network, partitioning and memory usage. It can also watch and display all system processes. The presentation and filtering of the collected data can be customized. It is possible to monitor different system parameters in various data pages or collect the data of various machines in parallel over the network. KSysguard can also run as a daemon on machines without a KDE environment. Find more information about this program in its integrated help function or in the SUSE help pages.

In the GNOME desktop, use GNOME Power Management and System Monitor applications.

# 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 sta-

tus information after the synchronization process has completed. Use an SMTP server implemented in the mail client for sending messages instead of the systemwide MTA postfix or sendmail to receive reliable feedback about unsent mail.

### Synchronizing Files and Directories

There are several utilities suitable for synchronizing data between a laptop and a workstation. For detailed information, refer to Chapter 37, *Copying and Sharing Files* (page 585).

## Wireless Communication

As well as connecting to a home or office network with a cable, a laptop can also wirelessly connect to other computers, peripherals, cellular phones or PDAs. Linux supports three types of wireless communication:

### WLAN

With the largest range of these wireless technologies, WLAN is the only one suitable for the operation of large and sometimes even spatially disjointed networks. Single machines can connect with each other to form an independent wireless network or access the Internet. Devices called *access points* act as base stations for WLAN-enabled devices and 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 34, *Wireless LAN* (page 539).

### 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 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.

## 32.1.4 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 11, *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 34, *Wireless LAN* (page 539).

## 32.2 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.

## 32.3 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 “Wireless Communication” (page 522). 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, in both cases, easily performed with the assistance of a wizard. Once the support for

Palm Pilots is configured, it is necessary to determine which type of data should be synchronized (addresses, appointments, etc.).

A more sophisticated synchronization solution is available with the program `open-sync` (see packages `libopensync`, `msyncntool` and the respective plug-ins for the different devices).

## 32.4 For More Information

The central point of reference for all questions regarding mobile devices and Linux is <http://tuxmobil.org/>. Various sections of that Web site deal with the hardware and software aspects of laptops, PDAs, cellular phones and other mobile hardware.

A similar approach to that of <http://tuxmobil.org/> is made by <http://www.linux-on-laptops.com/>. Information about laptops and handhelds can be found here.

SUSE maintains a mailing list in German dedicated to the subject of laptops. See <http://lists.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.

## 33.1 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.

## 33.2 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 33.2.3, “Troubleshooting” (page 532) for more information about troubleshooting ACPI problems.

## 33.2.1 Controlling the CPU Performance

The CPU can save energy in three ways:

- Frequency and Voltage Scaling (page 529)
- Throttling the Clock Frequency (T-states) (page 530)
- Putting the Processor to Sleep (C-states) (page 531)

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.

### Frequency and Voltage Scaling

PowerNow! and Speedstep are the designations AMD and Intel use for this technology. However, this technology is also applied in processors of other manufacturers. The clock frequency of the CPU and its core voltage are reduced at the same time, resulting in more than linear energy savings. This means that when the frequency is halved (half performance), far less than half of the energy is consumed. This technology is independent from ACPI.

There are two main approaches to performing CPU frequency scaling—by the kernel itself (CPUfreq infrastructure with in-kernel governors) or by a userspace applica-

tion. The in-kernel governors are policy governors that can change the CPU frequency based on different criteria (a sort of pre-configured power schemes for the CPU). The following governors are available with the CPUfreq subsystem:

#### Performance Governor

The CPU frequency is statically set to the highest possible for maximum performance. Consequently, saving power is not the focus of this governor.

#### Powersave Governor

The CPU frequency is statically set to the lowest possible. This can have severe impact on the performance, as the system will never rise above this frequency no matter how busy the processors are.

#### On-demand Governor

The kernel implementation of a dynamic CPU frequency policy: The governor monitors the processor utilization. As soon as it exceeds a certain threshold, the governor will set the frequency to the highest available. If the utilization is less than the threshold, the next lowest frequency is used. If the system continues to be underutilized, the frequency is again reduced until the lowest available frequency is set.

#### Conservative Governor

Similar to the on-demand implementation, this governor also dynamically adjusts frequencies based on processor utilization, except that it allows for a more gradual increase in power. If processor utilization exceeds a certain threshold, the governor does not immediately switch to the highest available frequency (as the on-demand governor does), but only to next higher frequency available.

The relevant files for the kernel governors are located at `/sys/devices/system/cpu/cpu*/cpufreq/`. If your machine has more than one CPU, `/sys/devices/system/cpu/` will hold a subdirectory for each processor: `cpu0`, `cpu1`, etc. If your system currently uses the on-demand or conservative governor, you will see a separate subdirectory for those governors in `cpufreq`, containing the parameters for the governors.

## Throttling the Clock Frequency (T-states)

This technology omits a certain percentage of the clock signal impulses for the CPU. At 25% throttling, every fourth impulse is omitted. At 87.5%, only every eighth impulse reaches the processor. However, the energy savings are a little less than linear.

Normally, throttling is only used if frequency scaling is not available or to maximize power savings. This technology must be controlled by a special process, as well. The system interface for Processor Throttling States (T-states) is `/proc/acpi/processor/*/throttling` .

## Putting the Processor to Sleep (C-states)

Modern processors have several power saving modes called C-states. They reflect the capability of an idle processor to turn off unused components in order to save power. The operating system puts the processor to sleep whenever there is no activity. In this case, the operating system sends the CPU a `halt` command. There are three idle states: C1, C2, and C3. In the most economic state, C3, even the synchronization of the processor cache with the main memory is halted. Therefore, this state can only be applied if no other device modifies the contents of the main memory via bus master activity. Some drivers prevent the use of C3. The current state is displayed in `/proc/acpi/processor/*/power` .

## 33.2.2 Tools

To view or adjust the current settings of the CPUfreq subsystem use the tools provided by `cpufrequtils` for that. After you have installed the `cpufrequtils` package, use the `cpufreq-info` to retrieve CPUfreq kernel information. The `cpufreq-set` command can be used to modify CPUfreq settings. For example, run the following command as `root` to activate the on-demand governor at runtime:

```
cpufreq-set -g ondemand
```

For more details and the available options, refer to the `cpufreq-info` and the `cpufreq-set` man pages or run `cpufreq-info --help` or `cpufreq-set --help`, respectively.

A useful tool for monitoring system power consumption is `powerTOP`, available after installation of the `powertop` package. It helps you to identify the reasons for unnecessary high power consumption (for example, processes that are mainly responsible for waking up a processor from its idle state) and to optimize your system settings to avoid these. It supports both Intel and AMD processors. For detailed information, refer to the `powerTOP` project page at <http://www.lesswatts.org/projects/powertop/> .

Apart from the tools above, the following ACPI utilities is available:

- To merely display information, like the battery charge level and the temperature, you can use the `acpi` command. For a list of available options, run `acpi --help`.
- For editing the ACPI tables in the BIOS, install the `acpica` package.

## 33.2.3 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 boot-

ing. 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 33.4.1, “ACPI Activated with Hardware Support but Functions Do Not Work” (page 535).

In the kernel configuration, there is a switch for activating ACPI debug messages. If a kernel with ACPI debugging is compiled and installed, experts searching for an error can be supported with detailed information.

If you experience BIOS or hardware problems, it is always advisable to contact the manufacturers. Especially if they do not always provide assistance for Linux, they should be confronted with the problems. Manufacturers will only take the issue seriously if they realize that an adequate number of their customers use Linux.

## For More Information

- <http://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)

## 33.3 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.

## 33.4 Troubleshooting

All error messages and alerts are logged in the file `/var/log/messages`. The following sections cover the most common problems.

### 33.4.1 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 33.1*** *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.asl` to any location (`/etc/DSDT.asl` 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.

## 33.4.2 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/*`.

## 33.4.3 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](http://en.opensuse.org//SDB:Suspend_to_RAM) to get more detailed information on how to modify the suspend and resume process.

## 33.5 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://wiki.opensuse.org/SDB:Suspend\\_to\\_RAM](http://wiki.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 35, *Using NetworkManager* (page 557).

### 34.1 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 34.1, “Overview of Various WLAN Standards” (page 540)). Additionally, many companies implement hardware with proprietary or draft features.

**Table 34.1** *Overview of Various WLAN Standards*

| Name          | Band (GHz) | Maximum Trans-<br>mission Rate<br>(Mbit/s) | Note                                                 |
|---------------|------------|--------------------------------------------|------------------------------------------------------|
| 802.11 Legacy | 2.4        | 2                                          | Outdated; virtual-<br>ly no end devices<br>available |

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.

## 34.2 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 are intending to use WPA security, then 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.passsys.nl> .

# 34.3 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 34.6.3, “Security” (page 552)), 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 34.4, “Encryption” (page 542).

## 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.

## 34.4 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.

## 34.5 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 34.4, “Encryption” (page 542) and Section 34.6.3, “Security” (page 552) 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 34.2, “Operating Modes” (page 540).

#### 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*.

## 34.5.1 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 34.5.2, “Configuration for Access Points” (page 545) or Section 34.5.3, “Establishing an Ad-Hoc Network” (page 549).

Otherwise confirm your changes with *OK* to write the network configuration.

## 34.5.2 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 supported by your WLAN card) . For configuration of networks without an access point, refer to Section 34.5.3, “Establishing an Ad-Hoc Network” (page 549).

### **Procedure 34.1** *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 23.4, “Configuring a Network Connection with YaST” (page 356).
- 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 34.2, “Entering the Encryption Details” (page 546).

**Figure 34.1** *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 34.2** *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 to 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.

## 34.5.3 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 chosen any of the other authentication modes, proceed with Procedure 34.2, “Entering the Encryption Details” (page 546).

- 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.

## 34.5.4 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.

## 34.6 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.

### 34.6.1 Utilities

The package `wireless-tools` contains utilities that allow to set wireless LAN specific parameters and get statistics. See [http://www.hp1.hp.com/personal/Jean\\_Tourrilhes/Linux/Tools.html](http://www.hp1.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.

### 34.6.2 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 `iw-`

`config` 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
```

## 34.6.3 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.

## 34.7 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)?

## 34.7.1 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: ...
```

## 34.7.2 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](http://old-en.opensuse.org/SDB:Name_Resolution_Does_Not_Work_with_Several_Concurrent_DHCP_Clients) .

## 34.7.3 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` .

## 34.8 For More Information

More information can be found on the following pages:

[http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/Wireless.html](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](http://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 do not need to 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 applet in the system tray.

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.

### 35.1 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.

## 35.2 Enabling 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, select *User Controlled with NetworkManager*.
- 4 Click *OK*.
- 5 After choosing the method for managing network connections, set up your network card using automatic configuration via DHCP or a static IP address or configure your modem (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 23.4, “Configuring a Network Connection with YaST” (page 356) and Chapter 34, *Wireless LAN* (page 539).

After having enabled NetworkManager, configure your network connections with the NetworkManager as described in Section 35.3, “Configuring Network Connections” (page 558).

If you want to deactivate NetworkManager and control network the traditional way, choose the *Traditional Method with ifup* option in the *Network Setup Method* field.

## 35.3 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. 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). By default, KNetworkManager also displays comprehensive tooltips for the input fields and options available on each tab.

---

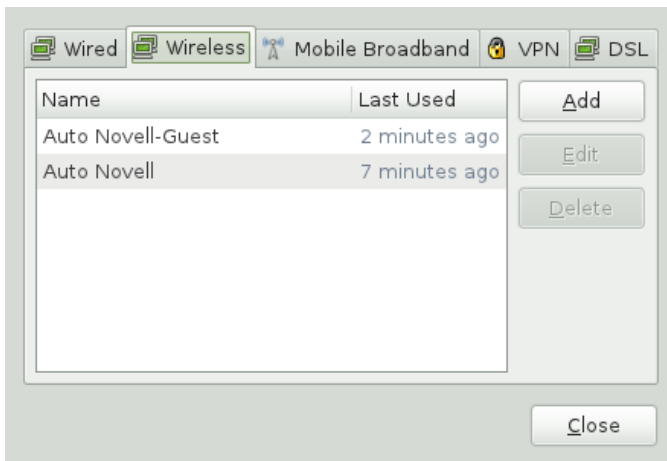
### NOTE: Bluetooth Connections

At the moment, Bluetooth connections cannot be configured with NetworkManager.

---

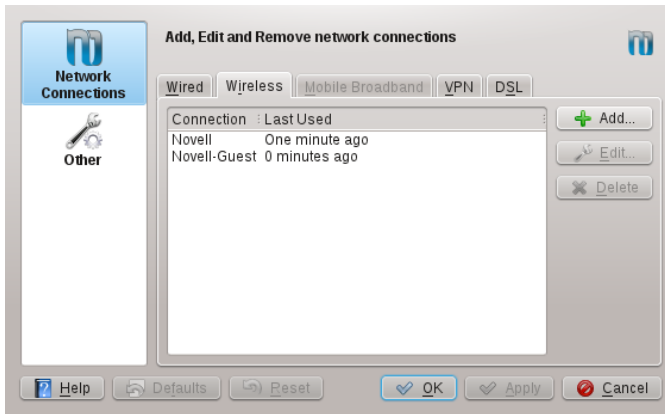
To open the network configuration dialog in GNOME, open the main menu and click the *Network* entry at the right. Alternatively, press **+ F2** and enter `nm-connection-editor` or select *Hardware > Network Connections* in the GNOME Control Center.

**Figure 35.1** GNOME Network Connections Dialog



If you use 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. Alternatively, press **+ F2** and enter `kcm_networkmanagement` or `.`

**Figure 35.2** KDE Network Configuration Dialog



Alternatively, you can also start the configuration dialogs from the NetworkManager applet in the system tray. In KDE, left-click the icon and select *Manage Connections*. In GNOME, right-click the icon and select *Edit Connections*.

---

#### **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.

---

#### **Procedure 35.1** *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 35.7.1, “User and System Connections” (page 569).

Currently, the `system connection` option is not available in KDE. To set up `system connections`, you need to use YaST in this case.

---

#### **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 or Extended Service

Set Identifier (SSID or ESSID) of the network because it cannot be detected automatically.

---

- 1 To add a new connection or edit an existing one, click the tab for the connection type you want to use and click *Add* or choose an existing connection and click *Edit*.
- 2 Enter a *Connection Name* and your connection details.
- 3 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 use KDE, do so by using the *Restrict to Interface* option. If you use GNOME, enter the *MAC address* of the device you want to tie the connection to and confirm your settings.

- 4 If you want NetworkManager to automatically use a certain connection, activate *Connect Automatically* for this connection.
- 5 To turn a connection into a `system connection` activate *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 applet.

**Figure 35.3** *KNetworkManager—Configured and Available Connections*



## 35.4 Using KNetworkManager

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 on the *Panel Tool Box* and choose *Add Widgets*. Double-click on the *Network Management* entry and click on 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.

### 35.4.1 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 menu with available *Connections* and *Interfaces*. Connections are being displayed in the right half of the plasmoid window, while interfaces show up in the left half. The connection and interface currently being used is marked with a blue globe.
- 2 Click on *Networking Interface* listed in the right half of the plasmoid window to get detailed information and statistics for that interface. Switch back to the interface overview by clicking on the blue arrow icon.
- 3 To disconnect an active connection, click on the red icon for the *Networking Interface* in the right half of the plasmoid window.
- 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 35.1, “Adding or Editing Connections” (page 560). 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 NetworkManager icon and uncheck *Enable Networking*.

## 35.4.2 Managing Wireless Network Connections

By default the NetworkManager frontend 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 35.2*** *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 ((E)SSID), click on *hidden network* and press . Fill in the connection details and proceed with *OK*.
- 3 Based on your security settings for NetworkManager (see Section 35.7.2, “Storing Passwords and Credentials” (page 569) for details), you may have to provide your KWallet password.
- 4 NetworkManager automatically connects to the configured network.

### ***Procedure 35.3*** *Managing Active Wireless Connections*

- 1 Click on *WLAN Interface* listed in the right half of the plasmoid window to get detailed information and statistics for that interface. Switch back to the interface overview by clicking on the blue arrow icon.
- 2 To disconnect an active connection, click on 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.

## 35.4.3 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 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 Start the dialog for configuring network connections as described in Section 35.3, “Configuring Network Connections” (page 558) and *Add a Wireless* connection.
- 2 On the *Wireless* tab provide a *Connection name* and an *SSID*. Set *Method* > *Ad-hoc*.
- 3 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 access 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 34.3, “Authentication” (page 541).

---

- 4 On the *IP Address* tab, make sure the *Method* option is set to *Shared*.
- 5 Confirm your configuration with *OK*.

## 35.5 Using GNOME NetworkManager Applet

In GNOME, NetworkManager can be controlled with the GNOME NetworkManager applet. If the network is set up for NetworkManager control, the applet 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 applet is probably not started. Press `+ F2` and enter `nm-applet` to start it manually.

## 35.5.1 Managing Wired Network Connections

If your computer is connected to an existing network with a network cable, use the NetworkManager applet to choose the network connection.

- 1 Left-click the applet icon to show a menu with available networks. The currently used connection is listed on top of the menu, followed by other active connections. Inactive connections are listed under *Available*.
- 2 Right-click the applet and choose *Connection Information* to get detailed information on all active interfaces.
- 3 To disconnect an active connection, left-click the applet and choose its *Disconnect* entry.
- 4 If you want to use a different configuration with the wired network, right-click the applet, choose *Edit Connections* and add another wired connection as described in Procedure 35.1, “Adding or Editing Connections” (page 560). 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 applet icon and uncheck *Enable Networking*.

## 35.5.2 Managing Wireless Network Connections

Available visible wireless networks are listed in the GNOME NetworkManager applet menu under *Available* and *More Networks*. The signal strength of each network is also shown in the menu. Encrypted wireless networks are marked with a lock icon.

### **Procedure 35.4** *Connecting to a Wireless Network*

- 1 To connect to a wireless network, left-click the applet 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 ((E)SSID) and therefore cannot be detected automatically, left-click the NetworkManager icon and choose *Connect to Hidden Wireless Network*.
- 4 In the dialog that opens, enter the ESSID and set encryption parameters on the *Wireless Security* tab if necessary.
- 5 To disable wireless networking, right-click the applet icon and 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.

## **35.5.3 Configuring Your Wireless Card as an Access Point**

If your wireless card supports access point mode, you can use NetworkManager for configuration.

---

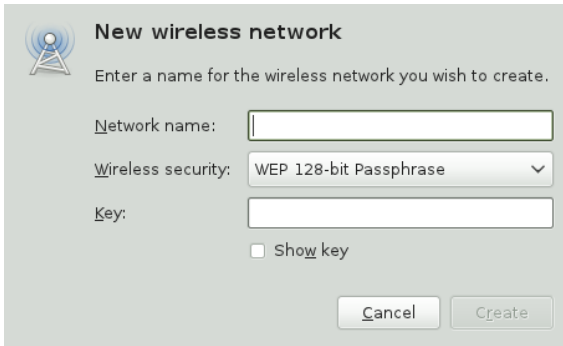
### **NOTE: Availability of Options**

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 applet and select *Create New Wireless Network*.





- 2 Enter a *Network Name* and set the encryption to use with the *Wireless Security* drop-down list.

---

**IMPORTANT: Unprotected Wireless Networks Are a Security Risk**

If you set *Wireless Security* to *None*, everybody can connect to your network, reuse your connectivity and intercept your network connection. To restrict access 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 34.3, “Authentication” (page 541).

---

## 35.6 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 applet.

### NovellVPN

To use this VPN technology, install

- `NetworkManager-novellvpn` and
- `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-open-vpn-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 35.3, “Configuring Network Connections” (page 558).

## 35.7 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 ESSID. This ensures that the card is disconnected.

## 35.7.1 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 connection 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 35.3, “Configuring Network Connections” (page 558).

For KDE, configuring system connections with NetworkManager are currently not supported (use YaST instead).

## 35.7.2 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.

Under 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 options, your credentials are stored in KWalletManager.

## **35.8 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 use 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 35.3, “Configuring Network Connections” (page 558). 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 35.3, “Configuring Network Connections” (page 558). 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 *IP 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 Network-Manager.
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 *DCHP Server* is not already shown as *Allowed Service*, select *DCHP 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 35.3, “Configuring Network Connections” (page 558). 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 *IP 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 35.7, “NetworkManager and Security” (page 568).

## 35.9 Troubleshooting

Connection problems can occur. Some common problems related to NetworkManager include the applet not starting or a missing VPN option. Methods for resolving and preventing these problems depend on the tool used.

### NetworkManager Desktop Applet Does Not Start

The GNOME and KDE NetworkManager applets start automatically if the network is set up for NetworkManager control. If the applet does not start, check if NetworkManager is enabled in YaST as described in Section 35.2, “Enabling NetworkManager” (page 558). 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 applet is installed but is not running for some reason, start it manually. Under GNOME press `+ F2` and enter `nm-applet`. Under KDE click on the *Panel Tool Box* and choose *Add Widgets*. Double-click on the *Network Management* entry and click on the *Panel Tool Box* again.

### NetworkManager Applet Does Not Include the VPN Option

Support for NetworkManager, applets, and VPN for NetworkManager is distributed in separate packages. If your NetworkManager applet does not include the VPN option, check if the packages with NetworkManager support for your VPN technology are installed. For more information, see Section 35.6, “NetworkManager and VPN” (page 567).

### 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`.

## 35.10 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 applets:

- `/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

## 36.1 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 hardwriting 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.

## 36.2 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.

## 36.3 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 36.1** *xvkbd Virtual Keyboard*

|         |           |     |      |    |    |      |     |    |     |     |     |           |              |       |          |     |       |       |
|---------|-----------|-----|------|----|----|------|-----|----|-----|-----|-----|-----------|--------------|-------|----------|-----|-------|-------|
| F1      | F2        | F3  | F4   | F5 | F6 | F7   | F8  | F9 | F10 | F11 | F12 | Backspace | xvkbd (v3.2) |       |          |     |       |       |
| Esc     | !         | @   | #    | \$ | %  | ^    | &   | *  | (   | )   | -   | =         |              | ~     | Num Lock | /   | +     | Focus |
| Tab     | Q         | W   | E    | R  | T  | Y    | U   | I  | O   | P   | {   | }         | Del          | 7     | 8        | 9   | +     |       |
| Control | A         | S   | D    | F  | G  | H    | J   | K  | L   | :   | "   |           | Return       | 4     | 5        | 6   | -     |       |
| Shift   | Z         | X   | C    | V  | B  | N    | M   | <  | >   | ?   | ,   | .         | Com pose     | Shift | 1        | 2   | 3     |       |
| xvkbd   | Caps Lock | Alt | Meta |    |    | Meta | Alt | ←  | →   | ↑   | ↓   | Focus     | 0            | Ins   |          | Del | Enter |       |

If you want to use xvkbd after login, start it from the main menu or with `xvkbd` from a shell.

## 36.4 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 `krandrtray` 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 36.7, “Troubleshooting” (page 582) for more information.

## 36.5 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.

### 36.5.1 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).

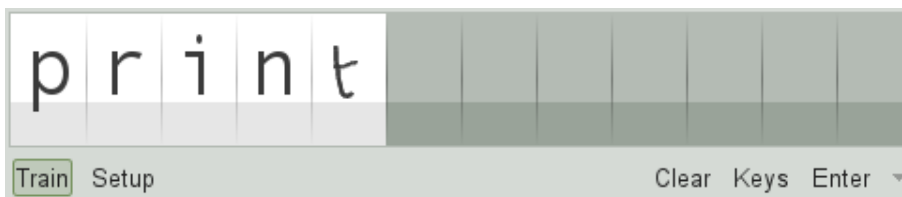
#### ***Procedure 36.1 Training CellWriter***

- 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 36.2** *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.

## 36.5.2 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 alpha-

bet that resembles the Graffiti\* alphabet. When activated, xstroke sends the input to the currently focused window.

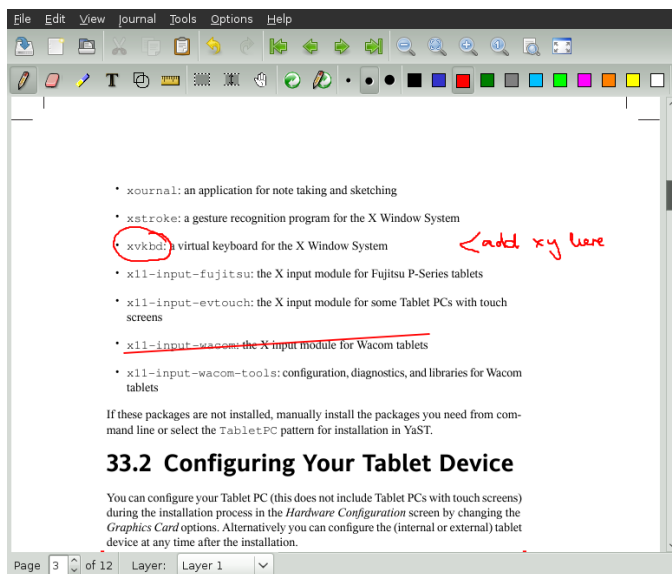
- 1 Start xstroke from the main menu or with `xstroke` from a shell. This adds a pencil icon to your system tray.
- 2 Start the application for which you want to create text input with the pen (for example, a terminal window, a text editor or an LibreOffice Writer).
- 3 To activate the gesture recognition 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.

## 36.6 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*.

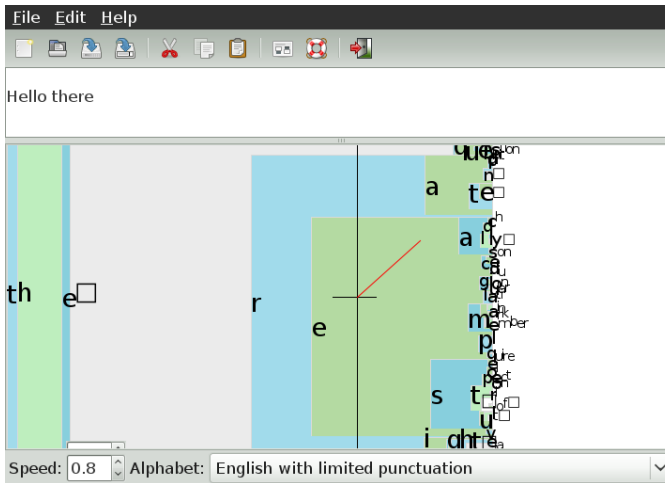
**Figure 36.3** Annotating a PDF with Xournal



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.

**Figure 36.4** *Editing Texts with Dasher*



The Dasher concept works for many languages. For more information, refer to the Dasher Web site, which offers comprehensive documentation, demonstrations and training texts. Find it at <http://www.inference.phy.cam.ac.uk/dasher/>

## 36.7 Troubleshooting

### Virtual Keyboard Does Not Appear on Login Screen

Occasionally, the virtual keyboard is not displayed on the login screen. To solve this, restart the X server by pressing `++` or press the appropriate key on your Tablet PC (if you use a slate model without integrated keyboard). If the virtual keyboard still does not show, connect an external keyboard to your slate model and log in using the hardware keyboard.

### Orientation of the Wacom Graphics Tablets Does Not Change

With the `xrandr` command, you can change the orientation of your display from within a shell. Enter `xrandr --help` to view the options available. To simultaneously change the orientation of your graphics tablet, the command needs to be modified as described below:

- For normal orientation (0° rotation):

```
xrandr -o 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.

## 36.8 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://risu-jin.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.

## 37.1 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 37.4, “Accessing Files on Different OS on the Same Computer” (page 589) 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 37.2, “Access Methods” (page 587).

### 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.

## 37.2 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 37.5.4, “Copying Files with FTP” (page 596) 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 28, *Sharing File Systems with NFS* (page 441).

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 37.5.2, “Transferring Files with rsync” (page 592) 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 37.5.3, “Transferring Files with Unison” (page 594) 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 plugins. 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 29, *Samba* (page 455) 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 37.8, “Sharing Files between Linux and Windows with Samba” (page 602).

## 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 13, *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.

# 37.3 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](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 `rcsshd start as root`.

Proceed as follows:

### ***Procedure 37.1 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 37.2 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.

## **37.4 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 **+ 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.

---

## 37.5 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 |
|-----------------|---------------------|

---

### 37.5.1 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 “`/etc/hosts` ” (page 384).) 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 **+ 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
```

## 37.5.2 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 37.3, “Advanced Setup for `rsync` Synchronization” (page 593)).

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.

## 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/
```

## 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 37.3*** *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 `inserv 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.

## 37.5.3 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`.

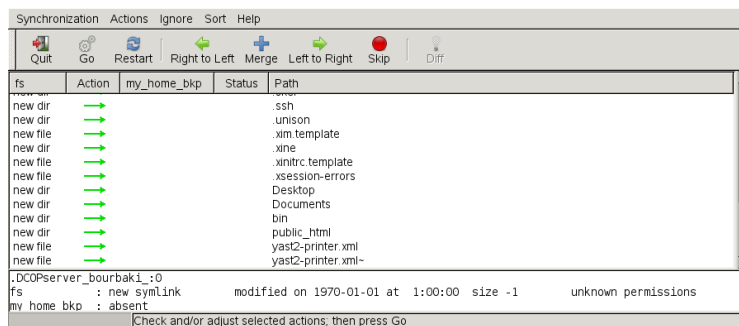
## Using the GUI

To synchronize different directories with Unison's GUI, proceed as follows:

- 1 Start Unison by pressing **+ 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 37.1** *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.

## 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.

## 37.5.4 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 graph-

ical 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.

## 37.6 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/~sgtatham/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.

---

## 37.7 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.

### 37.7.1 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 9, *Managing Users with YaST* (page 149) 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 `rcnfsdserver start` as `root`. To stop the server, enter `rcnfsdserver 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).

---

## 37.7.2 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.

### 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 **+ 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 603).

**2** Log in with the username and password. The password is set in Step 4 (page 603) 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.

## 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.

## 37.8 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 “Accessing Shares with KDE and GNOME” (page 601). 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 37.4, “Setting Up a Samba Server” (page 602).

---

#### **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 37.4*** *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 `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 , 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
```

## 37.9 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/Network_File_System)
- [http://en.wikipedia.org/wiki/File\\_Transfer\\_Protocol](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](http://en.wikipedia.org/wiki/Samba_software)

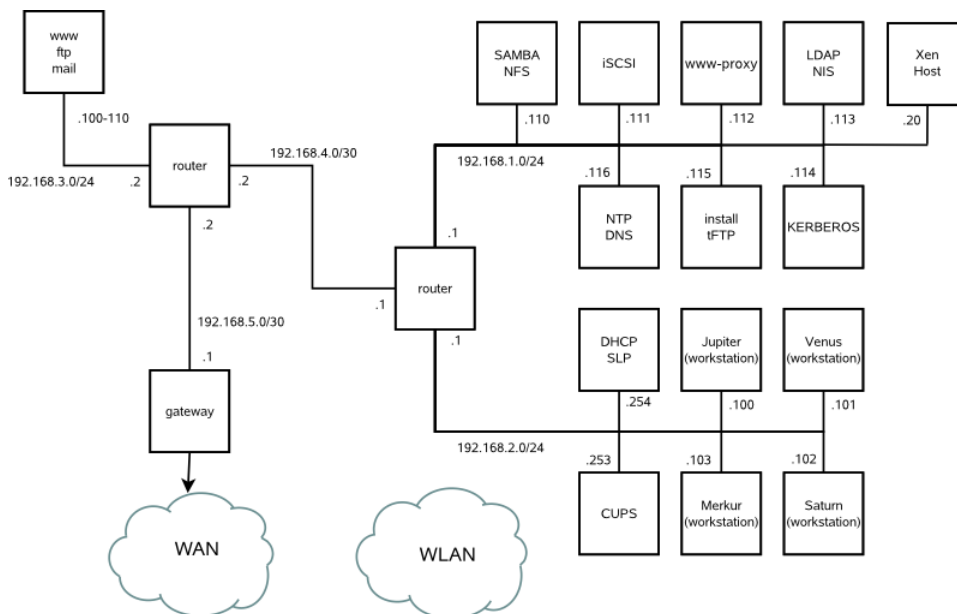




# A

## 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

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. 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.

## APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the

covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

**A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

**B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

**C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.

**D.** Preserve all the copyright notices of the Document.

**E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

**F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

**G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

**H.** Include an unaltered copy of this License.

**I.** Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

**J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

**K.** For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

**L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

**M.** Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

**N.** Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

**O.** Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the



same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

## COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

## ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.2  
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.