

SUSE Linux Enterprise Desktop

11

www.novell.com

February 23, 2009

Administration Guide



Administration Guide

All content is copyright © 2006- 2009 Novell, Inc.

Legal Notice

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

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

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

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

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

Contents

About This Guide	ix
Part I Support and Common Tasks	1
1 YaST Online Update	3
1.1 Installing Patches Manually Using the Qt Interface	4
1.2 Installing Patches Manually Using the gtk Interface	6
1.3 Automatic Online Update	7
2 Gathering System Information for Support	9
2.1 Novell Support Link Overview	9
2.2 Using Supportconfig	10
2.3 Submitting Information to Novell	12
2.4 For More Information	14
3 YaST in Text Mode	15
3.1 Navigation in Modules	16
3.2 Restriction of Key Combinations	17
3.3 YaST Command Line Options	18
4 GNOME Configuration for Administrators	21
4.1 The GConf System	21
4.2 Customizing Main Menu, Panel, and Application Browser	24
4.3 Starting Applications Automatically	24
4.4 Automounting and Managing Media Devices	25
4.5 Changing Preferred Applications	25

4.6	Managing Profiles Using Sabayon	25
4.7	Adding Document Templates	30
4.8	Desktop Lock Down Features	30
4.9	For More Information	31
5	Managing Software with Command Line Tools	33
5.1	Using Zypper	33
5.2	RPM—the Package Manager	38
6	Accessing Remote Desktops with Nomad	51
6.1	Nomad Prerequisites	52
6.2	Installation and Set-Up	53
6.3	Using Nomad	54
6.4	Troubleshooting	54
6.5	For More Information	55
7	Bash and Bash Scripts	57
7.1	What is The Shell?	57
7.2	Writing Shell Scripts	63
7.3	Redirecting Command Events	64
7.4	Using Aliases	65
7.5	Using Variables in Bash	65
7.6	Grouping And Combining Commands	68
7.7	Working with Common Flow Constructs	69
7.8	For More Information	70
Part II	System	71
8	32-Bit and 64-Bit Applications in a 64-Bit System Environment	73
8.1	Runtime Support	73
8.2	Software Development	74
8.3	Software Compilation on Biarch Platforms	75
8.4	Kernel Specifications	76
9	Bootng and Configuring a Linux System	77
9.1	The Linux Boot Process	77
9.2	The init Process	81
9.3	System Configuration via /etc/sysconfig	90

10	The Boot Loader GRUB	93
10.1	Booting with GRUB	94
10.2	Configuring the Boot Loader with YaST	103
10.3	Uninstalling the Linux Boot Loader	109
10.4	Creating Boot CDs	109
10.5	The Graphical SUSE Screen	110
10.6	Troubleshooting	111
10.7	For More Information	112
11	Special System Features	113
11.1	Information about Special Software Packages	113
11.2	Virtual Consoles	120
11.3	Keyboard Mapping	121
11.4	Language and Country-Specific Settings	122
12	Printer Operation	127
12.1	The Workflow of the Printing System	129
12.2	Methods and Protocols for Connecting Printers	129
12.3	Installing the Software	130
12.4	Network Printers	131
12.5	Graphical Printing Interfaces	133
12.6	Printing from the Command Line	134
12.7	Special Features in SUSE Linux Enterprise Desktop	134
12.8	Troubleshooting	137
13	Dynamic Kernel Device Management with udev	145
13.1	The /dev Directory	145
13.2	Kernel uevents and udev	146
13.3	Drivers, Kernel Modules, and Devices	146
13.4	Bootling and Initial Device Setup	147
13.5	Monitoring the Running udev Daemon	147
13.6	Influencing Kernel Device Event Handling with udev Rules	149
13.7	Persistent Device Naming	156
13.8	Files used by udev	156
13.9	For More Information	157
14	The X Window System	159
14.1	Manually Configuring the X Window System	159
14.2	Installing and Configuring Fonts	166
14.3	For More Information	172

15	Accessing File Systems with FUSE	173
15.1	Configuring FUSE	173
15.2	Mounting an NTFS Partition	173
15.3	Mounting Remote File System with SSHFS	174
15.4	Mounting an ISO File System	175
15.5	Available FUSE Plug-ins	175
15.6	For More Information	176
Part III	Mobile Computers	177
16	Mobile Computing with Linux	179
16.1	Laptops	179
16.2	Mobile Hardware	187
16.3	Cellular Phones and PDAs	188
16.4	For More Information	188
17	Power Management	191
17.1	Power Saving Functions	191
17.2	ACPI	192
17.3	Rest for the Hard Disk	196
17.4	Troubleshooting	198
17.5	For More Information	200
18	Using Tablet PCs	201
18.1	Installing Tablet PC Packages	202
18.2	Configuring Your Tablet Device	203
18.3	Using the Virtual Keyboard	204
18.4	Rotating Your Display	205
18.5	Using Gesture Recognition	205
18.6	Taking Notes and Sketching with the Pen	208
18.7	Troubleshooting	210
18.8	For More Information	211
Part IV	Services	213
19	Basic Networking	215
19.1	IP Addresses and Routing	218
19.2	IPv6—The Next Generation Internet	221
19.3	Name Resolution	230

19.4	Configuring a Network Connection with YaST	232
19.5	NetworkManager	252
19.6	Configuring a Network Connection Manually	253
19.7	smpppd as Dial-up Assistant	268
20	Wireless Communication	271
20.1	Wireless LAN	271
21	SLP Services in the Network	281
21.1	Installation	281
21.2	Activating SLP	282
21.3	SLP Front-Ends in SUSE Linux Enterprise Desktop	282
21.4	Providing Services via SLP	283
21.5	For More Information	284
22	Time Synchronization with NTP	285
22.1	Configuring an NTP Client with YaST	286
22.2	Manually Configuring ntp in the Network	289
22.3	Setting Up a Local Reference Clock	289
23	Using NetworkManager	291
23.1	Use Cases for NetworkManager	291
23.2	Enabling NetworkManager	292
23.3	Configuring Network Connections	293
23.4	Using KDE NetworkManager Widget	294
23.5	Using GNOME NetworkManager Applet	295
23.6	NetworkManager and VPN	297
23.7	NetworkManager and Security	298
23.8	Frequently Asked Questions	299
23.9	Troubleshooting	301
23.10	For More Information	302
24	Samba	305
24.1	Terminology	305
24.2	Configuring a Samba Server	307
24.3	Configuring Clients	307
24.4	Samba as Login Server	307
24.5	For More Information	308

25	Sharing File Systems with NFS	311
25.1	Installing the Required Software	311
25.2	Importing File Systems with YaST	311
25.3	Importing File Systems Manually	312
25.4	NFS with Kerberos	314
25.5	For More Information	315
26	File Synchronization	317
26.1	Available Data Synchronization Software	317
26.2	Determining Factors for Selecting a Program	319
26.3	Introduction to CVS	322
26.4	Introduction to rsync	324
26.5	For More Information	326

About This Guide

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

This guide contains the following:

Administration

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

System

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

Mobile Computing

Laptops, and the communication between mobile devices like PDAs, or cellular phones and SUSE Linux Enterprise need some special attention. Take care for power conservation and for the integration of different devices into a changing network environment. Also get in touch with the background technologies that provide the needed functionality.

Services

SUSE Linux Enterprise is designed to be a network operating system. SUSE® Linux Enterprise Desktop includes client support for many network services. It integrates well into heterogeneous environments including MS Windows clients and servers.

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

For an overview of the documentation available for your product and the latest documentation updates, refer to <http://www.novell.com/documentation>.

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:

GNOME User Guide (↑GNOME User Guide)

Introduces the GNOME desktop of SUSE Linux Enterprise Desktop. It guides you through using and configuring the desktop and helps you perform key tasks. It is intended mainly for end users who want to make efficient use of GNOME desktop as their default desktop.

Application Guide (↑Application Guide)

Learn how to use and configure key desktop applications on SUSE Linux Enterprise Desktop. This guide introduces browsers and e-mail clients as well as office applications and collaboration tools. It also covers graphics and multimedia applications.

Deployment Guide (↑Deployment Guide)

Shows how to install single or multiple systems and how to exploit the product inherent capabilities for a deployment infrastructure. Choose from various approaches, ranging from a local installation or a network installation server to a mass deployment using a remote-controlled, highly-customized, and automated installation technique.

Administration Guide (page 1)

Covers system administration tasks like maintaining, monitoring and customizing an initially installed system.

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 Xen (↑Virtualization with Xen)

Offers an introduction to virtualization technology of your product. It features an overview of the various fields of application and installation types of each of the platforms supported by SUSE Linux Enterprise Server as well as a short description of the installation procedure.

In addition to the comprehensive manuals, several quick start guides are available:

Installation Quick Start (↑Installation Quick Start)

Lists the system requirements and guides you step-by-step through the installation of SUSE Linux Enterprise Desktop from DVD, or from an ISO image.

Linux Audit Quick Start

Gives a short overview how to enable and configure the auditing system and how to execute key tasks such as setting up audit rules, generating reports, and analyzing the log files.

Novell AppArmor Quick Start

Helps you understand the main concepts behind Novell® AppArmor.

Find HTML versions of most SUSE Linux Enterprise Desktop 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/documentation> where you can download PDF or HTML versions of the manuals for your product.

2 Feedback

Several feedback channels are available:

- To report bugs for a product component or to submit enhancements requests, please use <https://bugzilla.novell.com/>. If you are new to Bugzilla, you might find the *Bug Writing FAQs* helpful, available from the Novell Bugzilla home page.
- We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation and enter your comments there.

3 Documentation Conventions

The following typographical conventions are used in this manual:

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

Part I. Support and Common Tasks

YaST Online Update

Novell offers a continuous stream of software security updates for your product. By default openSUSE Updater is used to keep your system up-to-date. Refer to Section “Keeping the System Up-to-date” (Chapter 6, *Installing or Removing Software*, ↑Deployment Guide) for further information on openSUSE Updater. This chapter covers the alternative tool for updating software packages: YaST Online Update.

The current patches for SUSE® Linux Enterprise Desktop are available from an update software repository. If you have registered your product during the installation, an update repository is already configured. If you have not registered SUSE Linux Enterprise Desktop, you can do so by running *Software > Online Update Configuration* in YaST and start *Advanced > Register for Support and Get Update Repository*. 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 “Managing Software Repositories and Services” (Chapter 6, *Installing or Removing Software*, ↑Deployment Guide).

NOTE: Error on Accessing the Update Catalog

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

In case of an access denial to the update catalog you will see a warning message with a recommendation to visit the Novell Customer Center and check your

subscription. The Novell Customer Center is available at <http://www.novell.com/center/>.

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, whereas *Optional* updates fix non-security relevant issues or provide enhancements.

To install updates and improvements with YaST, run *Software > Online Update* from YaST. All new patches (except the optional ones) that are currently available for your system are already marked for installation. Clicking *Accept* or *Apply* automatically installs these patches. After the installation has completed, confirm with *Finish*. Your system is now up-to-date.

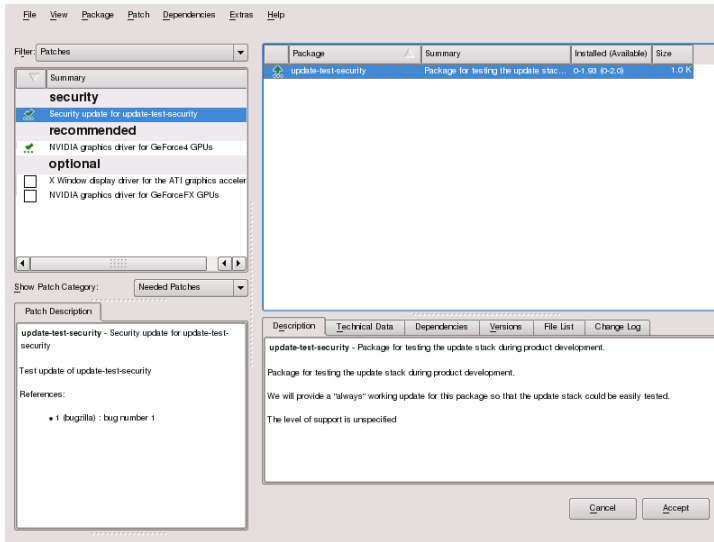
TIP: Disabling deltarpm

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 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`.

1.1 Installing Patches Manually Using the Qt Interface

The *Online Update* window consists of four sections. The list of all patches available is on the left. Find the description of the selected patch displayed below the list of patches. The right column lists the packages included in the selected patch (a patch can consist of several packages) and, below, a detailed description of the selected package. Optionally, the disk usage can be displayed at the bottom of the left column (this display is faded out by default—use the dotted slider to make it visible).

Figure 1.1 *YaST Online Update*



The patch display lists the available patches for SUSE Linux Enterprise Desktop. The patches are sorted by security relevance. *security*, *recommended*, and *optional*. There are three different views on patches. Use *Show Patch Category* to toggle the views:

Needed Patches (default view)

Currently not 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 which requirements already have been fulfilled (because it has already been updated from another source).

All Patches

All patches available for SUSE Linux Enterprise Desktop.

A list entry consists of a symbol and the patch name. For a list of possible symbols, press **Shift + F1**. Actions required by *Security* and *Recommended* patches are automatically preset. These actions are *Autoinstall*, *Autoupdate*, or *Autodelete*. Actions for *Optional* patches are not preset—right-click on a patch and choose an action from the list.

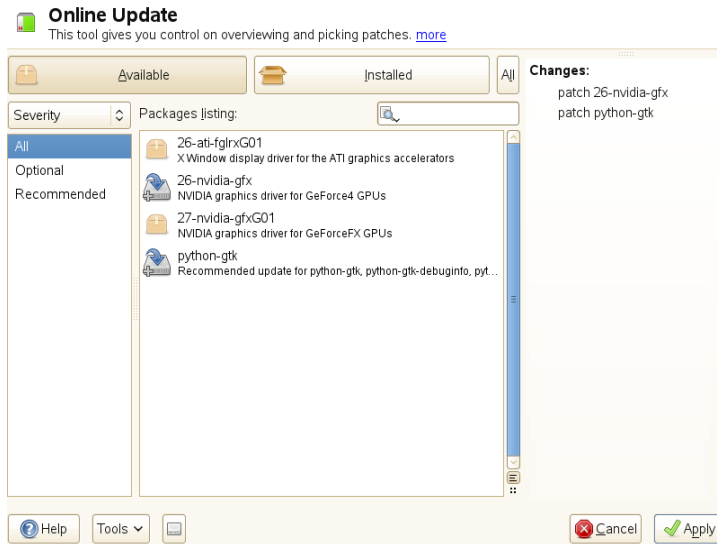
If you install an up-to-date package from a repository other than the update repository, the requirements of a patch for this package may be fulfilled with this installation. In this case a check mark is displayed in front of the patch summary. The patch will be visible in the list until you mark it 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.

Most patches include updates for several packages. If you want to change actions for single packages, right-click on a package in the package window and choose an action. Once you have marked all patches and packages as desired, proceed with *Accept*.

1.2 Installing Patches Manually Using the gtk Interface

The *Online Update* window consists of two main sections. The left pane lists all patches and provides different filters for the patch list. See the right pane for a list of changes that will be carried out once you *Apply* them.

Figure 1.2 *YaST Online Update*



Patch List Filters

Available

Currently not installed patches that apply to packages installed on your system.

Installed

Patches that are already installed.

All

Patches that are either already installed or available.

Severity

Only show *Optional*, *Recommended*, or *Security* patches. By default, *All* patches are shown.

Repositories

This filter lets you display the patches per repository.

Packages Listing

Apply your custom filter here.

Click on a patch entry to open a row with detailed information, about the patch in the bottom area of the left pane. 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.

1.3 Automatic Online Update

YaST also offers the possibility to set up an automatic update. Open *Software > Online Update Configuration*. Check *Automatic Online Update* and choose whether to update *Daily*, *Weekly*, or *Monthly*. Some patches, such as kernel updates, require user interaction, which would cause the automatic update procedure to stop. Therefore you should check *Skip Interactive Patches*, if you want the update procedure to proceed fully automatically. Having done so, you should run a manual *Online Update* from time to time in order to install patches that require interaction.

Gathering System Information for Support

2

Once a problem arises, `supportconfig` can be used to collect system information, like the current kernel version being used, the hardware, RPM database, partitions, and others. The result is used to help the Novell Support Center finding your problem.

2.1 Novell Support Link Overview

Novell Support Link (NSL) is new to SUSE Linux Enterprise Desktop. It is a tool that gathers system information and allows you to upload that information to another server for further analysis. Novell Support Center uses Novell Support Link to gather system information from problematic servers and sends the information to Novell's public FTP server. System information gathered includes: current kernel version being used, the hardware, RPM database, partitions, and more. The result is used to help the Novell Support Center resolve your open service request.

There are two ways to use Novell Support Link:

1. Use the YaST Support module,
2. Use the command line utility `supportconfig`.

The YaST Support module calls `supportconfig` to gather system information.

2.2 Using Supportconfig

The following sections describes how to use `supportconfig` with YaST, from the command line, and what options do you have.

2.2.1 Using YaST to Collect Information

To use YaST to gather your system information, proceed as follows:

- 1 Open the URL <http://www.novell.com/center/eservice> and create a service request number.
- 2 Start YaST.
- 3 Open the *Support* module.
- 4 Click on *Create report tarball*.
- 5 Select an option from the radio button list. If you want to test it first, use *Only gather a minimum amount of info*. Proceed with *Next*.
- 6 Enter your contact information. Use your service request number from **Step 1** (page 10) and enter it in the text field labeled *Novell 11 digit service request number*. Proceed with *Next*.
- 7 The information gathering is being started. After the process is finished, continue with *Next*.
- 8 Review the data collection and use *Remove from Data*, if you do not need the respective filename. Continue with *Next*.
- 9 Save your tarball. If you want to upload to the Novell customer center, make sure *Upload log files tarball into URL* is activated. Finish with *Next*.

2.2.2 Using Supportconfig Directly to Collect Information

To use `supportconfig` from the the commandline, proceed as follows:

- 1 Open a shell and become `root`.
- 2 Running `supportconfig` without any options gathers the default system information.
- 3 Wait for the tool to complete.
- 4 The default archive location is `/var/log` with the filename format `nts_HOST_DATE_TIME.tbz`

2.2.3 Common Supportconfig Options

The `supportconfig` utility has a variety of startup options. You can see these options with `supportconfig -h` or use the man page. Generally `supportconfig` is run without any options. The following is a summary of some of the more common startup options:

- Use the minimal option (`-m`) to reduce the size of your information being gathered:

```
supportconfig -m
```

- Include additional contact information in the output (in one line):

```
supportconfig -E tux@example.org -N "Tux Penguin" -O "Penguin Inc." ...
```

- While troubleshooting a problem, you may want to gather information only about the area of the problem you are currently working on. For example, if you have problems with LVM, and recently found the problem with the default `supportconfig` output. After making changes, you want to gather the current LVM information. The following would gather the minimum `supportconfig` information and LVM only.

```
supportconfig -i LVM
```

To see a complete feature list, run:

```
supportconfig -F
```

- Use the `-u` and `-r` options to upload a supportconfig tar ball with the associated service request number. For example, say you have opened a service request with Novell, and the tracking number is 12345678901, then run the following:

```
supportconfig -ur 12345678901
```

2.3 Submitting Information to Novell

You can use the YaST Support module or the supportconfig command line utility to submit system information to Novell. When you experience a server issue and would like Novell's assistance, you will need to open a service request and submit your server information to Novell. Both YaST and command line methods are described.

Procedure 2.1 *Submitting Information to Novell with YaST*

- 1 Open the URL <http://www.novell.com/center/eservice> and create a service request number.
- 2 Write down your 11 digit service request number. The following examples will assume the service request number is 12345678901.
- 3 Click on *Create report tarball* in the YaST Support module window.
- 4 Select the *Use custom* radio button. Proceed with *Next*.
- 5 Enter your contact information, fill in *Novell 11 digit service request number* and include Novell's upload target URL.
 - For the secure upload target, use: <https://secure-www.novell.com/upload?appname=supportconfig&file={tarball}>.
 - For the normal FTP upload target, use: <ftp://ftp.novell.com/incoming>.

Proceed with *Next*. Information gathering starts. After the process is finished, continue with *Next*.

- 6 Review the data collection, and use *Remove from Data* to remove any files you want excluded from the tar ball uploaded to Novell. Continue with *Next*.
- 7 By default a copy of the tarball will be saved in `/root`. Confirm you are using one of the Novell upload targets described above and the *Upload log files tarball into URL* is activated. Finish with *Next*.
- 8 Click *Finish*.

Procedure 2.2 *Submitting Information to Novell with supportconfig*

- 1 Open the URL <http://www.novell.com/center/eservice> and create a service request number.
- 2 Write down your 11 digit service request number. The following examples will assume the service request number is 12345678901.
- 3 Servers with Internet connectivity:

3a To use the default upload target, run:

```
supportconfig -ur 12345678901
```

3b For the secure upload target, use the following on one line:

```
supportconfig -r 12345678901 -U  
'https://secure-www.novell.com/upload?appname=supportconfig&file={tarball}'
```

- 4 Servers *without* Internet connectivity

4a Run the following:

```
supportconfig -r 12345678901
```

4b Manually upload the `/var/log/nts_SR12345678901*tbz` tarball to Novell's FTP server (<ftp://ftp.novell.com/incoming>).

4c You can also attach the tar ball to your service request using the service request URL: <http://www.novell.com/center/eservice>.

5 Once the tar ball is in the <ftp://ftp.novell.com/incoming> directory, it will get automatically attached to your service request.

2.4 For More Information

Find more information about gathering system information in the following documents:

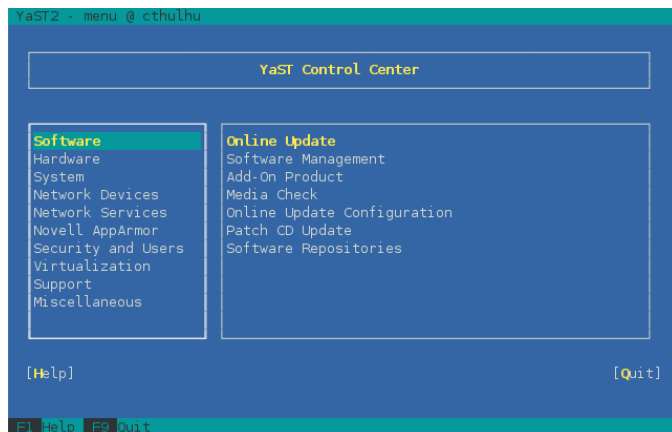
- `man supportconfig`—The manpage of supportconfig
- `man supportconfig.conf`—The manpage of supportconfig configuration file
- <http://www.novell.com/communities/print/node/4097>—A Basic Server Health Check with Supportconfig
- <http://www.novell.com/communities/print/node/4827>—Create Your Own Supportconfig Plugin
- <http://www.novell.com/communities/print/node/4800>—Creating a Central Supportconfig Repository

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

Figure 3.1 *Main Window of YaST in Text Mode*



When YaST is started in text mode, the YaST Control Center appears first, see [Figure 3.1](#). The main window consists of three areas. The left frame, which is surrounded by a thick white border, features the categories to which the various modules belong. The

active category is indicated by a colored background. The right frame, which is surrounded by a thin white border, provides an overview of the modules available in the active category. The bottom frame contains the buttons for *Help* and *Quit*.

When the YaST Control Center is started, the category *Software* is selected automatically. Use ↓ and ↑ to change the category. To start a module from the selected category, press →. The module selection now appears with a thick border. Use ↓ and ↑ to select the desired module. Keep the arrow keys pressed to scroll through the list of available modules. When a module is selected, the module title appears with a colored background.

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

3.1 Navigation in Modules

The following description of the control elements in the YaST modules assumes that all function keys and Alt key combinations work and are not assigned to different global functions. Read [Section 3.2, “Restriction of Key Combinations”](#) (page 17) 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 Alt + Tab or Shift + 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 Shift + → or Shift + ← to scroll horizontally to the right and left. Alternatively, use Ctrl + E or Ctrl + A. This combination can also be used if using → or ← would result in changing the active frame or the current selection list, as in the Control Center.

Buttons, Radio Buttons, and Check Boxes

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

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

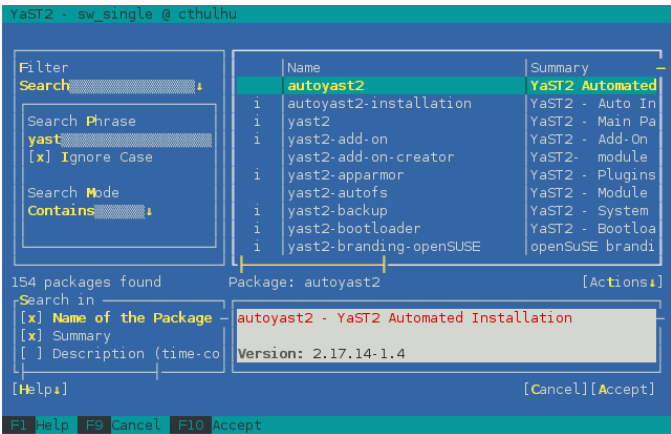
Function Keys

The F keys (F1 to F12) enable quick access to the various buttons. Available F key shortcuts are shown in the bottom line of the YaST screen. Which function keys are actually mapped to which buttons depends on the active YaST module, because the different modules offer different buttons (Details, Info, Add, Delete, etc.). Use F10 for *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 left part of the window to select configuration dialogs. In ncurses mode, Enter must be pressed after selection in the navigation tree to show the selected dialog. This is intentional behaviour to save time consuming redraws when browsing through the navigation tree.

Figure 3.2 The Software Installation Module



3.2 Restriction of Key Combinations

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

Replacing Alt with Esc

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

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

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

Restriction of Function Keys

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

3.3 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
```

3.3.1 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`.

3.3.2 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 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 5.1, “Using Zypper”](#) (page 33).

3.3.3 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 a 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.
```


GNOME Configuration for Administrators

This chapter discusses GNOME configuration options Administrators can adjust system-wide, such as customizing menus, installing themes, configuring fonts, changing preferred applications, and locking down capabilities.

These configuration options are stored in the GConf system. Access the GConf system with tools such as the `gconftool-2` command line interface or the `gconf-editor` GUI.

4.1 The GConf System

The GNOME desktop makes use of GConf for configuration. It is a hierarchically structured database or registry where the user can change his own settings and the system administrator can set default or mandatory values for all users. You reach GConf settings by specifying access paths such as

`/desktop/gnome/background/picture_filename`—this, for example, is the key holding the file name of the desktop background picture.

Use the graphical `gconf-editor` if you want to browse through all options conveniently. For a short usage description of `gconf-editor`, see [Section 4.1.1, “The Graphical `gconf-editor`”](#) (page 22). If you need a scriptable solution, see [Section 4.1.2, “The `gconftool-2` Command Line Interface”](#) (page 23).

WARNING: GNOME Control Center Dialogs

Accessing the Gconf System directly can result in an unusable system, if done inappropriately.

Unexperienced users who want to adjust some common desktop features only, are recommended to use the GNOME Control Center configuration dialogs. To start the GNOME Control Center, click *Computer > Control Center*. For more information, see Section “The Control Center” (Chapter 3, *Customizing Your Settings*, ↑GNOME User Guide).

4.1.1 The Graphical gconf-editor

The GNOME application gconf-editor lets you browse through GConf settings and change them interactively. To start gconf-editor in the normal view (*Settings Window*) click *Computer > More Applications* and then in the *System* group click *GNOME Configuration Editor*.

In the normal view the user can change settings for his own desktop and the administrator can prepare settings for specifying default or mandatory values. For example, if you want to enable the typing break feature as a mandatory feature for all users, proceed as follows:

- 1 As `root` start gconf-editor.
- 2 In the tree pane on the left side, expand `/desktop/gnome/typing_break`.
- 3 Right-click *enabled* and select *Set as Mandatory*. Once this is done, you can actually enable it.
- 4 Open the *Mandatory settings* window by clicking *File > New Mandatory Window*.
- 5 In the tree pane of the *Mandatory settings* window expand `/desktop/gnome/typing_break` click *enabled*.
- 6 Close the window to save the setting by clicking *File > Close Window*.

For more information about gconf-editor, see the Configuration Editor Manual at <http://library.gnome.org/users/gconf-editor/stable/>.

4.1.2 The gconftool-2 Command Line Interface

To change settings from the command line or within scripts, use `gconftool-2`. Here are some examples:

As `root`, use the following command to list the values of all keys:

```
gconftool-2 --recursive-list /
```

If you are interested in a subset only, specify an access path such as `/desktop/gnome/typing_break`:

```
gconftool-2 --recursive-list /desktop/gnome/typing_break
```

To list mandatory settings:

```
gconftool-2 --recursive-list \  
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory /
```

To set a mandatory setting such as `typing_break`:

```
gconftool-2 \  
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \  
  --type bool \  
  --set /desktop/gnome/typing_break/enabled true
```

To unset a mandatory setting:

```
gconftool-2 \  
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \  
  --unset /desktop/gnome/typing_break/enabled
```

For default settings, use `/etc/gconf/gconf.xml.default`.

For more information about `gconftool-2`, see the GNOME Desktop System Administration Guide, Section GConf Command Line Tool at <http://library.gnome.org/admin/system-admin-guide/stable/gconf-6.html.en> and the `gconftool-2` manpage (`man gconftool-2`).

4.2 Customizing Main Menu, Panel, and Application Browser

Control the default items shown in various sections of the main menu (*Computer*) by customizing the following files:

- **/usr/share/gnome-main-menu/applications.xbel:** List of default favorite applications.
- **/usr/share/gnome-main-menu/documents.xbel:** List of default favorite documents.
- **/usr/share/gnome-main-menu/system-items.xbel:** Items shown in the system section.

With `gconf-editor` you can customize displaying how many items to show:

- **/desktop/gnome/applications/main-menu/file-area/min_recent_items:** Minimal number of recent items.
- **/desktop/gnome/applications/main-menu/file-area/max_total_items:** Maximal number of total items.

You can customize the application browser in various ways, for example its behavior when users launch items or the number of items displayed in the *New Applications* category. Look up the keys

`/desktop/gnome/applications/main-menu/ab_*` with `gconf-editor`.

For more information, see the Section Customizing Menus in the GNOME Desktop System Administration Guide at <http://library.gnome.org/admin/system-admin-guide/stable/menustructure-0.html.en>.

4.3 Starting Applications Automatically

To automatically start applications in GNOME, use one of the following methods:

- **To run applications for every user:** Put `.desktop` files in `/usr/share/gnome/autostart`.
- **To run applications for an individual user:** Put `.desktop` files in `~/.config/autostart`.

To disable an application that starts automatically, add `X-Autostart-enabled=false` to the `.desktop` file.

4.4 Automounting and Managing Media Devices

Nautilus (`nautilus`) monitors volume-related events and responds with a user-specified policy. You can use the Nautilus to automatically mount hot-plugged drives and inserted removable media, automatically run programs, automatically play audio CDs and video DVDs, and automatically import photos from a digital camera.

System administrators can set system-wide defaults. For more information, see [Section 4.5, “Changing Preferred Applications”](#) (page 25).

4.5 Changing Preferred Applications

To change users' preferred applications, edit `/etc/gnome_defaults.conf`. Find further hints within this file.

After editing the file, run `SuSEconfig --module glib2`.

For more information about MIME types, see <http://www.freedesktop.org/Standards/shared-mime-info-spec>.

4.6 Managing Profiles Using Sabayon

Sabayon is a system administration tool you can use to create and apply desktop environment profiles. A profile is a collection of default settings and restrictions that can

be applied to either individual users or groups of users. Sabayon lets you edit GConf defaults and mandatory keys using a graphical tool.

Profile definition is done through a graphical session similar to the one a user would be running, only inside a desktop window. You can change properties (such as the desktop background, toolbars, and available applets) in the usual way. Sabayon also detects changes to the default settings in most desktop applications.

Files or documents that are left in the simulated home directory or on the desktop are included in the finished profile. This includes many application-specific databases, such as Tomboy notes. Using this mechanism, it is easy to supply introductory notes or templates in a manner easily accessible to new users.

A user profile can inherit its settings from a parent profile, overriding or adding specific values. This enables hierarchical sets of settings. For example, you can define an Employee profile and derive Artist and Quality Assurance profiles from that.

In addition to providing defaults, Sabayon can also lock down settings. This makes the setting resistant to change by users. For instance, you can specify that the desktop background cannot be changed to something other than the default you provide. It prevents casual tampering with settings, potentially reducing the number of helpdesk calls, and enabling kiosk-like environments. However, it does not provide absolute security and should not be relied on for such.

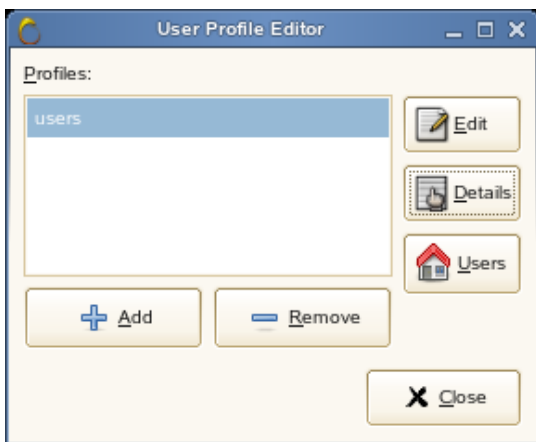
Sabayon also provides a list of settings for applications and generic user interface elements that have built-in lock down support, including OpenOffice.org, and the GNOME panel. For example, the panel can be set up to allow only specific applets to be added to it and prevent changing its location or size on the screen. Likewise, the Save menu items can be disabled across all applications that use it, preventing users from saving documents.

The profiles are transferable to other computers. They reside in `/etc/desktop-profiles/`, and each profile is saved in a separate ZIP file.

4.6.1 Creating a Profile

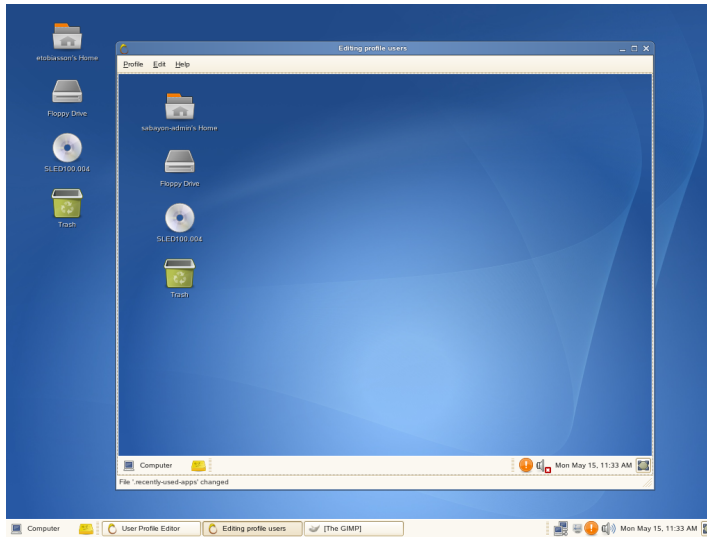
Profiles are saved in ZIP files located in `/etc/desktop-profiles`. Each profile you save is stored in a separate ZIP file as *name-of-the-profile.zip*. You can copy or move profiles to other computers.

- 1 Click *Computer > More Applications > System > User Profile Editor*.
- 2 If you are not logged in as `root`, type the `root` password, then click *Continue*.



- 3 Click *Add*.
- 4 Specify a name for the profile, then click *Add*.
- 5 Select the profile, then click *Edit*.

A new desktop session opens in an Xnest window.



- 6 In the Xnest window, make the changes to the settings that you want.

Each setting you change appears in the Xnest window.

You can choose to make each setting mandatory (click *Edit > Enforce Mandatory*), to ignore a setting (click *Edit > Changes > Ignore*), or make a setting default (do not selecting either *Ignore* or *Mandatory*).

- 7 To lock settings for users, click *Edit > Lockdown* in the Xnest window.

You can choose from the following options:

General: Lets you disable the command line, printing, print setup, and the save to disk feature.

Panel: Lets you lock down the panels, disable force quit, disable lock screen, disable log out, and disable any of the applets in the *Disabled Applets* list.

OpenOffice.org: Lets you define the macro security level for OpenOffice.org documents, load and save options, and user interface options.

Epiphany Web Browser: Lets you hide the menu bar, make the window full screen, and disable quit, arbitrary URLs, bookmark and toolbar editing, and unsafe protocols.

- 8 To save the profile, click *Profile > Save*.

The profile is saved in `/etc/desktop-profiles`.

- 9 Click *Profile > Quit* to close the Xnest window, then click *Close* to exit Sabayon.

4.6.2 Applying a Profile

You can apply a profile to individual users or to all users on a workstation.

- 1 Click *Computer > More Applications > System > Desktop Profile Editor*.
- 2 If you are not logged in as `root`, type the `root` password, then click *Continue*.
- 3 Select the profile you want to apply, then click *Users*.



- 4 Select the users you want to use this profile.

To apply this profile to all users on this workstation, click *Use this profile for all users*.

- 5 Click *Close*.

4.7 Adding Document Templates

To add document templates for users, fill in the `Templates` directory in a user's home directory. You can do this manually for each user by copying the files into `~/Templates`, or system-wide by adding a `Templates` directory with documents to `/etc/skel` before the user is created.

A user creates a new document from a template by right-clicking the desktop and selecting *Create Document*.

4.8 Desktop Lock Down Features

Sometimes it is wanted to remove or disable desktop features or user access to the underlying operating system. GNOME offers so-called lock down features to change the desktop accordingly. Technically you set GConf keys to implement those changes.

For example, if you open `gconf-editor`, you can see lock down keys for the main menu in

`/desktop/gnome/applications/main-menu/lock-down/application_browser_link_visible`.

There find also descriptions for all the keys. Other lock down keys are:

`/desktop/gnome/lockdown/disable_command_line`

if set then terminals are not shown in the main menu and the AppBrowser

`/apps/panel/global/disable_log_out`

`/apps/panel/global/disable_lock_screen`

if set main menu does not show these items

Find Firefox lock down keys in `/apps/firefox/lockdown`.

For more information, see the “Desktop Administrators' Guide to GNOME Lockdown and Preconfiguration” by Sayamindu Dasgupta: <http://library.gnome.org/admin/deployment-guide/>.

4.9 For More Information

For more information, see <http://library.gnome.org/admin/>.

Managing Software with Command Line Tools

This chapter describes Zypper and RPM, two command line tools for managing software.

5.1 Using Zypper

Zypper is a command line tool for installing and updating packages. zypper's syntax is similar to that of `rug`. In contrast to `rug`, zypper does not require the `zmd` daemon to run behind the scenes. For more information about `rug` compatibility, see `man zypper`, section “COMPATIBILITY WITH RUG”. It is especially useful to accomplish remote software management tasks or to manage software from shell scripts.

zypper has a help overview built in:

```
zypper help
```

5.1.1 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, run the command without asking anything, decide on your own:

```
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, apply all needed patches to the system without asking for confirming any licenses—all of them were read in advance:

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

Some of the commands require one or more arguments:

```
zypper install mplayer
```

Some of the options also require an argument. The following means, list all known patterns:

```
zypper search -t pattern
```

You can combine all of the above. For example, the following means, install `mplayer` and `amarok` packages using the `factory` repository only and be verbose:

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

5.1.2 Installing and Removing Software with Zypper

To install a package from registered repositories, use

```
zypper install package_name
```

To install a specific version of a package, use

```
zypper install package_name=version
```

zypper also supports wild cards. For example, to install all packages starting with *package_name* use

```
zypper install package_name*
```

You can also install a local or remote RPM directly—Zypper will also install all packages *package_name* depends on automatically:

```
zypper install http://www.example.com/package_name.rpm
```

To remove an installed package, use

```
zypper remove package_name
```

To install and remove packages in one go use the `+/-` or `~/!` modifiers:

```
zypper install emacs -vim
```

Or:

```
zypper remove emacs +vim
```

Or, if you choose to use `-` with the first package you specify, you must write `--` before it to prevent its interpretation as a command option:

```
zypper install -- -vim emacs
```

WARNING: Do not Remove Packages Mandatory for the System

Do not remove packages such as `glibc`, `zypper`, `kernel`, or similar. These packages are mandatory for the system and if missing, the system may stop working.

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

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

This option allows using `zypper` in scripts and cron jobs.

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

```
zypper source-install package_name
```

With this command, you will also install the build dependencies of the specified package.

If you do not want this, add the switch `--no-build-deps` as follows:

```
zypper source-install --no-build-deps package_name
```

Of course, this will only work, if you have the repository with the source packages added to your repository list. Enter `zypper search -t srcpackage` to get a

list of source packages available in your repositories. For more information about adding repositories, see [Section 5.1.4, “Managing Repositories”](#) (page 37).

If an error occurs during installation, or anytime you feel the need, verify whether all dependencies are still fulfilled:

```
zypper verify
```

5.1.3 Updating Software with Zypper

There are two different ways to update software using zypper. To integrate all officially released patches into your system, just run

```
zypper patch
```

In this case, all patches available in your repositories are checked for relevance, and installed if necessary. After registering your SUSE Linux Enterprise installation, an official update repository containing such patches will be added to your system. The above command is all you need to enter to apply them when needed.

If a repository just contains 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, use the update command with arguments:

```
zypper update package_name
```

Or the installation command:

```
zypper install package_name
```

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

```
zypper list-updates
```

Similarly, to list all needed patches, use:

```
zypper list-patches
```


5.1.4 Managing Repositories

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

```
zypper repos
```

The result will look similar to the following output:

#	Alias	Enabled	Refresh	Name
1	SUSE-Linux-Enterprise-Server 11-0	Yes	No	SUSE-Linux-Enterprise-Server 11-0
2	SLES-11-Updates	Yes	Yes	SLES 11 Online Updates
3	broadcomdrv	Yes	No	Broadcom Drivers

When specifying repositories in various commands, an alias, URI, or repository number from the `zypper repos` command output can be used. Note however that the numbers can change after modifying the list of repositories. The alias will never change by itself.

If you want to remove a repository from the list, use the command `zypper removerepo` together with the alias or number of the repository you want to delete. To remove the `Broadcom Drivers` from the example, use the following command:

```
zypper removerepo 3
```

To add a repository, run

```
zypper addrepo URI Alias
```

URI can either be an Internet repository, a directory, or a CD or DVD. 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.

To make working with repositories more convenient, use short and easy to remember aliases. A repository alias can be changed using the `renamerepo` command. For example, to rename the lengthy `SUSE-Linux-Enterprise-Server 11-0` from the example to the short and handy label `main`, enter:

```
zypper renamerepo 1 main
```

5.1.5 Querying

Various querying commands such as `search`, `info`, or `what-provides` are available.

`search` works on package names or, optionally, on package summaries and descriptions, and displays status (S) information in the first column of the list of found packages.

`info` with a package name as an arguments displays detailed information about a package.

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.

For more query commands and detailed usage information, see the `zypper` manpage (`man zypper`).

5.1.6 For More Information

For more information about managing software from the command line, enter `zypper help` or `zypper help command` or see the `zypper(8)` manpage.

5.2 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 `kdelibs3-devel`.

5.2.1 Verifying Package Authenticity

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

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

The command `rpm --checksig package-1.2.3.rpm` can be used to verify the signature of an RPM package to determine whether it really originates from SUSE or from another trustworthy facility. This is especially recommended for update packages from the Internet. The SUSE public package signature key normally resides in `/root/.gnupg/`. The key is additionally located in the directory `/usr/lib/rpm/gnupg/` to enable normal users to verify the signature of RPM packages.

5.2.2 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 dependencies are fulfilled and there are no conflicts with other packages. With an error message, `rpm` requests

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

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

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

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

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

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

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

5.2.3 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 large amounts of data. However the SUSE RPM offers a feature enabling the installation of patches in packages.

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

Is the patch RPM suitable for my system?

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

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

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

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

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

Which files are replaced by the patch?

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

```
rpm -qpPl pine-4.44-224.i586.patch.rpm
/etc/pine.conf
/etc/pine.conf.fixed
/usr/bin/pine
```

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

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

How can a patch RPM be installed in the system?

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

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

A list of all patches installed in the system can be displayed with the command

`rpm -qPa`. If only one patch is installed in a new system (as in this example), the list appears as follows:

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

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

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

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

5.2.4 Delta RPM Packages

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

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

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

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

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

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

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

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

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

5.2.5 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 5.1, “The Most Important RPM Query Options”](#) (page 43).

Table 5.1 *The Most Important RPM Query Options*

<code>-i</code>	Package information
<code>-l</code>	File list
<code>-f FILE</code>	Query the package that contains the file <i>FILE</i> (the full path must be specified with <i>FILE</i>)
<code>-s</code>	File list with status information (implies <code>-l</code>)

<code>-d</code>	List only documentation files (implies <code>-l</code>)
<code>-c</code>	List only configuration files (implies <code>-l</code>)
<code>--dump</code>	File list with complete details (to be used with <code>-l</code> , <code>-c</code> , or <code>-d</code>)
<code>--provides</code>	List features of the package that another package can request with <code>--requires</code>
<code>--requires, -R</code>	Capabilities the package requires
<code>--scripts</code>	Installation scripts (preinstall, postinstall, uninstall)

For example, the command `rpm -q -i wget` displays the information shown in [Example 5.1, “rpm -q -i wget”](#) (page 44).

Example 5.1 `rpm -q -i wget`

```

Name           : wget                               Relocations: (not relocatable)
Version        : 1.9.1                             Vendor: SUSE LINUX AG,
Nuernberg, Germany
Release       : 50                                  Build Date: Sat 02 Oct 2004
03:49:13 AM CEST
Install date: Mon 11 Oct 2004 10:24:56 AM CEST      Build Host: f53.suse.de
Group          : Productivity/Networking/Web/Utilities Source RPM:
wget-1.9.1-50.src.rpm
Size           : 1637514                             License: GPL
Signature      : DSA/SHA1, Sat 02 Oct 2004 03:59:56 AM CEST, Key ID
a84edae89c800aca
Packager       : http://www.suse.de/feedback
URL            : http://wget.sunsite.dk/
Summary        : A tool for mirroring FTP and HTTP servers
Description    :
Wget enables you to retrieve WWW documents or FTP files from a server.
This can be done in script files or via the command line.
[...]
```

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

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

results in:


```
rpm-4.1.1-191
wget-1.9.1-50
```

If only part of the filename is known, use a shell script as shown in [Example 5.2, “Script to Search for Packages”](#) (page 45). Pass the partial filename to the script shown as a parameter when running it.

Example 5.2 *Script to Search for Packages*

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

The command `rpm -q --changelog rpm` displays a detailed list of change information about a specific package, sorted by date. This example shows information about the package `rpm`.

With the help of the installed RPM database, verification checks can be made. Initiate these with `-V`, `-y`, or `--verify`. With this option, `rpm` shows all files in a package that have been changed since installation. `rpm` uses eight character symbols to give some hints about the following changes:

Table 5.2 *RPM Verify Options*

5	MD5 check sum
S	File size
L	Symbolic link
T	Modification time
D	Major and minor device numbers
U	Owner
G	Group

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

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

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

5.2.6 Installing and Compiling Source Packages

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

TIP

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

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

SOURCES

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

SPECS

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

BUILD

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

RPMS

where the completed binary packages are stored

SRPMS

here are the source RPMs

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

WARNING

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

The following example uses the `wget.src.rpm` package. After installing the package with YaST, you should have files similar to the following listing:

```
/usr/src/packages/SOURCES/nops_doc.diff
/usr/src/packages/SOURCES/toplev_destdir.diff
/usr/src/packages/SOURCES/wget-1.9.1+ipvmisc.patch
/usr/src/packages/SOURCES/wget-1.9.1-brovertime.patch
/usr/src/packages/SOURCES/wget-1.9.1-passive_ftp.diff
/usr/src/packages/SOURCES/wget-LFS-20040909.tar.bz2
/usr/src/packages/SOURCES/wget-wrong_charset.patch
/usr/src/packages/SPECS/wget.spec
```

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

`-bp`

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

`-bc`

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

`-bi`

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

`-bb`

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

`-ba`

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

`--short-circuit`

Skip some steps.

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

5.2.7 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.

5.2.8 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`.

KDE offers the `kpackage` tool as a front-end for `rpm`. A full-featured package manager is available as a YaST module (see Chapter 6, *Installing or Removing Software* (↑Deployment Guide)).

Accessing Remote Desktops with Nomad

Nomad (Novell Open Mobile Agile Desktop) ships with SUSE® Linux Enterprise Desktop and allows you to run desktop sessions detached from any graphics hardware. It consists of the following core components:

Proxy X Server

Supports modern X extensions like Composite, XVideo, and RANDR.

Session Manager

Responsible for spawning and keeping track of desktop sessions that can be accessed remotely.

Connection Handler

Uses the Remote Desktop Protocol (RDP) as a transport and security layer. RDP is a multi-channel protocol that allows a user to connect to a computer running Microsoft Terminal Services. However, when supported by the client software, the connection handler uses a virtual X11 channel (rdpx11) that transfers unfiltered X11 traffic to the local X server displaying the desktop. The connection handler can always fall back to plain RDP commands if necessary, which means that remote desktops can be accessed from any existing RDP client.

Client Program

A special RDP client is provided for SUSE Linux Enterprise Desktop that implements Nomad-specific extensions for X11 protocol forwarding and the ability to composite remote desktops locally when appropriate compositing manager plugins are loaded.

Compositing Manager Extensions

Compositing allows for advanced visual effects of application windows, such as transparency, fading, scaling, contorting, shuffling, and redirecting.

Nomad lets you remotely access desktops from different physical locations, for example, you can access the same session from home or from work. After an interruption of your work session, just move to another terminal and resume work there. It is also possible to copy the currently running environment to a mobile device like a laptop. With Nomad you can also share desktops for collaboration or training purposes, allowing remote control and administration.

6.1 Nomad Prerequisites

In order to use Nomad the `rdesktop` package needs to be installed on your local machine. Additionally, the following packages can be installed:

- `compiz`
- `compiz-plugins-dmx`
- `compiz-fusion-plugins-main`
- `libcompizconfig`
- `python-compizconfig`
- `compiz-manager`
- `simple-ccsm`
- `tsclient`

On the remote machine supplying the desktop, the `xrdp` package needs to be installed, containing an open source remote desktop protocol (RDP) server.

Additionally, the following packages can be installed:

- `compiz`

- `compiz-plugins-dmx`
- `compiz-fusion-plugins-main`
- `libcompizconfig`
- `python-compizconfig`
- `compiz-manager`
- `simple-ccsm`

6.2 Installation and Set-Up

The local machine acting as host does not need any special configuration. As soon as the `rdesktop` package is installed, you can use the `rdesktop` command line tool to connect to the remote machine that provides the desktop. If you prefer a graphical user interface, additionally install the `tsclient` package. `tsclient` (Terminal Server Client) is a GNOME front-end for `rdesktop` and other remote desktop tools, supporting also Xnest and VNC clients (`vncviewer`). For improved performance and desktop effects, install the additional `compiz` packages.

However, you need to prepare the remote machine providing the desktop as follows:

- 1 Install the `xrdp` package. This will automatically add the `xrpd` server to runlevel 5. To start or stop the service manually, run `/etc/init.d/xrdp start` or `/etc/init.d/xrdp stop` as root.
- 2 Configure the firewall to allow connections to port 3389 as this port is used for RDP connections. Start YaST and select *Security and Users > Firewall*. Click *Allowed Services* and select the zone for which to allow the service. Click *Advanced* and enter 3389 as *TCP Port*. Confirm your settings in YaST.
- 3 If you want to use 3D desktop effects, install the additional `compiz` packages. This will improve performance significantly when using a client with support for virtual channels. By enabling desktop effects on both the local and remote desktop, the local compositing manager will be able to apply effects to the elements coming from the remote desktop.

NOTE: Desktop Effects

If you intend to use desktop effects on the remote desktop, make sure the `compiz-plugins-dmx` package is installed on both systems: the system that provides the remote desktop as well as local system accessing the remote desktop.

6.3 Using Nomad

As soon as `xrpd` is running and port 3389 is open on the remote machine, you can connect to the remote host with your RDP client. To connect, either use the `rdesktop` command line tool or the `tsclient` providing a graphical user interface.

6.3.1 Connecting to the Server Using `rdesktop`

To establish a connection with compressed mode for user `tux`, run the following command from a shell:

```
rdesktop -u tux -z server
```

with `server` being the hostname or IP address of the remote machine.

This starts a login screen for the specified user where he can log in to the remote desktop. Desktop sessions via `xrpd` are independent and do not conflict with regular display managers like GDM or KDM.

You can set a number of options when establishing the connection. For example, you can use full screen mode, choose a certain keyboard layout or adjust the geometry. Learn more about the available `rdesktop` options with `rdesktop --help`.

6.4 Troubleshooting

If you have difficulties establishing a connection, proceed according to the following list.

Is the xrdp Server Up and Running on the Remote Machine?

1. Check if the `xrdp` package is installed on the remote machine providing the desktop.
2. Check if the `xrdp` service is running.
3. If not, start or restart it manually by executing the following command as `root`:
`/etc/init.d/xrdp start` or `/etc/init.d/xrdp restart`.

Two processes should be running after starting the `xrdp` service: `xrdp` and `xrdp-sesman`. If one of them fails to start for some reason, starting these processes manually in the foreground will most likely tell you what is wrong.

4. To start the processes manually, become `root` and run
`/usr/sbin/xrdp-sesman -n` and `/usr/sbin/xrdp -nodaemon`.
5. Also check the `xrdp-sesman` output in `/var/log/xrdp-sesman.log` and the `xrdp` output in `/var/log/messages` for more information.

6.5 For More Information

For more information about Nomad, refer to <http://en.opensuse.org/Nomad>.

Bash and Bash Scripts

Today, 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 execution of automatic tasks. Shells are a good addition to GUIs and this chapter gives you an overview about some aspects of shells, in this case the Bash.

7.1 What is “The Shell”?

Traditionally, *the* shell is Bash (Bourne again Shell). When this chapter speaks about “the shell” it means the Bash. Actually more shells are available than Bash, with different features and characteristics. If you need further information of other shells, search for *shell* in YaST.

7.1.1 Knowing The Bash Configuration Files

A shell can be invoked:

1. as an 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. as an “ordinary”, interactive shell. This is normally the case when starting xterm, konsole, or similar tools.

3. as an non-interactive shell. This is used when invoking a shell script at the commandline.

Depending on which type of shell you use, different configuration files are being read. The following tables shows the login and non-login shell configuration files.

Table 7.1 *Bash Configuration Files for Login Shells*

File	Description
/etc/profile	Do not modify this file, otherwise your modifications can be destroyed during your next update!
/etc/profile.local	use this file if you extent /etc/profile
/etc/profile.d/	contains system-wide configuration files for specific programs
~/.profile	insert user specific configuration for login shells here

Table 7.2 *Bash Configuration Files for Non-Login Shells*

/etc/bash.bashrc	Do not modify this file, otherwise your modifications can be destroyed during your next update!
/etc/bash.bashrc.local	use this file, to insert your system-wide modifications for Bash only
~/bashrc	insert user specific configuration here

Additionally, the Bash uses some more files:

Table 7.3 *Special Files for Bash*

File	Description
~/.bash_history	contains a list of all commands you have been typing

File	Description
<code>~/.bash_logout</code>	used when logging out

7.1.2 The Directory Structure

The following table provides a short overview of the most important higher-level directories you find on a Linux system. Find more detailed information about the directories and important subdirectories in the following list.

Table 7.4 *Overview of a Standard Directory Tree*

Directory	Contents
<code>/</code>	Root directory—the starting point of the directory tree.
<code>/bin</code>	Essential binary files, such as commands that are needed by both the system administrator and normal users. Usually also contains the shells, such as Bash.
<code>/boot</code>	Static files of the boot loader.
<code>/dev</code>	Files needed to access host-specific devices.
<code>/etc</code>	Host-specific system configuration files.
<code>/home</code>	Holds the home directories of all users who have an account on the system. Only <code>root</code> 's home directory is not located in <code>/home</code> but in <code>/root</code> .
<code>/lib</code>	Essential shared libraries and kernel modules.
<code>/media</code>	Mount points for removable media.
<code>/mnt</code>	Mount point for temporarily mounting a file system.
<code>/opt</code>	Add-on application software packages.

Directory	Contents
<code>/root</code>	Home directory for the superuser <code>root</code> .
<code>/sbin</code>	Essential system binaries.
<code>/srv</code>	Data for services provided by the system.
<code>/tmp</code>	Temporary files.
<code>/usr</code>	Secondary hierarchy with read-only data.
<code>/var</code>	Variable data such as log files.
<code>/windows</code>	Only available if you have both Microsoft Windows* and Linux installed on your system. Contains the Windows data.

The following list provides more detailed information and gives some examples 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 SUSE Linux Enterprise Desktop.

`/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 form of hidden files and directories. KDE users find the personal configuration data for their desktop in `.kde` or `.kde4` respectively, 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 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 additional software. Optional software and larger add-on program packages can be found there. KDE3 is located there, whereas KDE4 and GNOME have moved to `/usr` now.

`/root`

Home directory for the `root` user. Personal data of `root` is located here.

`/sbin`

As the `s` indicates, this directory holds utilities for the superuser. `/sbin` contains 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.

`/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 various hosts compliant to 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 programs reserved for the system administrator, such as repair functions.

`/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 relating 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 example, the log files of your system are in `/var/log/messages` (only accessible for `root`).

`/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 an NTFS file system, however, you can only read your Windows files from Linux, but not modify them. .

7.2 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 7.1 *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 it does.
- ❸ The third line uses the builtin command `echo` to print the respective 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 yourself.
2. You can save the script wherever you want. However, it is a good idea to save it in a directory where the shell searches for it. The search path in a shell is determined by the environment variable `PATH`. For example, save it in the directory `~/bin/` under 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 with either `~/bin/hello.sh` or `hello.sh`. The first call uses an absolute path whereas the second one searches for the command in each directory given by the `PATH` environment variable.

7.3 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 into a variable:

```
read a < foo
```

Command1 | Command2

Redirects the output of the left command as input for the right command.

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`:

```
find / -name "foo*" 2>/dev/null
```

7.4 Using Aliases

An alias is shortcut definition of one or more commands. 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`.

7.5 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. If you need a special variable, insert the name of your variable as an argument:

```
printenv PATH
```

A variable can also be viewed with `echo`:

```
echo $PATH
```

This prints the `PATH` variable. 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 an variable, use `unset`:

```
unset NAME
```

The following table contains some common environment variables which can be used in you shell scripts:

Table 7.5 *Useful Environment Variables*

HOME	the home directory of the current user
HOST	the current host name
LANG	when a tool is localized, it uses the language from this environment variable. English can also be set to C.
PATH	the search path of the shell, a list of directories separated by colon.
PS1	specifies the normal prompt printed before each command
PS2	specifies the secondary prompt printed when you execute a multi-line command

PWD	current working directory
USER	the current user

7.5.1 Using Argument Variables

For example, if you have a script `foo.sh` you can execute it like this:

```
foo.sh "Tux Penguin" 2000
```

To access all these arguments which are passed to your script, you need positional parameters. These are `$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" "" ""
```

7.5.2 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
```

7.6 Grouping And Combining Commands

The shells allows 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"; }
```

You can call this function like this:

```
hello Tux
```

which prints:

```
Hello Tux
```

7.7 Working with Common Flow Constructs

To control the flow of your script, a shell has `while`, `if`, `for`, and `case` constructs.

7.7.1 The `if` Control Command

The `if` is used to check expressions. For example, the following code tests, if 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 expressions checks if the file `foo.txt` exists:

```
if test -e /tmp/foo.txt
then
```

```
    echo "Found foo.txt"
fi
```

Find more expressions to discover at <http://www.cyberciti.biz/nixcraft/linux/docs/uniqlinuxfeatures/lsst/ch03sec02.html>.

7.7.2 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
```

7.8 For More Information

Important information about Bash is provided in the man pages `man sh`. 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>—Sh - the Bourne Shell

Part II. System

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

8

SUSE® Linux Enterprise Desktop is available for 64-bit platforms. This does not necessarily mean that all the applications included have already been ported to 64-bit platforms. SUSE Linux Enterprise Desktop supports the use of 32-bit applications in a 64-bit system environment. This chapter offers a brief overview of how this support is implemented on 64-bit SUSE Linux Enterprise Desktop 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.

SUSE Linux Enterprise Desktop 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.

8.1 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 from this rule is PAM (pluggable authentication modules). SUSE Linux Enterprise Desktop uses PAM in the authentication process as a layer that mediates between user and application. On a 64-Bit operating system that also runs 32-Bit applications it is necessary to always install both versions of a PAM module.

To be executed correctly, every application requires a range of libraries. Unfortunately, the names for the 32-bit and 64-bit versions of these libraries are identical. They must be differentiated from each other in another way.

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

All 64-bit libraries and object files are located in directories called `lib64`. The 64-bit object files 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 whose data content does not depend on the word size are not moved. This scheme conforms to LSB (Linux Standards Base) and FHS (File System Hierarchy Standard).

8.2 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 SUSE Linux Enterprise Desktop environment is designed according to this principle. In the case of manually updated libraries, resolve these issues yourself.

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

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

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

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

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

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

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  
  
make  
make install
```

8.4 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

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

Booting a Linux system involves various different components. The hardware itself is initialized by the BIOS, which starts the kernel by means of a boot loader. After this point, the boot process with `init` and the runlevels is completely controlled by the operating system. The runlevel concept enables you to maintain setups for everyday usage as well as to perform maintenance tasks on the system.

9.1 The Linux Boot Process

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

1. **BIOS** After the computer has been turned on, the BIOS initializes the screen and keyboard and tests the main memory. Up to this stage, the machine does not access any mass storage media. Subsequently, the information about the current date, time, and the most important peripherals are loaded from the CMOS values. When the first hard disk and its geometry are recognized, the system control passes from the BIOS to the boot loader.
2. **Boot Loader** The first physical 512-byte data sector of the first hard disk is loaded into the main memory and the *boot loader* that resides at the beginning of this sector takes over. The commands executed by the boot loader determine the remaining part of the boot process. Therefore, the first 512 bytes on the first hard disk are referred to as the *Master Boot Record* (MBR). The boot loader then passes control to the actual operating system, in this case, the Linux kernel. More

information about GRUB, the Linux boot loader, can be found in [Chapter 10, *The Boot Loader GRUB*](#) (page 93).

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 9.1.1, “initramfs”](#) (page 78).
4. **init on initramfs** This program performs all actions needed to mount the proper root file system, like providing kernel functionality for the needed file system and device drivers for mass storage controllers with `udev`. After the root file system has been found, it is checked for errors and mounted. If this has been successful, the initramfs is cleaned and the `init` program on the root file system is executed. For more information about `init`, refer to [Section 9.1.2, “init on initramfs”](#) (page 79). Find more information about `udev` in [Chapter 13, *Dynamic Kernel Device Management with udev*](#) (page 145).
5. **init** `init` handles the actual booting of the system through several different levels providing different functionality. `init` is described in [Section 9.2, “The init Process”](#) (page 81).

9.1.1 initramfs

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

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

changing the root file system, it is necessary to regenerate the devices. This is done by `boot.udev` with the command `udevtrigger`.

If you need to change hardware (e.g. hard disks) in an installed system and this hardware requires different drivers to be present in the kernel at boot time, you must update `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 SUSE® Linux Enterprise Desktop, 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.

9.1.2 `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 12, *Advanced Disk Setup* (↑Deployment Guide).

Managing Network Configuration

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

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

Finding the Installation Medium

As you start the installation process, your machine loads an installation kernel and a special `initrd` with the YaST installer from the installation medium. The YaST installer, which is run in a RAM file system, needs to have information about the location of the installation medium to access it and install the operating system.

Initiating Hardware Recognition and Loading Appropriate Kernel Modules

As mentioned in [Section 9.1.1, “initramfs”](#) (page 78), the boot process starts with a minimum set of drivers that can be used with most hardware configurations. `init` starts an initial hardware scanning process that determines the set of drivers suitable for your hardware configuration. The names of the modules needed for the boot process are written to `INITRD_MODULES` in `/etc/sysconfig/kernel`. These names are used to generate a custom `initramfs` that is needed to boot the system. If the modules are not needed for boot but for coldplug, the modules are written to `/etc/sysconfig/hardware/hwconfig-*`. All devices that are described with configuration files in this directory are initialized in the boot process.

Loading the Installation System or Rescue System

As soon as the hardware has been properly recognized, the appropriate drivers have been loaded, and udev has created the device special files, init starts the installation system, which contains the actual YaST installer, or the rescue system.

Starting YaST

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

9.2 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 9.2.1, “Runlevels”](#) (page 81)). 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 9.2.2, “Init Scripts”](#) (page 84)).

The entire process of starting the system and shutting it down is maintained by `init`. From this point of view, the kernel can be considered a background process whose task is to maintain all other processes and adjust CPU time and hardware access according to requests from other programs.

9.2.1 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 9.1, “Available Runlevels”](#) (page 82). 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 a the single number 3 to the boot prompt.

Table 9.1 *Available Runlevels*

Runlevel	Description
0	System halt
S or 1	Single user mode
2	Local multiuser mode without remote network (NFS, etc.)
3	Full multiuser mode with network
4	<i>User Defined</i> , this is not used unless the administrator configures this runlevel.
5	Full multiuser mode with network and X display manager—KDM, GDM, or XDM
6	System reboot

IMPORTANT: Avoid Runlevel 2 with a Partition Mounted via NFS

You should not use runlevel 2 if your system mounts a partition like `/usr` via NFS. The system might behave unexpectedly if program files or libraries are missing because the NFS service is not available in runlevel 2 (local multiuser mode without remote network).

To change runlevels while the system is running, enter `telinit` and the corresponding number as an argument. Only the system administrator is allowed to do this. The following list summarizes the most important commands in the runlevel area.

```
telinit 1 or shutdown now
```

The system changes to *single user mode*. This mode is used for system maintenance and administration tasks.

```
telinit 3
```

All essential programs and services (including network) are started and regular users are allowed to log in and work with the system without a graphical environment.

```
telinit 5
```

The graphical environment is enabled. Usually a display manager like XDM, GDM, or KDM is started. If autologin is enabled, the local user is logged in to the preselected window manager (GNOME or KDE or any other window manager).

```
telinit 0 or shutdown -h now
```

The system halts.

```
telinit 6 or shutdown -r now
```

The system halts then reboots.

Runlevel 5 is the default runlevel in all SUSE Linux Enterprise Desktop standard installations. Users are prompted for login with a graphical interface or the default user is logged in automatically. If the default runlevel is 3, the X Window System must be configured properly, as described in [Chapter 14, *The X Window System*](#) (page 159), before the runlevel can be switched to 5. If this is done, check whether the system works in the desired way by entering `telinit 5`. If everything turns out as expected, you can use YaST to set the default runlevel to 5.

WARNING: Errors in `/etc/inittab` May Result in a Faulty System Boot

If `/etc/inittab` is damaged, the system might not boot properly. Therefore, be extremely careful while editing `/etc/inittab`. Always let init reread `/etc/inittab` with the command `telinit q` before rebooting the machine.

Generally, two things happen when you change runlevels. First, stop scripts of the current runlevel are launched, closing down some programs essential for the current runlevel. Then start scripts of the new runlevel are started. Here, in most cases, a number of programs are started. For example, the following occurs when changing from runlevel 3 to 5:

1. The administrator (`root`) requests init to change to a different runlevel by entering `telinit 5`.

2. `init` checks the current runlevel (`runlevel`) and determines it should start `/etc/init.d/rc` with the new runlevel as a parameter.
3. Now `rc` calls the stop scripts of the current runlevel for which there is no start script in the new runlevel. In this example, these are all the scripts that reside in `/etc/init.d/rc3.d` (old runlevel was 3) and start with a `K`. The number following `K` specifies the order to run the scripts with the `stop` parameter, because there are some dependencies to consider.
4. The last things to start are the start scripts of the new runlevel. In this example, these are in `/etc/init.d/rc5.d` and begin with an `S`. Again, the number that follows the `S` determines the sequence in which the scripts are started.

When changing into the same runlevel as the current runlevel, `init` only checks `/etc/inittab` for changes and starts the appropriate steps, for example, for starting a `getty` on another interface. The same functionality may be achieved with the command `telinit q`.

9.2.2 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 clarity reasons and avoids duplicate scripts if they are used in several runlevels. Because every script can be executed as both a start and a stop script, these scripts must understand the parameters `start` and `stop`. The scripts also understand the `restart`, `reload`,

`force-reload`, and `status` options. These different options are explained in [Table 9.2, “Possible init Script Options”](#) (page 85). Scripts that are run directly by `init` do not have these links. They are run independently from the runlevel when needed.

Table 9.2 *Possible init Script Options*

Option	Description
<code>start</code>	Start service.
<code>stop</code>	Stop service.
<code>restart</code>	If the service is running, stop it then restart it. If it is not running, start it.
<code>reload</code>	Reload the configuration without stopping and restarting the service.
<code>force-reload</code>	Reload the configuration if the service supports this. Otherwise, do the same as if <code>restart</code> had been given.
<code>status</code>	Show the current status of service.

Links in each runlevel-specific subdirectory make it possible to associate scripts with different runlevels. When installing or uninstalling packages, these links are added and removed with the help of the program `insserv` (or using `/usr/lib/lsb/install_initd`, which is a script calling this program). See the `insserv(8)` man page for details.

All of these settings may also be changed with the help of the YaST module. If you need to check the status on the command line, use the tool `chkconfig`, described in the `chkconfig(8)` man page.

A short introduction to the boot and stop scripts launched first or last, respectively, follows as well as an explanation of the maintaining script.

`boot`

Executed while starting the system directly using `init`. It is independent of the chosen runlevel and is only executed once. Here, the `/proc` and `/dev/pts` file systems are mounted and `blogd` (boot logging daemon) is activated. If the system

is booted for the first time after an update or an installation, the initial system configuration is started.

The `blogd` daemon is a service started by `boot` and `rc` before any other one. It is stopped after the actions triggered by these scripts (running a number of subscripts, for example, making block special files available) are completed. `blogd` writes any screen output to the log file `/var/log/boot.msg`, but only if and when `/var` is mounted read-write. Otherwise, `blogd` buffers all screen data until `/var` becomes available. Get further information about `blogd` on the `blogd(8)` man page.

The `boot` script is also responsible for starting all the scripts in `/etc/init.d/boot.d` with a name that starts with `S`. There, the file systems are checked and loop devices are configured if needed. The system time is also set. If an error occurs while automatically checking and repairing the file system, the system administrator can intervene after first entering the root password. 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`, `chkconfig`, and `insserv`. Additionally consult the man pages of `startproc` and `killproc`.

WARNING: Faulty init Scripts May Halt Your System

Faulty init scripts may hang your machine. Edit such scripts with great care and, if possible, subject them to heavy testing in the multiuser environment. Find some useful information about init scripts in [Section 9.2.1, “Runlevels”](#) (page 81).

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 9.1, “A Minimal INIT INFO Block”](#) (page 87).

Example 9.1 *A Minimal INIT INFO Block*

```
### BEGIN INIT INFO
# Provides:          FOO
# Required-Start:    $syslog $remote_fs
# Required-Stop:     $syslog $remote_fs
# Default-Start:     3 5
# Default-Stop:      0 1 2 6
# Description:       Start FOO to allow XY and provide YZ
### END INIT INFO
```

In the first line of the `INFO` block, after `Provides :`, specify the name of the program or service controlled by this init script. In the `Required-Start :` and `Required-Stop :` lines, specify all services that need to be started or stopped before the service itself is started or stopped. This information is used later to generate the numbering of script names, as found in the runlevel directories. After `Default-Start :` and `Default-Stop :`, specify the runlevels in which the service should automatically be started or stopped. Finally, for `Description :`, provide a short description of the service in question.

To create the links from the runlevel directories (`/etc/init.d/rc?.d/`) to the corresponding scripts in `/etc/init.d/`, enter the command `insserv new-script-name`. The `insserv` program evaluates the `INIT INFO` header to create the necessary links for start and stop scripts in the runlevel directories (`/etc/init.d/rc?.d/`). The program also takes care of the correct start and stop order for each runlevel by including the necessary numbers in the names of these links. If you prefer

a graphical tool to create such links, use the runlevel editor provided by YaST, as described in [Section 9.2.3, “Configuring System Services \(Runlevel\) with YaST”](#) (page 88).

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.

9.2.3 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 SUSE Linux Enterprise Desktop system is runlevel 5 (full multiuser mode with network and X). A suitable alternative might be runlevel 3 (full multiuser mode with network).

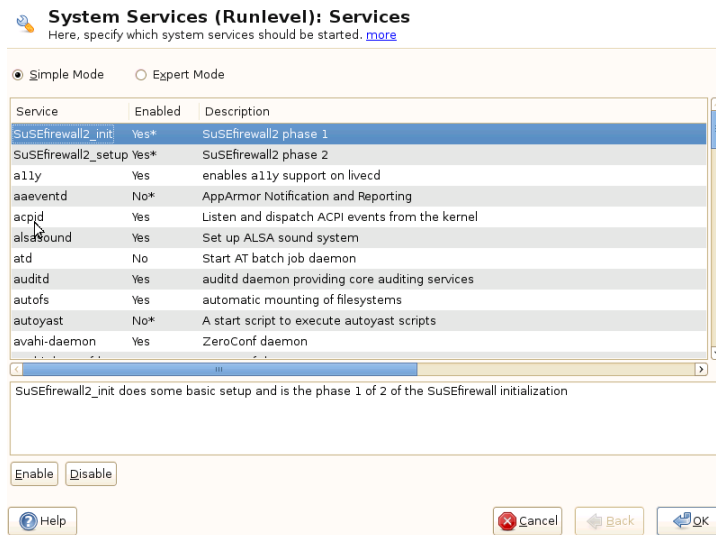
This YaST dialog allows the selection of one of the runlevels (as listed in [Table 9.1, “Available Runlevels”](#) (page 82)) as the new default. Additionally use the table in this window to enable or disable individual services and daemons. The table lists the services and daemons available, shows whether 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 9.1 *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 *Finish* saves the changed settings to disk.

9.3 System Configuration via `/etc/sysconfig`

The main configuration of SUSE Linux Enterprise Desktop 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.

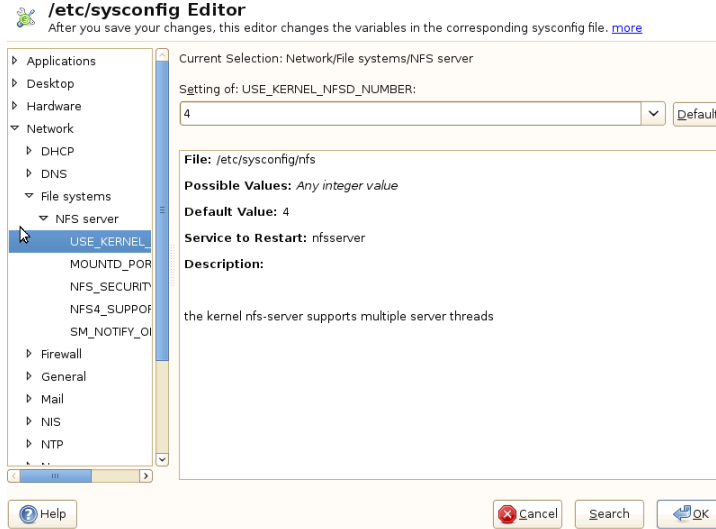
9.3.1 Changing the System Configuration Using the YaST `sysconfig` Editor

The YaST `sysconfig` editor provides an easy-to-use front-end to system configuration. Without any knowledge of the actual location of the configuration variable you need to change, you can just use the built-in search function of this module, change the value of the configuration variable as needed, and let YaST take care of applying these changes, updating configurations that depend on the values set in `sysconfig` and restarting services.

WARNING: Modifying `/etc/sysconfig/*` Files Can Damage Your Installation

Do not modify the `/etc/sysconfig` files if you lack previous experience and knowledge. It could do considerable damage to your system. The files in `/etc/sysconfig` include a short comment for each variable to explain what effect they actually have.

Figure 9.2 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.

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

NOTE: No GRUB on machines using UEFI

GRUB will routinely be installed on machines equipped with a traditional BIOS and on UEFI (Unified Extensible Firmware Interface) machines using a Compatibility Support Module (CSM). On UEFI machines without enabled CSM, `eLILo` will automatically be installed (provided DVD1 booted successfully). Refer to the `eLILo` documentation at `/usr/share/doc/packages/elilo/` on your system for details.

This chapter focuses on boot management and the configuration of the boot loader GRUB. The boot procedure as a whole is outlined in *Chapter 9, Booting and Configuring a Linux System* (page 77). A boot loader represents the interface between the machine (BIOS) and the operating system (SUSE Linux Enterprise Desktop). The configuration of the boot loader directly impacts the start of the operating system.

The following terms appear frequently in this chapter and might need some explanation:

Master Boot Record

The structure of the MBR is defined by an operating system-independent convention. The first 446 bytes are reserved for the program code. They typically hold

part of a boot loader program or an operating system selector. The next 64 bytes provide space for a partition table of up to four entries. 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 important basic data of the file system. In contrast, the boot sectors of Linux partitions are initially empty after setting up a file system other than XFS. Therefore, a Linux partition is not bootable by itself, even if it contains a kernel and a valid root file system. A boot sector with valid code for booting the system has the same magic number as the MBR in its last two bytes (AA55).

10.1 Booting with GRUB

GRUB (Grand Unified Bootloader) 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 three files that are described below:

`/boot/grub/menu.lst`

This file contains all information about partitions or operating systems that can be booted with GRUB. Without this information, the GRUB command line prompts the user for how to proceed (see [Section “Editing Menu Entries during the Boot Procedure”](#) (page 100) for details).

`/boot/grub/device.map`

This file translates device names from the GRUB and BIOS notation to Linux device names.

`/etc/grub.conf`

This file contains the commands, parameters, and options the GRUB shell needs for installing the boot loader correctly.

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

In GRUB, all boot parameters can be changed prior to booting. For example, errors made when editing the menu file can be corrected in this way. Boot commands can also be entered interactively at a kind of input prompt (see [Section “Editing Menu Entries during the Boot Procedure”](#) (page 100)). GRUB offers the possibility of determining the location of the kernel and the `initrd` prior to booting. In this way, you can even boot an installed operating system for which no entry exists in the boot loader configuration.

GRUB actually exists in two versions: as a boot loader and as a normal Linux program in `/usr/sbin/grub`. This program is referred to as the *GRUB shell*. It provides an emulation of GRUB in the installed system and can be used to install GRUB or test new settings before applying them. The functionality to install GRUB as the boot loader on a hard disk or floppy disk is integrated in GRUB in the form of the commands `install` and `setup`. This is available in the GRUB shell when Linux is loaded.

10.1.1 The GRUB Boot Menu

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

Every time the system is booted, GRUB loads the menu file from the file system. For this reason, GRUB does not need to be reinstalled after every change to the file. Use the YaST boot loader to modify the GRUB configuration as described in [Section 10.2, “Configuring the Boot Loader with YaST”](#) (page 103).

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 97). 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 98).

Naming Conventions for Hard Disks and Partitions

The naming conventions GRUB uses for hard disks and partitions differ from those used for normal Linux devices. It more closely resembles the simple disk enumeration the BIOS does and the syntax is similar to that used in some BSD derivatives. In GRUB, the numbering of the partitions starts with zero. This means that `(hd0, 0)` is the first partition of the first hard disk. On a common desktop machine with a hard disk connected as primary master, the corresponding Linux device name is `/dev/sda1`.

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

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

Being dependent on BIOS devices, GRUB does not distinguish between 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 10.1.2, “The File `device.map`”](#) (page 101).

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

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

An Example Menu File

The following example shows the structure of a GRUB menu file. The example installation has a Linux boot partition under `/dev/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 SUSE Linux Enterprise Desktop. The kernel (`vmlinuz`) is located in the first logical partition (the boot partition) of the first hard disk. Kernel parameters, such as the root partition and VGA mode, are appended here. The root partition is specified according to the Linux naming convention (`/dev/sda7/`), because this information is read by the kernel and has nothing to do with GRUB. The `initrd` is also located in the first logical partition of the first hard disk.
- 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 100).

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” (↑System Analysis and Tuning Guide) for a figure.

Editing menu entries facilitates the repair of a defective system that can no longer be booted, because the faulty configuration file of the boot loader can be circumvented by manually entering parameters. Manually entering parameters during the boot procedure is also useful for testing new settings without impairing the native system.

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

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

```
title linux
    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.

10.1.2 The File `device.map`

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

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

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

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
```

10.1.3 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.

10.1.4 Setting a Boot Password

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

IMPORTANT: Boot Password and Splash Screen

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

As the user `root`, proceed as follows to set a boot password:

- 1 At the root prompt, encrypt the password using `grub-md5-crypt`:

```
# grub-md5-crypt
Password: ****
Retype password: ****
Encrypted: $1$lS2dv/$JOYcdxIn7CJk9xShzzJVw/
```

- 2 Paste the encrypted string into the global section of the file `menu.lst`:

```
gfxmenu (hd0,4)/message
color white/blue black/light-gray
default 0
timeout 8
password --md5 $1$lS2dv/$JOYcdxIn7CJk9xShzzJVw/
```

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

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

```
title linux
    kernel (hd0,4)/vmlinuz root=/dev/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:

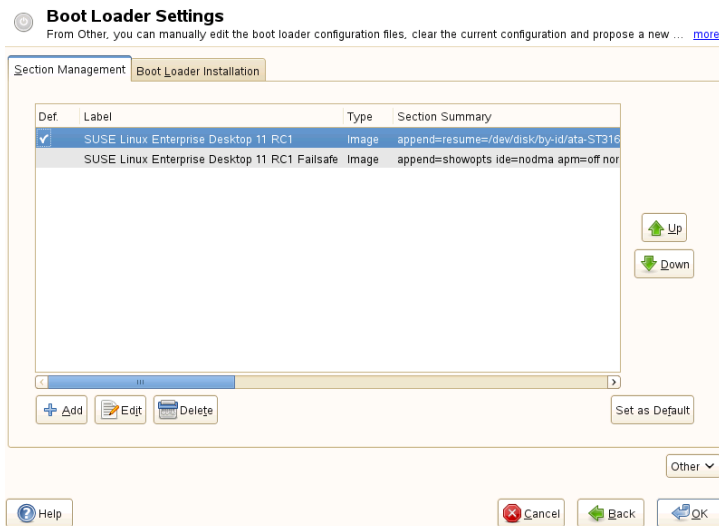
```
Error 32: Must be authenticated
```

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.

10.2 Configuring the Boot Loader with YaST

The easiest way to configure the boot loader in your SUSE Linux Enterprise Desktop system is to use the YaST module. In the YaST Control Center, select *System > Boot Loader*. As in **Figure 10.1**, “**Boot Loader Settings**” (page 103), this shows the current boot loader configuration of your system and allows you to make changes.

Figure 10.1 *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 10.1, “Booting with GRUB”](#) (page 94) first.

Use the *Boot Loader Installation* tab to view and change settings related to type, location, and advanced loader settings.

Access advanced configuration options from the drop-down menu that opens after you click on *Other*. The build-in editor lets you change the GRUB configuration files (see [Section 10.1, “Booting with GRUB”](#) (page 94) for details). You can also delete the existing configuration and *Start from Scratch* or let YaST *Propose a New Configuration*. It is also possible to write the configuration to disk or reread the configuration from the disk. To restore the original Master Boot Record (MBR) that was saved during the installation, choose *Restore MBR of Hard Disk*.

10.2.1 Adjusting the Default Boot Entry

To change the system that is booted by default, proceed as follows:

Procedure 10.1 *Setting the Default System*

- 1 Open the *Section Management* tab.
- 2 Select the desired entry from the list.
- 3 Click *Set as Default*.
- 4 Click *Finish* to activate these changes.

10.2.2 Modifying the Boot Loader Location

To modify the location of the boot loader, follow these steps:

Procedure 10.2 *Changing the Boot Loader Location*

- 1 Select the *Boot Loader Installation* tab and then choose one of the following options for *Boot Loader Location*:

Boot from Boot Partition

The boot sector of the `/boot` partition.

Boot from Extended Partition

This installs the boot loader in the extended partition container.

Boot from Master Boot Record

This installs the boot loader in the MBR of the first disk (according to the boot sequence preset in the BIOS).

Boot from Root Partition

This installs the boot loader in the boot sector of the `/` partition.

Custom Boot Partition

Use this option to specify the location of the boot loader manually.

- 2 Click *Finish* to apply your changes.

10.2.3 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 10.3 *Changing the Boot Loader Time-Out*

- 1 Open the *Boot Loader Installation* tab.
- 2 Click *Boot Loader Options*.
- 3 Change the value of *Time-Out in Seconds* by typing in a new value, clicking the appropriate arrow key with your mouse, or by using the arrow keys on the keyboard.
- 4 Click *OK*.
- 5 Click *Finish* to save the changes.

10.2.4 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 10.4 *Setting a Boot Loader Password*

- 1 Open the *Boot Loader Installation* tab.
- 2 Click *Boot Loader Options*.
- 3 Set your password in *Password for the Menu Interface*.
- 4 Click *OK*.
- 5 Click *Finish* to save the changes.

10.2.5 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 10.1.2, “The File device.map”](#) (page 101)). To do so, proceed as follows:

Procedure 10.5 *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* to save the changes.
- 5 Click *Finish* to save the changes.

10.2.6 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.

Debugging Flag

Sets GRUB in debug mode where it displays messages to show disk activity.

Write Generic Boot Code to MBR

Replaces the current MBR with generic, operating system independent code.

Hide Boot Menu

Hides the boot menu and boots the default entry.

Use Trusted GRUB

Starts the Trusted GRUB which supports trusted computing functionality.

Graphical Menu File

Path to the graphics file used when displaying the boot screen.

Serial Connection Parameters

If your machine is controlled via a serial console, you can specify which COM port to use at which speed. Also set *Terminal Definition* to “serial”. See `info grub` or <http://www.gnu.org/software/grub/manual/grub.html> for details.

Terminal Definition

If you are booting via serial console, enter “serial” here, otherwise leave blank. You also need to specify *Serial Connection Parameters* in this case.

10.2.7 Changing Boot Loader Type

Set the boot loader type in *Boot Loader Installation*. The default boot loader in SUSE Linux Enterprise Desktop is GRUB. To use LILO, proceed as follows:

Procedure 10.6 *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 SUSE Linux Enterprise Desktop.

Read Configuration Saved on Disk

Load your own `/etc/lilo.conf`. This action is not available during the installation of SUSE Linux Enterprise Desktop.

- 4** Click *OK* to save the changes
- 5** Click *Finish* in the main dialog to apply the changes.

During the conversion, the old GRUB configuration is saved to disk. To use it, simply change the boot loader type back to GRUB and choose *Restore Configuration Saved before Conversion*. This action is available only on an installed system.

NOTE: Custom Boot Loader

To use a boot loader other than GRUB or LILO, select *Do Not Install Any Boot Loader*. Read the documentation of your boot loader carefully before choosing this option.

10.3 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 on request.

To uninstall GRUB, start the YaST boot loader module (*System > Boot Loader*). Select *Other > Restore MBR of Hard Disk* and confirm with *Yes, Rewrite*.

10.4 Creating Boot CDs

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

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

Procedure 10.7 *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 Adjust the path entries in `boot/grub/menu.lst` to make them point to a CD-ROM device. Do this by replacing the device name of the hard disks, listed in the format `(hdx, y)`, in the pathnames with `(cd)`, the device name of the CD-ROM drive. 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 data file, but use the option for burning a CD image in your burning utility.

10.5 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 10, *The Boot Loader GRUB*** (page 93) provides more information about this. However, if you prefer the text mode, which was the default in earlier versions, set `vga=normal`.

Completely Disabling the SUSE Screen

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

TIP

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

10.6 Troubleshooting

This section lists some of the problems frequently encountered when booting with GRUB and a short description of possible solutions. Some of the problems are covered in articles in the Knowledge base at <http://support.novell.com/>. Use the search dialog to search for keywords like *GRUB*, *boot*, and *boot loader*.

GRUB and XFS

XFS leaves no room for `stage1` in the partition boot block. Therefore, do not specify an XFS partition as the location of the boot loader. This problem can be solved by creating a separate boot partition that is not formatted with XFS.

GRUB Reports GRUB Geom Error

GRUB checks the geometry of connected hard disks when the system is booted. Sometimes, the BIOS returns inconsistent information and GRUB reports a GRUB Geom Error. If this is the case, 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 IDE disk as `hd0` and the SCSI disk as `hd1`, although the boot sequence in the BIOS is reversed (SCSI *before* IDE).

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

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

Booting Windows from the Second Hard Disk

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

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

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

10.7 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 Technical Information Search at <http://www.novell.com/support> 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 is finished by a section about language and country-specific settings (I18N and L10N).

11.1 Information about Special Software Packages

The programs `bash`, `cron`, `logrotate`, `locate`, `ulimit`, and `free` are very important for system administrators and many users. Man pages and info pages are two useful sources of information about commands, but both are not always available. GNU Emacs is a popular and very configurable text editor.

11.1.1 The `bash` Package and `/etc/profile`

Bash is the default system shell. When used as a login shell, it reads several initialization files. Bash processes them in the order they appear in this list:

1. `/etc/profile`
2. `~/.profile`
3. `/etc/bash.bashrc`
4. `~/.bashrc`

Make custom settings in `~/.profile` or `~/.bashrc`. To ensure the correct processing of these files, it is necessary to copy the basic settings from `/etc/skel/.profile` or `/etc/skel/.bashrc` into the home directory of the user. It is recommended to copy the settings from `/etc/skel` after an update. Execute the following shell commands to prevent the loss of personal adjustments:

```
mv ~/.bashrc ~/.bashrc.old
cp /etc/skel/.bashrc ~/.bashrc
mv ~/.profile ~/.profile.old
cp /etc/skel/.profile ~/.profile
```

Then copy personal adjustments back from the `*.old` files.

11.1.2 The cron Package

If you want to run commands regularly and automatically in the background at predefined times, cron is the tool to use. cron is driven by specially formatted time tables. Some of them come with the system and users can write their own tables if needed.

The cron tables are located in `/var/spool/cron/tabs`. `/etc/crontab` serves as a systemwide cron table. Enter the username to run the command directly after the time table and before the command. In [Example 11.1, “Entry in /etc/crontab”](#) (page 114), `root` is entered. Package-specific tables, located in `/etc/cron.d`, have the same format. See the `cron` man page (`man cron`).

Example 11.1 *Entry in /etc/crontab*

```
1-59/5 * * * * root test -x /usr/sbin/atrun && /usr/sbin/atrun
```

You cannot edit `/etc/crontab` by calling the command `crontab -e`. This file must be loaded directly into an editor, modified, then saved.

A number of packages install shell scripts to the directories `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly`, and `/etc/cron.monthly`, whose execution is controlled by `/usr/lib/cron/run-crons`. `/usr/lib/cron/run-crons` is run every 15 minutes from the main table (`/etc/crontab`). This guarantees that processes that may have been neglected can be run at the proper time.

To run the `hourly`, `daily`, or other periodic maintenance scripts at custom times, remove the time stamp files regularly using `/etc/crontab` entries (see [Example 11.2, “/etc/crontab: Remove Time Stamp Files”](#) (page 115), which removes the `hourly` one before every full hour, the `daily` one once a day at 2:14 a.m., etc.).

Example 11.2 */etc/crontab: Remove Time Stamp Files*

```
59 * * * * root rm -f /var/spool/cron/lastrun/cron.hourly
14 2 * * * root rm -f /var/spool/cron/lastrun/cron.daily
29 2 * * 6 root rm -f /var/spool/cron/lastrun/cron.weekly
44 2 1 * * root rm -f /var/spool/cron/lastrun/cron.monthly
```

Alternatively, set `DAILY_TIME` in `/etc/sysconfig/cron` to the time at which `cron.daily` should start. The setting of `MAX_NOT_RUN` ensures that the daily jobs get triggered to run, even if the user did not turn on the computer at the specified `DAILY_TIME` for a longer period of time. The maximum value of `MAX_NOT_RUN` is 14 days.

The daily system maintenance jobs are distributed to various scripts for reasons of clarity. They are contained in the package `aaa_base`. `/etc/cron.daily` contains, for example, the components `suse.de-backup-rpmdb`, `suse.de-clean-tmp`, or `suse.de-cron-local`.

11.1.3 Log Files: Package logrotate

There are a number of system services (*daemons*) that, along with the kernel itself, regularly record the system status and specific events to log files. This way, the administrator can regularly check the status of the system at a certain point in time, recognize errors or faulty functions, and troubleshoot them with pinpoint precision. These log files are normally stored in `/var/log` as specified by FHS and grow on a daily basis. The `logrotate` package helps control the growth of these files.

Configure logrotate with the file `/etc/logrotate.conf`. In particular, the `include` specification primarily configures the additional files to read. Programs that produce log files install individual configuration files in `/etc/logrotate.d`. For example, such files ship with the packages, e.g. `apache2` (`/etc/logrotate.d/apache2`) and `syslogd` (`/etc/logrotate.d/syslog`).

Example 11.3 *Example for `/etc/logrotate.conf`*

```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own lastlog or wtmp - we'll rotate them here
#/var/log/wtmp {
#    monthly
#    create 0664 root utmp
#    rotate 1
#}

# system-specific logs may be also be configured here.
```

logrotate is controlled through cron and is called daily by `/etc/cron.daily/logrotate`.

IMPORTANT

The `create` option reads all settings made by the administrator in `/etc/permissions*`. Ensure that no conflicts arise from any personal modifications.

11.1.4 The locate Command

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

11.1.5 The ulimit Command

With the `ulimit` (*user limits*) command, it is possible to set limits for the use of system resources and to have these displayed. `ulimit` is especially useful for limiting the memory available for applications. With this, an application can be prevented from using too much memory on its own, which could bring the system to a standstill.

`ulimit` can be used with various options. To limit memory usage, use the options listed in [Table 11.1, “ulimit: Setting Resources for the User”](#) (page 117).

Table 11.1 *ulimit: Setting Resources for the User*

<code>-m</code>	The maximum resident set size
<code>-v</code>	The maximum amount of virtual memory available to the shell
<code>-s</code>	The maximum size of the stack
<code>-c</code>	The maximum size of core files created
<code>-a</code>	All current limits are reported

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

Example 11.4 *ulimit: Settings in ~/.bashrc*

```
# Limits maximum resident set size (physical memory):
ulimit -m 98304

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

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

IMPORTANT

Not all shells support `ulimit` directives. PAM (for instance, `pam_limits`) offers comprehensive adjustment possibilities if you depend on encompassing settings for these restrictions.

11.1.6 The `free` Command

The `free` command is somewhat misleading if your goal is to find out how much RAM is currently being used. That information can be found in `/proc/meminfo`. These days, users with access to a modern operating system, such as Linux, should not really need to worry much about memory. The concept of *available RAM* dates back to before the days of unified memory management. The slogan *free memory is bad memory* applies well to Linux. As a result, Linux has always made the effort to balance out caches without actually allowing free or unused memory.

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

The kernel also contains other caches, such as the *slab cache*, where the caches used for network access are stored. This may explain differences between the counters in `/proc/meminfo`. Most, but not all of them, can be accessed via `/proc/slabinfo`.

11.1.7 Man Pages and Info Pages

For some GNU applications (such as `tar`), the man pages are no longer maintained. For these commands, use the `--help` option to get a quick overview of the info pages, which provide more in-depth instructions. Info is GNU's hypertext system. Read an introduction to this system by entering `info info`. Info pages can be viewed with Emacs by entering `emacs -f info` or directly in a console with `info`. You can also use `tkinfo`, `xinfo`, or the help system to view info pages.

11.1.8 Selecting Man-Pages Using the `man` Command

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 expected to type the section as the answer.

If you want to get back the previous behavior, set `MAN_POSIXLY_CORRECT=1` in a shell initialization file such as `~/ .bashrc`.

11.1.9 Settings for GNU Emacs

GNU Emacs is a complex work environment. The following sections cover the configuration files processed when GNU Emacs is started. More information is available at <http://www.gnu.org/software/emacs/>.

On start-up, Emacs reads several files containing the settings of the user, system administrator, and distributor for customization or preconfiguration. The initialization file `~/ .emacs` is installed to the home directories of the individual users from `/etc/skel`. `.emacs`, in turn, reads the file `/etc/skel/ .gnu-emacs`. To customize the program, copy `.gnu-emacs` to the home directory (with `cp /etc/skel/ .gnu-emacs ~/ .gnu-emacs`) and make the desired settings there.

`.gnu-emacs` defines the file `~/ .gnu-emacs-custom` as `custom-file`. If users make settings with the `customize` options in Emacs, the settings are saved to `~/ .gnu-emacs-custom`.

With SUSE Linux Enterprise Desktop, the `emacs` package installs the file `site-start.el` in the directory `/usr/share/emacs/site-lisp`. The file `site-start.el` is loaded before the initialization file `~/ .emacs`. Among other things, `site-start.el` ensures that special configuration files distributed with Emacs add-on packages, such as `psgml`, are loaded automatically. Configuration files of this type are located in `/usr/share/emacs/site-lisp`, too, and always begin with `suse-start-`. The local system administrator can specify systemwide settings in `default.el`.

More information about these files is available in the Emacs info file under *Init File*: [info:/emacs/InitFile](#). Information about how to disable loading these files (if necessary) is also provided at this location.

The components of Emacs are divided into several packages:

- The base package `emacs`.
- `emacs-x11` (usually installed): the program *with* X11 support.
- `emacs-nox`: the program *without* X11 support.
- `emacs-info`: online documentation in info format.
- `emacs-el`: the uncompiled library files in Emacs Lisp. These are not required at runtime.
- Numerous add-on packages can be installed if needed: `emacs-auctex` (for LaTeX), `psgml` (for SGML and XML), `gnuserv` (for client and server operation), and others.

11.2 Virtual Consoles

Linux is a multiuser and multitasking system. The advantages of these features can be appreciated even on a stand-alone PC system. In text mode, there are six virtual consoles

available. Switch between them using Alt + F1 to Alt + F6. The seventh console is reserved for X and the tenth console shows kernel messages. More or fewer consoles can be assigned by modifying the file `/etc/inittab`.

To switch to a console from X without shutting it down, use Ctrl + Alt + F1 to Ctrl + Alt + F6. To return to X, press Alt + F7.

11.3 Keyboard Mapping

To standardize the keyboard mapping of programs, changes were made to the following files:

```
/etc/inputrc
/etc/X11/Xmodmap
/etc/skel/.Xmodmap
/etc/skel/.exrc
/etc/skel/.less
/etc/skel/.lesskey
/etc/csh.cshrc
/etc/termcap
/usr/lib/terminfo/x/xterm
/usr/share/X11/app-defaults/XTerm
/usr/share/emacs/VERSION/site-lisp/term/*.el
```

These changes only affect applications that use `terminfo` entries or whose configuration files are changed directly (`vi`, `less`, etc.). Applications not shipped with the system should be adapted to these defaults.

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

Further settings are possible using the X Keyboard Extension (XKB). This extension is also used by the desktop environments GNOME (gswitchit) and KDE (kxkb).

TIP: For More Information

Information about XKB is available in `/etc/X11/xkb/README` and the documents listed there.

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

11.4 Language and Country-Specific Settings

The system is, to a very large extent, internationalized and can be modified for local needs in a flexible manner. In other words, internationalization (*I18N*) allows specific localizations (*L10N*). The abbreviations *I18N* and *L10N* are derived from the first and last letters of the words and, in between, the number of letters omitted.

Settings are made with `LC_` variables defined in the file `/etc/sysconfig/language`. This refers not only to *native language support*, but also to the categories *Messages* (Language), *Character Set*, *Sort Order*, *Time and Date*, *Numbers*, and *Money*. Each of these categories can be defined directly with its own variable or indirectly with a master variable in the file `language` (see the `locale` man page).

`RC_LC_MESSAGES`, `RC_LC_CTYPE`, `RC_LC_COLLATE`, `RC_LC_TIME`,
`RC_LC_NUMERIC`, `RC_LC_MONETARY`

These variables are passed to the shell without the `RC_` prefix and represent the listed categories. The shell profiles concerned are listed below. The current setting can be shown with the command `locale`.

`RC_LC_ALL`

This variable, if set, overwrites the values of the variables already mentioned.

`RC_LANG`

If none of the previous variables are set, this is the fallback. By default, only `RC_LANG` is set. This makes it easier for users to enter their own values.

`ROOT_USES_LANG`

A `yes` or `no` variable. If it is set to `no`, `root` always works in the POSIX environment.

The variables can be set with the YaST `sysconfig` editor (see [Section 9.3.1, “Changing the System Configuration Using the YaST sysconfig Editor”](#) (page 90)). The value of

such a variable contains the language code, country code, encoding, and modifier. The individual components are connected by special characters:

```
LANG=<language>[_<COUNTRY>].<Encoding>[@<Modifier>]]
```

11.4.1 Some Examples

You should always set the language and country codes together. Language settings follow the standard ISO 639 available at <http://www.evertype.com/standards/iso639/iso639-en.html> and <http://www.loc.gov/standards/iso639-2/>. Country codes are listed in ISO 3166 available at http://www.din.de/gremien/nas/nabd/iso3166ma/codlstp1/en_listp1.html.

It only makes sense to set values for which usable description files can be found in `/usr/lib/locale`. Additional description files can be created from the files in `/usr/share/i18n` using the command `localedef`. The description files are part of the `glibc-i18ndata` package. A description file for `en_US.UTF-8` (for English and United States) can be created with:

```
localedef -i en_US -f UTF-8 en_US.UTF-8
```

```
LANG=en_US.UTF-8
```

This is the default setting if American English is selected during installation. If you selected another language, that language is enabled but still with UTF-8 as the character encoding.

```
LANG=en_US.ISO-8859-1
```

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

```
LANG=en_IE@euro
```

The above example explicitly includes the Euro sign in a language setting. Strictly speaking, this setting is obsolete now, because UTF-8 also covers the Euro symbol. It is only useful if an application does not support UTF-8, but ISO-8859-15.

SuSEconfig reads the variables in `/etc/sysconfig/language` and writes the necessary changes to `/etc/SuSEconfig/profile` and `/etc/SuSEconfig/csh.cshrc`. `/etc/SuSEconfig/profile` is read or *sourced* by `/etc/profile`. `/etc/SuSEconfig/csh.cshrc` is sourced by `/etc/csh.cshrc`. This makes the settings available systemwide.

Users can override the system defaults by editing their `~/.bashrc` accordingly. For instance, if you do not want to use the systemwide `en_US` for program messages, include `LC_MESSAGES=es_ES` so messages are displayed in Spanish instead.

11.4.2 Locale Settings in `~/.i18n`

If you are not satisfied with locale system defaults, change the settings in `~/.i18n` according to the Bash scripting syntax. Entries in `~/.i18n` override system defaults from `/etc/sysconfig/language`. Use the same variable names but without the `RC_` namespace prefixes, for example, use `LANG` instead of `RC_LANG`:

```
LANG=cs_CZ.UTF-8
LC_COLLATE=C
```

11.4.3 Settings for Language Support

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

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

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

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

If desired, use the Norwegian variants `Nynorsk` and `Bokmål` instead (with additional fallback to `no`):

```
LANG="nn_NO"
```



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

or

```
LANG="nb_NO"
```

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

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

One problem that can arise is a separator used to delimit groups of digits not being recognized properly. This occurs if `LANG` is set to only a two-letter language code like `de`, but the definition file `glibc` uses is located in `/usr/share/lib/de_DE/LC_NUMERIC`. Thus `LC_NUMERIC` must be set to `de_DE` to make the separator definition visible to the system.

11.4.4 For More Information

- *The GNU C Library Reference Manual*, Chapter “Locales and Internationalization”. It is included in `glibc-info`.
- Markus Kuhn, *UTF-8 and Unicode FAQ for Unix/Linux*, currently at <http://www.cl.cam.ac.uk/~mgk25/unicode.html>.
- *Unicode-Howto*, by Bruno Haible: `/usr/share/doc/howto/en/html/Unicode-HOWTO.html`.

Printer Operation

SUSE® Linux Enterprise Desktop supports printing with many types of printers, including remote network printers. Printers can be configured with YaST or manually. For configuration instructions, refer to Section “Setting Up a Printer” (Chapter 5, *Setting Up Hardware Components with YaST*, ↑Deployment Guide). 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 12.8, “Troubleshooting”](#) (page 137).

CUPS is the standard print system in SUSE Linux Enterprise Desktop. CUPS is highly user-oriented. In many cases, it is compatible with LPRng or can be adapted with relatively little effort. LPRng is included in SUSE Linux Enterprise Desktop only for reasons of compatibility.

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

PostScript Printers

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

Standard 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 a decent print result. Linux may not be able to address some functions of extremely new and fancy printers, because the open source developers may still be working on these features. Except for HP developing HPLIP, there are currently no printer manufacturers who develop Linux drivers and make them available to Linux distributors under an open source license. Most of these printers are in the medium price range.

Proprietary Printers (Also Called GDI Printers)

These printers do not support any of the common printer languages. They use their own undocumented printer languages, which are subject to change when a new edition of a model is released. Usually only Windows drivers are available for these printers. See [Section 12.8.1, “Printers without Standard Printer Language Support”](#) (page 137) 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/en/OpenPrinting/>

The OpenPrinting.org printer database.

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

The Ghostscript Web page.

`/usr/share/doc/packages/ghostscript-library/catalog.devices`
List of included drivers.

The online databases always show the latest Linux support status. However, a Linux distribution can only integrate the drivers available at production time. Accordingly, a printer currently rated as “perfectly supported” may not have had this status when the latest SUSE Linux Enterprise Desktop version was released. Thus, the databases may not necessarily indicate the correct status, but only provide an approximation.

12.1 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.

12.2 Methods and Protocols for Connecting Printers

There are various possibilities for connecting a printer to the system. The configuration of the CUPS print system does not distinguish between a local printer and a printer connected to the system over the network. In Linux, local printers must be connected as described in the manual of the printer manufacturer. CUPS supports serial, USB, parallel, and SCSI connections.

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.

12.3 Installing the Software

PPD (PostScript printer description) is the computer language that describes the properties, like resolution, and options, such as the availability of a duplex unit. These descriptions are required for using various printer options in CUPS. Without a PPD file, the print data would be forwarded to the printer in a “raw” state, which is usually not desired. During the installation of SUSE Linux Enterprise Desktop, 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 12.7.2, “PPD Files in Various Packages”](#) (page 135) and [Section 12.8.2, “No Suitable PPD File Available for a PostScript Printer”](#) (page 138).

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” (Chapter 5, *Setting Up Hardware Components with YaST*, ↑Deployment Guide)). Subsequently, the PPD file can be selected during the installation.

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

12.4 Network Printers

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

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

`socket`

Socket refers to a connection in which the data is sent to an Internet socket without first performing a data handshake. Some of the socket port numbers that are commonly used are 9100 or 35. The device URI (uniform resource identifier) syntax is `socket://IP.of.the.printer:port`, for example,
`socket://192.168.2.202:9100/`.

`LPD` (Line Printer Daemon)

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

`IPP` (Internet Printing Protocol)

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

SMB (Windows Share)

CUPS also supports printing on printers connected to Windows shares. The protocol used for this purpose is SMB. SMB uses the port numbers 137, 138, and 139.

Example device URIs are

```
smb://user:password@workgroup/smb.example.com/printer,  
smb://user:password@smb.example.com/printer, and  
smb://smb.example.com/printer.
```

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

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

12.4.1 Configuring CUPS with Command Line Tools

Apart from setting CUPS options with YaST when configuring a network printer, CUPS can be configured with command line tools like `lpadmin` and `lpoptions`. You need a device URI consisting of a back-end, such as `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 class and print queues. To add a print queue, use the following syntax:

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

Then the device (`-v`) is available as *queue* (`-p`), using the specified PPD file (`-P`). This means that you must know the PPD file and the 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-levell.ppd.gz -E
```

For more options of `lpadmin`, see the man page of `lpadmin(1)`.

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

1 First, list all options:

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

Example:

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

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

2 Change the option with `lpadmin`:

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

3 Check the new setting:

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

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

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

12.5 Graphical Printing Interfaces

Tools such as `xpp` and the KDE program `KPrinter` provide a graphical interface for choosing queues and setting both CUPS standard options and printer-specific options made available through the PPD file. You can even use `KPrinter` as the standard printing interface of non-KDE applications. In the print dialog of these applications, specify

either `kprinter` or `kprinter --stdin` as the print command. The command to use depends on how the application transmits the data—just try which one works. If set up correctly, the application should open the KPrinter dialog whenever a print job is issued from it, so you can use the dialog to select a queue and set other printing options. This requires that the application's own print setup does not conflict with that of KPrinter and that printing options are only changed through KPrinter after it has been enabled. More information on KPrinter is available in Chapter 7, *Managing Print Jobs* (↑KDE User Guide).

12.6 Printing from the Command Line

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

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

12.7 Special Features in SUSE Linux Enterprise Desktop

A number of CUPS features have been adapted for SUSE Linux Enterprise Desktop. Some of the most important changes are covered here.

12.7.1 CUPS and Firewall

After having performed a default installation of SUSE Linux Enterprise Desktop, SuSEfirewall2 is active and the network interfaces are configured to be in the `External Zone` which blocks incoming traffic. These default settings have to be adjusted when using CUPS. More information about the SuSEfirewall2 configuration is available in Section “SuSEfirewall2” (Chapter 15, *Masquerading and Firewalls*, ↑Security Guide).

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.

12.7.2 PPD Files in Various Packages

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

The configuration using only PPD files and no other information sources has the advantage that the PPD files in `/usr/share/cups/model` can be modified freely. The YaST printer configuration recognizes changes and regenerates the vendor and model database. For example, if you only have PostScript printers, normally you do not need the Foomatic PPD files in the `cups-drivers` package or the 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 any 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.

12.8 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.

12.8.1 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 because the Windows driver uses the GDI interface, these printers are also called *GDI printers*. The actual problem is not the programming interface, but the fact that these printers can only be addressed with the proprietary printer language of the respective printer model.

Some GDI printers can be switched to operate either in GDI mode or one of the standard printer languages. See the manual of the printer whether it is possible. Some models require a special Windows software to do the switch (note that the Windows printer driver may always switch the printer back into GDI mode when printing from Windows). For other GDI printers there are extension modules for a standard printer language available.

Some manufacturers provide proprietary drivers for their printers. The disadvantage of proprietary printer drivers is that there is no guarantee that these work with the installed print system and that they are suitable for the various hardware platforms. In contrast,

printers that support a standard printer language do not depend on a special print system version or a special hardware platform.

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

12.8.2 No Suitable PPD File Available for a PostScript Printer

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

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

12.8.3 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 12.1](#), “`/etc/modprobe.conf: Interrupt Mode for the First Parallel Port`” (page 139). Before activating the interrupt mode, check the file `/proc/interrupts` to see which interrupts are already in use. Only the interrupts currently being used are displayed. This may change depending on which hardware components are active. The interrupt for the parallel port must not be used by any other device. If you are not sure, use the polling mode with `irq=none`.

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

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

12.8.4 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 in [Example 12.2, “Error Message from lpd”](#) (page 140), the problem is caused by the remote `lpd`.

Example 12.2 *Error Message from lpd*

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

Checking a Remote `cupsd`

By default, the CUPS network server should broadcast its queues every 30 seconds on UDP port 631. Accordingly, the following command can be used to test whether there is a CUPS network server in the network. 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 12.3, “Broadcast from the CUPS Network Server”](#) (page 140).

Example 12.3 *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 a lot of print jobs. Because this is caused by the spooler in the print server box, there is nothing you can do about it. As a work-around, circumvent the spooler in the print server box by addressing the printer connected to the print server box directly with TCP socket. See [Section 12.4, “Network Printers”](#) (page 131).

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

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

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

Enter a command like

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

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

12.8.5 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 the further processing on the recipient fails, for example, if the printer is not able to print the printer-specific data, the print system does not notice this. If the printer is not able to print the printer-specific data, select a different PPD file that is more suitable for the printer.

12.8.6 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 decides whether and how many attempts make sense until the data transfer is reported as impossible. Because further attempts would be in vain, `cupsd` disables printing for the respective queue. After eliminating the cause of the problem, the system administrator must reenable printing with the command `cupsenable`.

12.8.7 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` accepts a print job, it is assigned a new job number. Therefore, the job number on the client host is different from the job number on the server. Because a print job is usually forwarded immediately, it cannot be deleted with the job number on the client host, because the client `cupsd` regards the print job as completed as soon as it has been forwarded to the server `cupsd`.

To delete the print job on the server, use a command such as `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
```

12.8.8 Defective Print Jobs and Data Transfer Errors

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

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

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

12.8.9 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.

12.8.10 For More Information

Solutions to many specific problems are presented in the Novell Knowledgebase (<http://support.novell.com/>). Locate the relevant articles with a text search for CUPS.

Dynamic Kernel Device Management with udev

13

The kernel can add or remove almost any device in the running system. Changes in device state (whether a device is plugged in or removed) need to be propagated to userspace. Devices need to be configured as soon as they are plugged in and discovered. Users of a certain device need to be informed about any state changes of this device. udev provides the needed infrastructure to dynamically maintain the device node files and symbolic links in the `/dev` directory. udev rules provide a way to plug external tools into the kernel device event processing. This enables you to customize udev device handling, for example, by adding certain scripts to execute as part of kernel device handling, or request and import additional data to evaluate during device handling.

13.1 The `/dev` Directory

The device nodes in the `/dev` directory provide access to the corresponding kernel devices. With udev, the `/dev` directory reflects the current state of the kernel. Every kernel device has one corresponding device file. If a device is disconnected from the system, the device node is removed.

The content of the `/dev` directory is kept on a temporary file system and all files are created from scratch at every system start-up. Manually created or changed files intentionally do not survive a reboot. Static files and directories that should always be present in the `/dev` directory regardless of the state of the corresponding kernel device can be placed in the `/lib/udev/devices` directory. At system start-up, the contents of that directory is copied to the `/dev` directory with the same ownership and permissions as the files in `/lib/udev/devices`.

13.2 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 13.6, “Influencing Kernel Device Event Handling with udev Rules”](#) (page 149).

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.

13.3 Drivers, Kernel Modules, and Devices

The kernel bus drivers probe for devices. For every detected device, the kernel creates an internal device structure and the driver core sends a uevent to the udev daemon. Bus devices identify themselves by a specially-formatted ID, which tells what kind of device it is. Usually these IDs consist of vendor and product ID and other subsystem-specific values. Every bus has its own scheme for these IDs, called `MODALIAS`. The kernel takes the device information, composes a `MODALIAS` ID string from it, and sends that string along with the event. For a USB mouse, it looks like this:

```
MODALIAS=usb:v046DpC03Ed2000dc00dsc00dp00ic03isc01ip02
```

Every device driver carries a list of known aliases for devices it can handle. The list is contained in the kernel module file itself. The program `depmod` reads the ID lists and creates the file `modules.alias` in the kernel's `/lib/modules` directory for all currently available modules. With this infrastructure, module loading is as easy as

calling `modprobe` for every event that carries a `MODALIAS` key. If `modprobe $MODALIAS` is called, it matches the device alias composed for the device with the aliases provided by the modules. If a matching entry is found, that module is loaded. All this is triggered by `udev` and happens automatically.

13.4 Booting and Initial Device Setup

All device events happening during the boot process before the `udev` daemon is running are lost, because the infrastructure to handle these events lives on the root file system and is not available at that time. To cover that loss, the kernel provides a `uevent` file 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.

13.5 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 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`.

13.6 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 13.1 *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 assign 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 a look. 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/usb/lp0`.

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.

13.6.1 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.

13.6.2 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.

13.6.3 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.

13.6.4 Using udev Assign Keys

In contrast to the match keys described above, assign keys do not describe conditions that must be met, but 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. 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.

13.7 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
```

13.8 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.

13.9 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.

udev
Information about the udev event managing daemon.

The X Window System

The X Window System (X11) is the de facto standard for graphical user interfaces in UNIX. X is network-based, enabling applications started on one host to be displayed on another host connected over any kind of network (LAN or Internet). This chapter describes the setup and optimization of the X Window System environment, and provides background information about the use of fonts in SUSE® Linux Enterprise Desktop.

14.1 Manually Configuring the X Window System

By default, the X Window System is configured with the SaX2 interface, described in Section “Setting Up Graphics Card and Monitor” (Chapter 5, *Setting Up Hardware Components with YaST*, ↑Deployment Guide). Alternatively it can be configured manually by editing the its configuration files.

WARNING: Faulty X Configurations can Damage Your Hardware

Be very careful when configuring your X Window System. Never start the X Window System until the configuration is finished. A misconfigured system can cause irreparable damage to your hardware (this applies especially to fixed-frequency monitors). The creators of this book and SUSE Linux Enterprise Desktop cannot be held responsible for any resulting damage. This information has been carefully researched, but this does not guarantee that all methods presented here are correct and cannot damage your hardware.

The command `sax2` creates the `/etc/X11/xorg.conf` file. This is the primary configuration file of the X Window System. Find all the settings here concerning your graphics card, mouse, and monitor.

IMPORTANT: Using X -configure

Use `X -configure` to configure your X setup if previous tries with SUSE Linux Enterprise Desktop's SaX2 have failed. If your setup involves proprietary binary-only drivers, `X -configure` cannot work.

The following sections describe the structure of the configuration file `/etc/X11/xorg.conf`. It consists of several sections, each one dealing with a certain aspect of the configuration. Each section starts with the keyword `Section <designation>` and ends with `EndSection`. The following convention applies to all sections:

```
Section "designation"
    entry 1
    entry 2
    entry n
EndSection
```

The section types available are listed in [Table 14.1, “Sections in /etc/X11/xorg.conf”](#) (page 160).

Table 14.1 *Sections in /etc/X11/xorg.conf*

Type	Meaning
Files	The paths used for fonts and the RGB color table.
ServerFlags	General switches for the server behavior.
Module	A list of modules the server should load
InputDevice	Input devices, like keyboards and special input devices (touch-pads, joysticks, etc.), are configured in this section. Important parameters in this section are <code>Driver</code> and the options defining the <code>Protocol</code> and <code>Device</code> . You normally have one <code>InputDevice</code> section per device attached to the computer.

Type	Meaning
Monitor	The monitor used. Important elements of this section are the <code>Identifier</code> , which is referred to later in the <code>Screen</code> definition, the refresh rate <code>VertRefresh</code> , and the synchronization frequency limits (<code>HorizSync</code> and <code>VertRefresh</code>). Settings are given in MHz, kHz, and Hz. Normally, the server refuses any modeline that does not correspond with the specification of the monitor. This prevents too high frequencies from being sent to the monitor by accident.
Modes	The modeline parameters for the specific screen resolutions. These parameters can be calculated by <code>SaX2</code> on the basis of the values given by the user and normally do not need to be changed. Intervene manually at this point if, for example, you want to connect a fixed frequency monitor. Find details of the meaning of individual number values in the HOWTO files in <code>/usr/share/doc/howto/en/html/XFree86-Video-Timings-HOWTO</code> (available in the <code>howtoenh</code> package). To calculate VESA modes manually, you can use the tool <code>cvt</code> . For example, to calculate a modeline for a 1680x1050@60Hz monitor, use the command <code>cvt 1680 1050 60</code> .
Device	A specific graphics card. It is referenced by its descriptive name. The options available in this section strongly depend on the driver used. For example, if you use the <code>i810</code> driver, find more information about the available options in the manual page <code>man 4 i810</code> .
Screen	Combines a <code>Monitor</code> and a <code>Device</code> to form all the necessary settings for <code>X.Org</code> . In the <code>Display</code> subsection, specify the size of the virtual screen (<code>Virtual</code>), the <code>ViewPort</code> , and the <code>Modes</code> used with this screen. Note that some drivers demand that all of the used configurations must be present in the <code>Display</code> section at some place. For ex-

Type	Meaning
	ample, if you use a laptop and want to use an external monitor that is bigger than the internal LCD, it might be necessary to add a bigger resolution than supported by the internal LCD at the end of the <code>Modes</code> line.
<code>ServerLayout</code>	The layout of a single or multihead configuration. This section binds the input devices <code>InputDevice</code> and the display devices <code>Screen</code> .
<code>DRI</code>	Provides information for the Direct Rendering Infrastructure (DRI).

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

There can be several different `Monitor` and `Device` sections in `xorg.conf`. Even multiple `Screen` sections are possible. The `ServerLayout` section determines which of these sections is used.

14.1.1 Screen Section

The screen section combines a monitor with a device section and determines the resolution and color depth to use. A screen section might resemble [Example 14.1, “Screen Section of the File `/etc/X11/xorg.conf`”](#) (page 163).

Example 14.1 Screen Section of the File */etc/X11/xorg.conf*

```
Section "Screen"❶
    DefaultDepth 16❷
    SubSection "Display"❸
        Depth 16❹
        Modes "1152x864" "1024x768" "800x600"❺
        Virtual 1152x864❻
    EndSubSection
    SubSection "Display"
        Depth 24
        Modes "1280x1024"
    EndSubSection
    SubSection "Display"
        Depth 32
        Modes "640x480"
    EndSubSection
    SubSection "Display"
        Depth 8
        Modes "1280x1024"
    EndSubSection
    Device "Device[0]"
    Identifier "Screen[0]"❼
    Monitor "Monitor[0]"
EndSection
```

- ❶ Section determines the section type, in this case `Screen`.
- ❷ `DefaultDepth` determines the color depth to use by default unless another color depth is explicitly specified.
- ❸ For each color depth, different `Display` subsections are specified.
- ❹ `Depth` determines the color depth to be used with this set of `Display` settings. Possible values are 8, 15, 16, 24, and 32, though not all of these might be supported by all X server modules or resolutions.
- ❺ The `Modes` section comprises a list of possible screen resolutions. The list is checked by the X server from left to right. For each resolution, the X server searches for a suitable `Modeline` in the `Modes` section. The `Modeline` depends on the capability of both the monitor and the graphics card. The `Monitor` settings determine the resulting `Modeline`.

The first resolution found is the `Default` mode. With `Ctrl + Alt + +` (on the number pad), switch to the next resolution in the list to the right. With `Ctrl + Alt + -` (on the number pad), switch to the previous. This enables you to vary the resolution while X is running.

- ⑥ The last line of the `Display` subsection with `Depth 16` refers to the size of the virtual screen. The maximum possible size of a virtual screen depends on the amount of memory installed on the graphics card and the desired color depth, not on the maximum resolution of the monitor. If you omit this line, the virtual resolution is just the physical resolution. Because modern graphics cards have a large amount of video memory, you can create very large virtual desktops. However, you may no longer be able to use 3D functionality if you fill most of the video memory with a virtual desktop. If, for example, the card has 16 MB of video RAM, the virtual screen can take up to 4096x4096 pixels in size at 8-bit color depth. Especially for accelerated cards, however, it is not recommended to use all your memory for the virtual screen, because the card's memory is also used for several font and graphics caches.
- ⑦ The `Identifier` line (here `Screen[0]`) gives this section a defined name with which it can be uniquely referenced in the following `ServerLayout` section. The lines `Device` and `Monitor` specify the graphics card and the monitor that belong to this definition. These are just links to the `Device` and `Monitor` sections with their corresponding names or *identifiers*. These sections are discussed in detail below.

14.1.2 Device Section

A device section describes a specific graphics card. You can have as many device entries in `xorg.conf` as you like, provided their names are differentiated using the keyword `Identifier`. If you have more than one graphics card installed, the sections are simply numbered in order. The first one is called `Device[0]`, the second one `Device[1]`, and so on. The following file shows an excerpt from the `Device` section of a computer with a Matrox Millennium PCI graphics card (as configured by SaX2):

```
Section "Device"
    BoardName      "MGA2064W"
    BusID          "0:19:0"❶
    Driver         "mga"❷
    Identifier     "Device[0]"
    VendorName     "Matrox"
    Option         "sw_cursor"
EndSection
```

- ❶ The `BusID` refers to the PCI or AGP slot in which the graphics card is installed. This matches the ID displayed by the command `lspci`. The X server needs details

in decimal form, but `lspci` displays these in hexadecimal form. The value of `BusID` is automatically detected by `SaX2`.

- ② The value of `Driver` is automatically set by `SaX2` and specifies which driver to use for your graphics card. If the card is a Matrox Millennium, the driver module is called `mga`. The X server then searches through the `ModulePath` defined in the `Files` section in the `drivers` subdirectory. In a standard installation, this is the `/usr/lib/xorg/modules/drivers` directory or the `/usr/lib64/xorg/modules/drivers` directory for 64-Bit operating systems directory. `_drv.o` is added to the name, so, in the case of the `mga` driver, the driver file `mga_drv.o` is loaded.

The behavior of the X server or of the driver can also be influenced through additional options. An example of this is the option `sw_cursor`, which is set in the device section. This deactivates the hardware mouse cursor and depicts the mouse cursor using software. Depending on the driver module, there are various options available, which can be found in the description files of the driver modules in the directory `/usr/share/doc/package_name`. Generally valid options can also be found in the manual pages (`man xorg.conf`, `man 4 <driver module>`, and `man 4 chips`).

If the graphics card has multiple video connectors, it is possible to configure the different devices of this single card as one single view. Use `SaX2` to set up your graphics interface this way.

14.1.3 Monitor and Modes Section

Like the `Device` sections, the `Monitor` and `Modes` sections describe one monitor each. The configuration file `/etc/X11/xorg.conf` can contain as many `Monitor` sections as desired. Each `Monitor` section references a `Modes` section with the line `UseModes` if available. If no `Modes` section is available for the `Monitor` section, the X server calculates appropriate values from the general synchronization values. The server layout section specifies which `Monitor` section is relevant.

Monitor definitions should only be set by experienced users. The modelines are an important part of the `Monitor` sections. Modelines set horizontal and vertical timings for the respective resolution. The monitor properties, especially the allowed frequencies, are stored in the `Monitor` section. Standard VESA modes can be generated with the utility `cvt`. For more information read the manual page of `cvt` `man cvt`.

WARNING

Unless you have in-depth knowledge of monitor and graphics card functions, do not change the modelines, because this could severely damage your monitor.

Those who try to develop their own monitor descriptions should be very familiar with the documentation in `/usr/share/X11/doc`. Install the package `xorg-x11-doc` to find PDFs and HTML pages.

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

14.2 Installing and Configuring Fonts

The installation of additional fonts in SUSE Linux Enterprise Desktop is very easy. Simply copy the fonts to any directory located in the X11 font path (see [Section 14.2.1, “X11 Core Fonts”](#) (page 167)). To enable use of the fonts, the installation directory should be a subdirectory of the directories configured in `/etc/fonts/fonts.conf` (see [Section 14.2.2, “Xft”](#) (page 168)) 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>
<include ignore_missing="yes">conf.d</include>
```

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

```
<dir>/usr/lib64/ooo-2.0/share/fonts</dir>
<dir>/usr/lib/jvm/java-1_4_2-sun-1.4.2.11/jre/lib/fonts</dir>
<dir>/usr/lib64/jvm/java-1.5.0-sun-1.5.0_07/jre/lib/fonts</dir>
<dir>/usr/X11R6/lib/Acrobat7/Resource/Font</dir>
<dir>/usr/X11R6/lib/Acrobat7/Resource/Font/PFM</dir>
```

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

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

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

The procedure is the same for bitmap fonts, TrueType and OpenType fonts, and Type1 (PostScript) fonts. All these font types can be installed into any directory.

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

14.2.1 X11 Core Fonts

Today, the X11 core font system supports not only bitmap fonts but also scalable fonts, like Type1 fonts, TrueType, and OpenType fonts. Scalable fonts are only supported without antialiasing and subpixel rendering and the loading of large scalable fonts with glyphs for many languages may take a long time. Unicode fonts are also supported, but their use may be slow and require more memory.

The X11 core font system has a few inherent weaknesses. It is outdated and can no longer be extended in a meaningful way. Although it must be retained for reasons of backward compatibility, the more modern Xft and fontconfig system should be used if at all possible.

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

If the X server is already active, newly installed fonts in mounted directories can be made available with the command `xset fp rehash`. This command is executed by `SuSEconfig --module fonts`. Because the command `xset` needs access to the running X server, this only works if `SuSEconfig --module fonts` is started from a shell that has access to the running X server. The easiest way to achieve this is to assume `root` permissions by entering `su` and the `root` password. `su` transfers the access permissions of the user who started the X server to the `root` shell. To check if the fonts were installed correctly and are available by way of the X11 core font system, use the command `xlsfonts` to list all available fonts.

By default, SUSE Linux Enterprise Desktop uses UTF-8 locales. Therefore, Unicode fonts should be preferred (font names ending with `iso10646-1` in `xlsfonts` output). All available Unicode fonts can be listed with `xlsfonts | grep iso10646-1`. Nearly all Unicode fonts available in SUSE Linux Enterprise Desktop contain at least the glyphs needed for European languages (formerly encoded as `iso-8859-*`).

14.2.2 Xft

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

languages. Direct access to the font files is very useful for embedding fonts for printing to make sure that the printout looks the same as the screen output.

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

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

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

and end with

```
</fontconfig>
```

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

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

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

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

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

to disable antialiasing for all fonts or

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

```
</edit>
</match>
```

to disable antialiasing for specific fonts.

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

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

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

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

Use the command `fc-list` to find out which fonts are installed and available for use. For instance, the command `fc-list` returns a list of all fonts. To find out which of the available scalable fonts (`:scalable=true`) contain all glyphs required for Hebrew (`:lang=he`), their font names (`family`), their style (`style`), their weight (`weight`), and the name of the files containing the fonts, enter the following command:

```
fc-list ":lang=he:scalable=true" family style weight
```

The output of this command could look like the following:

```
Lucida Sans:style=Demibold:weight=200
DejaVu Sans:style=Bold Oblique:weight=200
```

```

Lucida Sans Typewriter:style=Bold:weight=200
FreeSerif:style=Bold, polkrepko:weight=200
FreeSerif:style=Italic, ležeče:weight=80
FreeSans:style=Medium, navadno:weight=80
DejaVu Sans:style=Oblique:weight=80
FreeSans:style=Oblique, ležeče:weight=80

```

Important parameters that can be queried with `fc-list`:

Table 14.2 *Parameters of `fc-list`*

Parameter	Meaning and Possible Values
<code>family</code>	Name of the font family, for example, <code>FreeSans</code> .
<code>foundry</code>	The manufacturer of the font, for example, <code>urw</code> .
<code>style</code>	The font style, such as <code>Medium</code> , <code>Regular</code> , <code>Bold</code> , <code>Italic</code> , or <code>Heavy</code> .
<code>lang</code>	The language that the font supports, for example, <code>de</code> for German, <code>ja</code> for Japanese, <code>zh-TW</code> for traditional Chinese, or <code>zh-CN</code> for simplified Chinese.
<code>weight</code>	The font weight, such as <code>80</code> for regular or <code>200</code> for bold.
<code>slant</code>	The slant, usually <code>0</code> for none and <code>100</code> for italic.
<code>file</code>	The name of the file containing the font.
<code>outline</code>	<code>true</code> for outline fonts or <code>false</code> for other fonts.
<code>scalable</code>	<code>true</code> for scalable fonts or <code>false</code> for other fonts.
<code>bitmap</code>	<code>true</code> for bitmap fonts or <code>false</code> for other fonts.
<code>pixelsize</code>	Font size in pixels. In connection with <code>fc-list</code> , this option only makes sense for bitmap fonts.

14.3 For More Information

Install the packages `xorg-x11-doc` and `howtoenh` to get more in-depth information on X11. More information on the X11 development can be found on the project's home page at <http://www.x.org>.

Many of the drivers delivered with the package `xorg-x11-driver-video` are described in detail in a manual page. For example, if you use the `radeon` driver, find more information about this driver in `man 4 radeon`.

Information about third-party drivers should be available in `/usr/share/doc/packages/<package_name>`. For example, the documentation of `x11-video-nvidiaG01` is available in `/usr/share/doc/packages/x11-video-nvidiaG01` after the package was installed.

Accessing File Systems with FUSE

15

FUSE is the acronym for *file system in userspace*. This means you can configure and mount a file system as an unprivileged user. Normally, you have to be `root` for this task. FUSE alone is a kernel module. Combined with plug-ins, it allows you to extend FUSE to access almost all file systems like remote SSH connections, ISO images, and more

15.1 Configuring FUSE

Before you can use FUSE, you have to install the package `fuse`. Depending which file system you want to use, you need additional plug-ins in different packages. Use YaST to search for these packages and use `fuse` or `file system` as keywords.

Generally you do not have to configure FUSE, you just use it. However, it is a good idea to create a directory where all your mountpoints are combined. For example, you can create a directory `~/mounts` and insert your subdirectories for your different file systems there.

15.2 Mounting an NTFS Partition

NTFS, the *New Technology File System*, is the default file system of several Windows versions, like Windows NT, 2000, XEP, and Vista. It supersedes the FAT file systems. To mount a Windows partition as a normal user, proceed as follows:

- 1 Become `root` and install the package `ntfs-3g`.
- 2 Create the directory `/media/windows`.
- 3 Find out which Window partition you need. Use YaST and start the partitioner module to see which partition belongs to Windows, but do not change anything. Alternatively, become `root` and execute `/sbin/fdisk -l`. Look for partitions with a partition type of HPFS/NTFS.
- 4 Mount the partition in read-write mode. Replace the placeholder *DEVICE* with your respective Windows partition:

```
ntfs-3g /dev/DEVICE /media/windows
```

If you want to use your Windows partition in readonly mode, append `-o`:

```
ntfs-3g /dev/DEVICE /media/windows -o ro
```

The command `ntfs-3g` uses the current user (uid) and group id (gid) to mount the given device. If you want to set the write permissions to a different user, use the command `id USER` to get the output of the uid and gid values. Set it with:

```
id tux
uid=1000(tux) gid=100(users) Gruppen=100(users),16(dialout),33(video)
ntfs-3g /dev/DEVICE /media/windows -o uid=1000,gid=100
```

Find additional options in the manpage.

To unmount the resource, use:

```
fusermount -u /media/windows
```

15.3 Mounting Remote File System with SSHFS

SSH, the secure shell network protocol, can be used to exchange data between two computers using a secure channel. To establish a SSH connection through FUSE, proceed as follows:

- 1 Install the package `sshfs`.

- 2 Create a directory, where you want to access the remote computer. A good idea is to use `~/mounts/HOST`. Replace *HOST* with the name of your remote computer.
- 3 Mount the remote file system:


```
sshfs USER:HOST ~/mounts/HOST
```

Replace *USER* and *HOST* with your respective values.
- 4 Enter your password for the remote computer.

15.4 Mounting an ISO File System

To look into an ISO image, you can mount it with the `fuseiso` package:

- 1 Install the package `fuseiso`.
- 2 Create the directory `~/mounts/iso`.
- 3 Mount the ISO image:

```
fuseiso ISO_IMAGE ~/mounts/iso
```

You can only read content from the ISO image, but you can not write back.

15.5 Available FUSE Plug-ins

FUSE depends on plug-ins. The following table lists common plug-ins.

Table 15.1 *Available FUSE Plug-ins*

<code>fuseiso</code>	mounts CD-ROM images with ISO9660 file systems in them
<code>ntfs-3g</code>	mount NTFS volumes (with read and write support)
<code>sshfs</code>	file system client based on SSH file transfer protocol

15.6 For More Information

See the homepage <http://fuse.sourceforge.net> of FUSE for more information.

Part III. Mobile Computers

Mobile Computing with Linux

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.

16.1 Laptops

The hardware of laptops differs from that of a normal desktop system. This is because criteria like exchangeability, occupied space, and power consumption are relevant properties. The manufacturers of mobile hardware have developed 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: SUSE Linux Enterprise Desktop and Tablet PCs

SUSE Linux Enterprise Desktop 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 18, Using Tablet PCs](#) (page 201).

16.1.1 Power Conservation

The inclusion of energy-optimized system components when manufacturing laptops contributes to their suitability for use without access to the electrical power grid. Their contribution towards conservation of power is at least as important as that of the operating system. SUSE® Linux Enterprise Desktop 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:

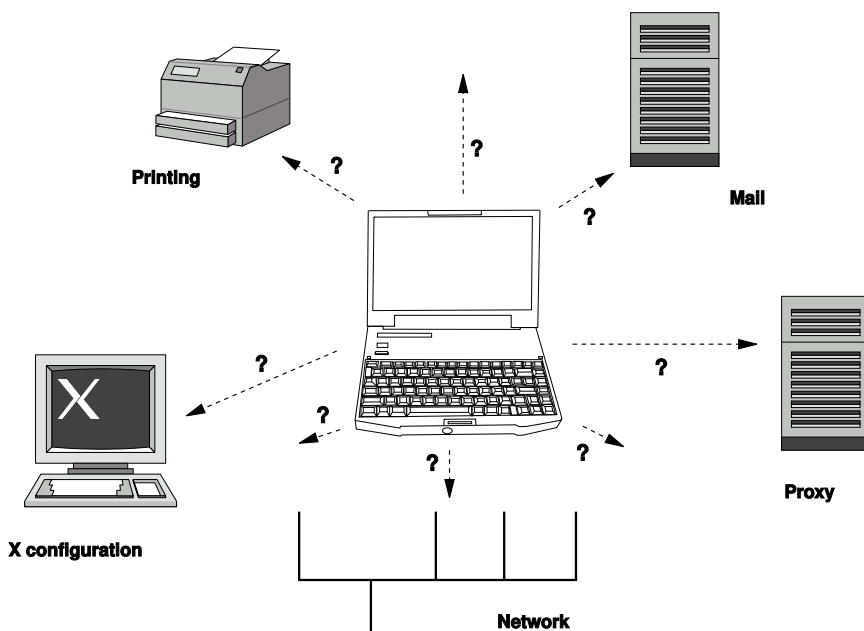
- Throttling the CPU speed.
- Switching off the display illumination during pauses.
- Manually adjusting the display illumination.
- Disconnecting unused, hotplug-enabled accessories (USB CD-ROM, external mouse, unused PCMCIA cards, WLAN, etc.).
- Spinning down the hard disk when idling.

For more information desktop specific power management, see the Section “Controlling Your Desktop’s Power Management” (Chapter 2, *Working with Your Desktop*, ↑GNOME User Guide) on how to use the GNOME Power Manager. More information about the KDE power management applet is available at Chapter 9, *Controlling Your Desktop’s Power Management* (↑KDE User Guide).

16.1.2 Integration in Changing Operating Environments

Your system needs to adapt to changing operating environments when used for mobile computing. A lot of services depend on the environment and the underlying clients must be reconfigured. SUSE Linux Enterprise Desktop handles this task for you.

Figure 16.1 *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 beamer or an external monitor, the different display configurations must be available.

SUSE Linux Enterprise Desktop offers several ways of integrating a laptop 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 to NetworkManager. For more information about the desktop applets, see [Section 23.4, “Using KDE NetworkManager Widget”](#) (page 294) and [Section 23.5, “Using GNOME NetworkManager Applet”](#) (page 295).

Table 16.1 *Use Cases for NetworkManager*

My computer...	Use NetworkManager
is a laptop	Yes
is sometimes attached to different networks	Yes
provides network services (such as DNS or DHCP)	No
only uses a static IP address	No

Use the YaST tools to configure networking whenever NetworkManager should not handle network configuration.

SCPM

SCPM (system configuration profile management) allows storage of arbitrary configuration states of a system into a kind of “snapshot” called a *profile*. Profiles can be created for different situations. They are useful when a system is operated in changing environments (home network, office network). It is always possible

to switch between profiles. To get SCPM up and running on your system, install the package `kscpm`, add the Profile Chooser KDE applet to your panel, enable SCPM using the YaST Profile Management module, and configure the users that should be allowed to switch profiles without the need of entering the `root` password. Determine whether profile changes should survive a system reboot or whether they should be discarded upon shutdown. Make sure all resource groups (i.e. services like network and printer, for example) are active. Proceed to creating actual profiles using the SUMF (SCPM Unified Management Front-End) tool which is started via Profile Chooser. Create profiles for all the different setups you want to use this system in. Switching between profiles can either be done in the running system via the Profile Chooser applet or at system boot time via the `F3` key. When switching profiles, SCPM automatically adjusts your system configuration to the new environment laid out in the profile you have chosen.

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 21, *SLP Services in the Network*](#) (page 281).

16.1.3 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 SUSE Linux Enterprise Desktop provides for each task.

System Monitoring

Two KDE system monitoring tools are provided by SUSE Linux Enterprise Desktop:

KPowersave

KPowersave is an applet that displays the state of the rechargeable battery in the control panel. The icon adjusts to represent the type of power supply. When working on AC power, a small plug icon is displayed. When working on batteries, the icon changes to a battery. The corresponding menu opens the YaST module for power management after requesting the `root` password. This allows setting the behavior of the system for different power sources.

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 Preferences and System Monitor.

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 as described in GNOME User Guide (↑GNOME User Guide) and KDE User Guide (↑KDE User Guide). The e-mail client must be configured so that the same folder is always accessed for `Sent` messages. This ensures that all messages are available along with their status information after the synchronization process has completed. Use an SMTP server implemented in the mail client for sending messages instead of the 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.

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 [Section 20.1, “Wireless LAN”](#) (page 271).

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 visible

range. Bluetooth is also used to connect wireless system components, like a keyboard or mouse. The range of this technology is, however, not sufficient to connect remote systems to a network. WLAN is the technology of choice for communicating through physical obstacles like walls.

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.

16.1.4 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. SUSE Linux Enterprise Desktop 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 SUSE Linux Enterprise Desktop 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 it takes place. 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 20, *Wireless Communication*** (page 271).

16.2 Mobile Hardware

SUSE Linux Enterprise Desktop 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 Volume* (GNOME) feature of either file manager. The handling of removable media by your desktop is described in more detail in GNOME User Guide (↑GNOME User Guide) and KDE User Guide (↑KDE User Guide).

External Hard Disks (USB and FireWire)

As soon as an external hard disk has been 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 The GIMP. For a short introduction to digiKam, f-spot and The GIMP, see Chapter 21, *Managing Your Digital Image Collection with DigiKam* (↑Application Guide), Chapter 22, *Managing Your Digital Image Collection with F-Spot* (↑Application Guide) and Chapter 20, *Manipulating Graphics with The GIMP* (↑Application Guide).

16.3 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 185). 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.). For more information, see GNOME User Guide (↑GNOME User Guide) and KDE User Guide (↑KDE User Guide).

A more sophisticated synchronization solution is available with the program `opensync` (see packages `libopensync`, `msynctool` and the respective plug-ins for the different devices).

16.4 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 SUSE Linux Enterprise Desktop. 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.

17.1 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.

17.2 ACPI

ACPI (advanced configuration and power interface) was designed to enable the operating system to set up and control the individual hardware components. ACPI supersedes both PnP and APM. It delivers information about the battery, AC adapter, temperature, fan, and system events, like “close lid” or “battery low.”

The BIOS provides tables containing information about the individual components and hardware access methods. The operating system uses this information for tasks like assigning interrupts or activating and deactivating components. Because the operating system executes commands stored in the BIOS, the functionality depends on the BIOS implementation. The tables ACPI can detect and load are reported in `/var/log/boot.msg`. See [Section 17.2.3, “Troubleshooting”](#) (page 195) for more information about troubleshooting ACPI problems.

17.2.1 Controlling the CPU Performance

The CPU can save energy in three ways. Depending on the operating mode of the computer, these methods can be combined. Saving energy also means that the system heats up less and the fans are activated less frequently.

Frequency and Voltage Scaling

PowerNow! and Speedstep are the designations AMD and Intel use for this technology. However, this technology is also applied in processors of other manufacturers. The clock frequency of the CPU and its core voltage are reduced at the same time, resulting in more than linear energy savings. This means that when the frequency is halved (half performance), far less than half of the energy is consumed. This technology is independent from ACPI. There are two main approaches to performing CPU frequency scaling—by the kernel itself or by a userspace application. Therefore, there are different kernel governors that can be set below `/sys/devices/system/cpu/cpu*/cpufreq/`.

userspace governor

If the userspace governor is set, the kernel gives the control of CPU frequency scaling to a userspace application, usually a daemon. In SUSE Linux Enterprise Desktop distributions, this daemon is the `powersaved` package. When this implementation is used, the CPU frequency is adjusted in regard to the current system load. By default, one of the kernel implementations is used. However, on some hardware or in regard to specific processors or drivers, the userspace implementation is still the only working solution.

ondemand governor

This is the kernel implementation of a dynamic CPU frequency policy and should work on most systems. As soon as there is a high system load, the CPU frequency is immediately increased. It is lowered on a low system load.

conservative governor

This governor is similar to the on demand implementation, except that a more conservative policy is used. The load of the system must be high for a specific amount of time before the CPU frequency is increased.

powersave governor

The cpu frequency is statically set to the lowest possible.

performance governor

The cpu frequency is statically set to the highest possible.

Throttling the Clock Frequency

This technology omits a certain percentage of the clock signal impulses for the CPU. At 25% throttling, every fourth impulse is omitted. At 87.5%, only every eighth impulse reaches the processor. However, the energy savings are a little less than linear. Normally, throttling is only used if frequency scaling is not available or to maximize power savings. This technology, too, must be controlled by a special process. The system interface is `/proc/acpi/processor/*/throttling`.

Putting the Processor to Sleep

The operating system puts the processor to sleep whenever there is nothing to do. In this case, the operating system sends the CPU a `halt` command. There are three states: C1, C2, and C3. In the most economic state, C3, even the synchronization of the processor cache with the main memory is halted. Therefore, this state can only be applied if no other device modifies the contents of the main memory via bus master activity. Some drivers prevent the use of C3. The current state is displayed in `/proc/acpi/processor/*/power`.

Frequency scaling and throttling are only relevant if the processor is busy, because the most economic C state is applied anyway when the processor is idle. If the CPU is busy, frequency scaling is the recommended power saving method. Often the processor only works with a partial load. In this case, it can be run with a lower frequency. Usually, dynamic frequency scaling controlled by the kernel on demand governor or a daemon, such as `powersaved`, is the best approach. A static setting to a low frequency is useful for battery operation or if you want the computer to be cool or quiet.

Throttling should be used as the last resort, for example, to extend the battery operation time despite a high system load. However, some systems do not run smoothly when they are throttled too much. Moreover, CPU throttling does not make sense if the CPU has little to do.

17.2.2 ACPI Tools

The range of more or less comprehensive ACPI utilities includes tools that merely display information, like the battery charge level and the temperature (`acpi`, `klaptopdaemon`, etc.), tools that facilitate the access to the structures in `/proc/acpi` or that assist in

monitoring changes (akpi, acpiw, gtlacpiw), and tools for editing the ACPI tables in the BIOS (package `pmtools`).

17.2.3 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, however, the problems are caused by the BIOS. Sometimes, deviations from the ACPI specification are purposely integrated in the BIOS to circumvent errors in the ACPI implementation in other widespread operating systems. Hardware components that have serious errors in the ACPI implementation are recorded in a blacklist that prevents the Linux kernel from using ACPI for these components.

The first thing to do when problems are encountered is to update the BIOS. If the computer does not boot at all, one of the following boot parameters may be helpful:

`pci=noacpi`

Do not use ACPI for configuring the PCI devices.

`acpi=ht`

Only perform a simple resource configuration. Do not use ACPI for other purposes.

`acpi=off`

Disable ACPI.

WARNING: Problems Booting without ACPI

Some newer machines (especially SMP systems and AMD64 systems) need ACPI for configuring the hardware correctly. On these machines, disabling ACPI can cause problems.

Sometimes, the machine is confused by hardware that is attached over USB or FireWire. If a machine refuses to boot, unplug all unneeded hardware and try again.

Monitor the boot messages of the system with the command `dmesg | grep -2i acpi` (or all messages, because the problem may not be caused by ACPI) after booting. If an error occurs while parsing an ACPI table, the most important table—the DSDT—can be replaced with an improved version. In this case, the faulty DSDT of the

BIOS is ignored. The procedure is described in [Section 17.4, “Troubleshooting”](#) (page 198).

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://www.cpqlinux.com/acpi-howto.html> (detailed ACPI HOWTO, contains DSDT patches)
- <http://www.intel.com/technology/iapc/acpi/index.htm> (Advanced Configuration & Power Interface)
- <http://www.lesswatts.org/projects/acpi/> (the ACPI4Linux project at Sourceforge)
- <http://www.poupinou.org/acpi/> (DSDT patches by Bruno Ducrot)

17.3 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.

The `hdparm` application can be used to modify various hard disk settings. The option `-y` instantly switches the hard disk to the standby mode. `-Y` puts it to sleep. `hdparm -S x` causes the hard disk to be spun down after a certain period of inactivity. Replace `x` as follows: 0 disables this mechanism, causing the hard disk to run continuously.

Values from 1 to 240 are multiplied by 5 seconds. Values from 241 to 251 correspond to 1 to 11 times 30 minutes.

Internal power saving options of the hard disk can be controlled with the option `-B`. Select a value from 0 to 255 for maximum saving to maximum throughput. The result depends on the hard disk used and is difficult to assess. To make a hard disk quieter, use the option `-M`. Select a value from 128 to 254 for quiet to fast.

Often, it is not so easy to put the hard disk to sleep. In Linux, numerous processes write to the hard disk, waking it up repeatedly. Therefore, it is important to understand how Linux handles data that needs to be written to the hard disk. First, all data is buffered in the RAM. This buffer is monitored by the `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 and Ext3, 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. See `/usr/src/linux/Documentation/laptop-mode.txt` for details.

Another important factor is the way active programs behave. For example, good editors regularly write hidden backups of the currently modified file to the hard disk, causing the disk to wake up. Features like this can be disabled at the expense of data integrity.

In this connection, the mail daemon postfix makes use of the variable `POSTFIX_LAPTOP`. If this variable is set to `yes`, postfix accesses the hard disk far less frequently.

In SUSE Linux Enterprise Desktop these technologies are controlled by `laptop-mode-tools`.

17.4 Troubleshooting

All error messages and alerts are logged in the file `/var/log/messages`. If you cannot find the needed information, increase the verbosity of the messages of powersave using `DEBUG` in the file `/etc/sysconfig/powersave/common`. Increase the value of the variable to 7 or even 15 and restart the daemon. The more detailed error messages in `/var/log/messages` should help you to find the error. The following sections cover the most common problems with powersave and the different sleep modes.

17.4.1 ACPI Activated with Hardware Support but Functions Do Not Work

If you experience problems with ACPI, use the command `dmesg|grep -i acpi` to search the output of `dmesg` for ACPI-specific messages. A BIOS update may be required to resolve the problem. Go to the home page of your laptop manufacturer, look for an updated BIOS version, and install it. Ask the manufacturer to comply with the latest ACPI specification. If the errors persist after the BIOS update, proceed as follows to replace the faulty DSDT table in your BIOS with an updated DSDT:

- 1 Download the DSDT for your system from <http://acpi.sourceforge.net/dsdt/index.php>. Check if the file is decompressed and compiled as

shown by the file extension `.aml` (ACPI machine language). If this is the case, continue with step 3.

- 2 If the file extension of the downloaded table is `.asl` (ACPI source language), compile it with `iasl` (package `pmtools`). Enter the command `iasl -sa file.asl`.
- 3 Copy the file `DSDT.aml` to any location (`/etc/DSDT.aml` is recommended). Edit `/etc/sysconfig/kernel` and adapt the path to the DSDT file accordingly. Start `mkinitrd` (package `mkinitrd`). Whenever you install the kernel and use `mkinitrd` to create an `initrd`, the modified DSDT is integrated and loaded when the system is booted.

17.4.2 CPU Frequency Does Not Work

Refer to the kernel sources (`kernel-source`) to see if your processor is supported. You may need a special kernel module or module option to activate CPU frequency control. This information is available in `/usr/src/linux/Documentation/cpu-freq/*`.

17.4.3 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://www.opensuse.org/Pm-utils> and <http://www.opensuse.org/S2ram> to get more detailed information on how to modify the suspend and resume process.

17.5 For More Information

- <http://www.opensuse.org/S2ram>—How to get Suspend to RAM working
- <http://www.opensuse.org/Pm-utils>—How to modify the general suspend framework

Using Tablet PCs

SUSE® Linux Enterprise Desktop 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 Wacom devices, such as ACER TM C30x series, Fujitsu Lifebook T series (T30xx/T40xx/T50xx), Gateway C-140X/E-295C, HP Compaq TC1100/TC4200/TC4400, 2710p/2730p , IBM/Lenovo X41t/X61t, LG LT20, Motion M1200/M1400, OQO 02, Panasonic Toughbook CF-18, Toshiba Portege/Tecra M series, Satellite R15/R20.
- Tablet PCs with Wacom USB devices, such as ASUS R1E/R1F, Gateway C-120X/E-155C, HP Pavilion tx2000/tx2100/tx2500 series.
- Tablet PCs with FinePoint devices, such as Gateway C210X/M280E/CX2724, 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, dragging and dropping objects
- Using gesture recognition in applications of the X Window System
- Drawing with The GIMP
- Taking notes or sketching with applications like Jarnal or Xournal or editing larger amounts of text with Dasher

NOTE: Keyboard or Mouse Needed for Installation

During installation of SUSE Linux Enterprise Desktop, the pen cannot be used as an input device. If your Tablet PC does not feature a built-in keyboard or touch pad, connect an external keyboard or mouse to your Tablet PC for installation of your system.

18.1 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
- `wacom-kmp(-default)`: the kernel driver for Tablet PCs with USB Wacom devices

- `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
- `x11-input-wacom`: the X input module for Wacom tablets
- `x11-input-wacom-tools`: configuration, diagnostics, and libraries for Wacom tablets

If these packages are not installed, manually install the packages you need from command line or select the `TabletPC` pattern for installation in YaST.

18.2 Configuring Your Tablet Device

You can configure your Tablet PC (this does not include Tablet PCs with touch screens) during the installation process in the *Hardware Configuration* screen by changing the *Graphics Card* options. Alternatively you can configure the (internal or external) tablet device at any time after the installation.

- 1 Start SaX2 from the command line or by pressing `Alt + F2` and entering `sax2`.
- 2 If you use a Wacom or Finepoint device, click *Tablet* to show the *Tablet Properties*.

If you use a Tablet PC with a touch screen, click *Touchscreen* instead.

- 3 From the list on the right, select *TABLET PCs* as vendor, and the name of your tablet and check *Activate This Tablet*.

If your machine is not listed and you are sure that you have a Wacom device, select either *Wacom ISDV4 Tablet PC (SERIAL)* or *Wacom ISDV4 Tablet PC (USB)*.

- 4 Switch to the *Electronic Pens* tab and make sure the following options are activated: *Add Pen* and *Add Eraser*. If you have a Tablet PC with touch screen, also activate *Add Touch*.
- 5 Click *OK* to save the changes.

After finishing the X Window System configuration, restart your X server by logging out. Alternatively, leave the user interface and run `init 3 && init 5` in a virtual console.

After your tablet device has been configured, you can now make use of your pen (or, depending on your Tablet PC, your finger) as input device.

18.3 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 to 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 18.1 *xvkbd Virtual Keyboard*

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	Backspace	xvkbd (v3.0)					
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.

18.4 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 respective 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 18.7, “Troubleshooting”](#) (page 210) for more information.

18.5 Using Gesture Recognition

SUSE Linux Enterprise Desktop 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.

18.5.1 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 18.1 *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 the 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 grey. Repeat the gesture multiple times until the character changes its color to black. Untrained characters are shown on a light grey 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 windows shows a couple of empty cells in which to enter the gestures. The characters are not send 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 18.2 *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.

18.5.2 Using Xstroke

With xstroke, you can use gestures with your pen or other pointing devices as input for applications on the X Window System. The xstroke alphabet is a unistroke alphabet that resembles the Graffiti* alphabet. When activated, xstroke sends the input to the currently focused window.

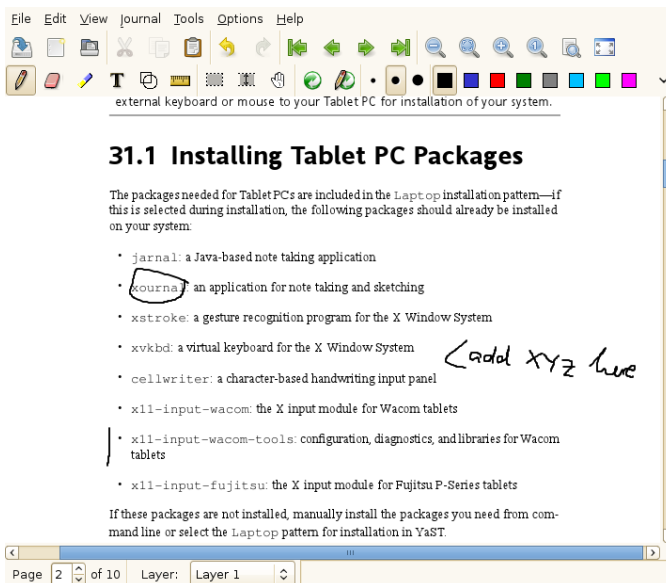
- 1 Start xstroke from the main menu or with `xstroke` from a shell. This adds a pencil icon to your system tray.
- 2 Start the application for which you want to create text input with the pen (for example, a terminal window, a text editor, or an OpenOffice.org Writer).
- 3 To activate the gesture recognition 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.

18.6 Taking Notes and Sketching with the Pen

To create drawings with the pen, you can use a professional graphics editor like The 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*.

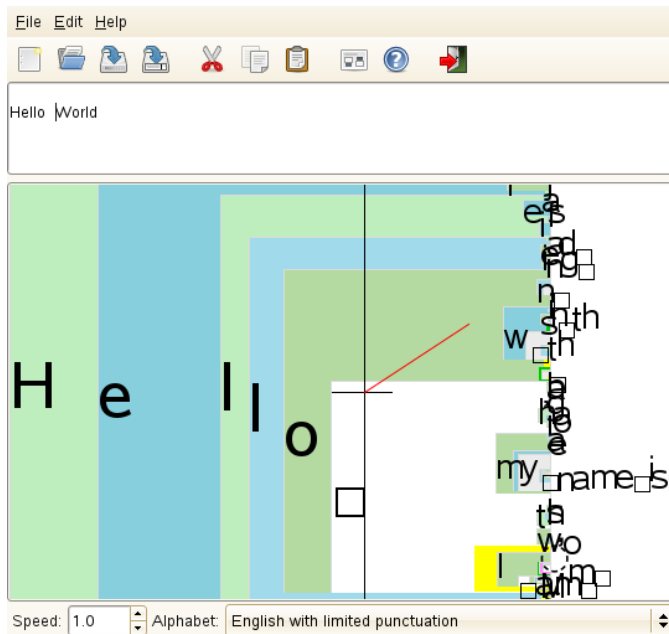
Figure 18.3 *Annotating a PDF with Xournal*



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 18.4 *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/>

18.7 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 `Ctrl + Alt + <`— 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 --output LVDS ---rotate normal && xsetwacom set "Mouse[7]" Rotate NONE
```

- For 90° rotation (clockwise, portrait):

```
xrandr --output LVDS ---rotate right && xsetwacom set "Mouse[7]" Rotate CW
```

- For 180° rotation (landscape):

```
xrandr --output LVDS --rotate inverted && xsetwacom set "Mouse[7]" Rotate HALF
```

- For 270° rotation (counterclockwise, portrait):

```
xrandr --output LVDS --rotate left && xsetwacom set "Mouse[7]" Rotate CCW
```

Note that the commands above depend on the contents of your `/etc/X11/xorg.conf` configuration file. If you have configured your device with SaX2 as described in [Section 18.2, “Configuring Your Tablet Device”](#) (page 203), the commands should work as they are written. If you have changed the `Identifier` of the tablet stylus input device in `xorg.conf` manually, replace `"Mouse[7]"` with the new `Identifier`. 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.

18.8 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/tcl1000/jarnal.htm#documentation>
- Find the xstroke man page at <http://davesource.com/Projects/xstroke/xstroke.txt>
- Find a HOWTO for configuring X on the Linux Wacom Web site: <http://linuxwacom.sourceforge.net/index.php/howto/x11>
- Find a very informative Web site about the Dasher project at <http://www.inference.phy.cam.ac.uk/dasher/>
- Find more information and documentation about CellWriter at <http://risujin.org/cellwriter/>
- Information on gnome-display-properties can be found at <http://en.opensuse.org/GNOME/Multiscreen>

Part IV. Services

Basic Networking

Linux offers the necessary networking tools and features for integration into all types of network structures. The customary Linux protocol, TCP/IP, has various services and special features, which are discussed here. Network access using a network card, modem, or other device can be configured with YaST. Manual configuration is also possible. 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 19.1, “Several Protocols in the TCP/IP Protocol Family”** (page 216) 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.”

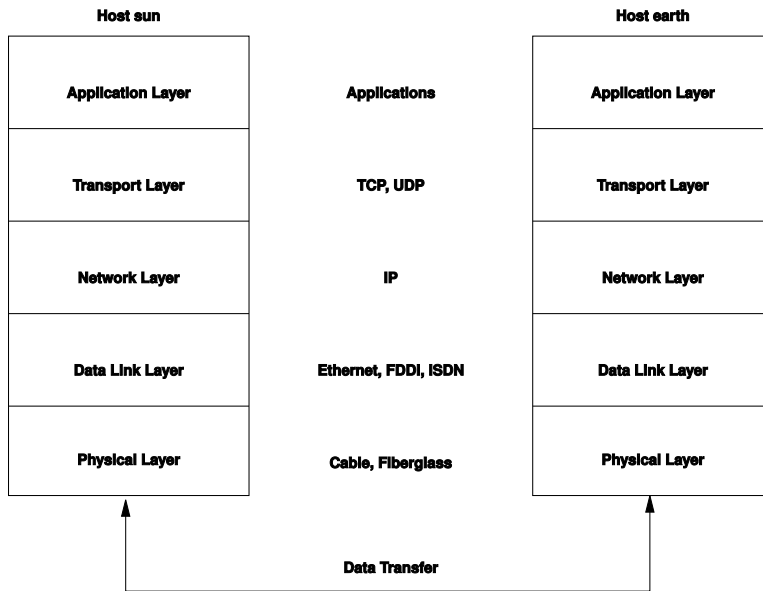
RFC stands for *Request for Comments*. RFCs are documents that describe various Internet protocols and implementation procedures for the operating system and its applications. The RFC documents describe the setup of Internet protocols. To expand your knowledge about any of the protocols, refer to the appropriate RFC documents. These are available at <http://www.ietf.org/rfc.html>.

Table 19.1 *Several Protocols in the TCP/IP Protocol Family*

Protocol	Description
TCP	Transmission Control Protocol: a connection-oriented secure protocol. The data to transmit is first sent by the application as a stream of data 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 during the transmission or the order of the data got mixed up. TCP is implemented wherever the data sequence matters.
UDP	User Datagram Protocol: a connectionless, insecure protocol. The data to transmit is sent in the form of packets generated by the application. The order in which the data arrives at the recipient is not guaranteed and data loss is possible. UDP is suitable for record-oriented applications. It features a smaller latency period than TCP.
ICMP	Internet Control Message Protocol: Essentially, this is not a protocol for the end user, but a special control protocol that issues error reports and can control the behavior of machines participating in TCP/IP data transfer. In addition, it provides a special echo mode that can be viewed using the program ping.
IGMP	Internet Group Management Protocol: This protocol controls machine behavior when implementing IP multicast.

As shown in [Figure 19.1, “Simplified Layer Model for TCP/IP”](#) (page 217), data exchange takes place in different layers. The actual network layer is the insecure data transfer via IP (Internet protocol). On top of IP, TCP (transmission control protocol) guarantees, to a certain extent, security of the data transfer. The IP layer is supported by the underlying hardware-dependent protocol, such as ethernet.

Figure 19.1 *Simplified Layer Model for TCP/IP*



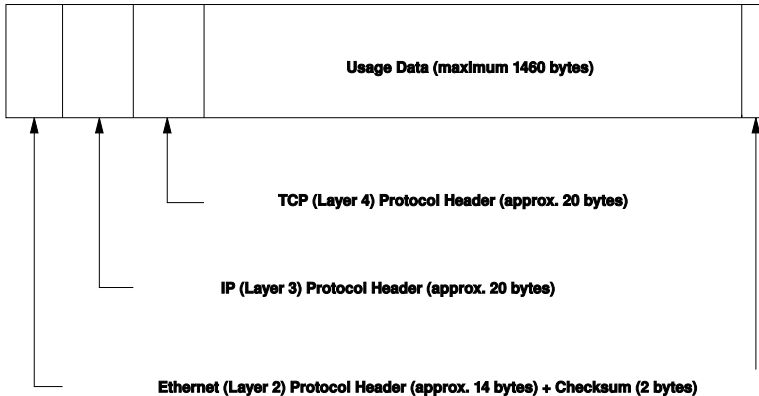
The diagram provides one or two examples for each layer. The layers are ordered according to *abstraction levels*. The lowest layer is very close to the hardware. The uppermost layer, however, is almost a complete abstraction from the hardware. Every layer has its own special function. The special functions of each layer are mostly implicit in their description. The data link and physical layers represent the physical network used, such as ethernet.

Almost all hardware protocols work on a packet-oriented basis. The data to transmit is packaged in *packets*, because it cannot be sent all at once. The maximum size of a TCP/IP packet is approximately 64 KB. Packets are normally quite a bit smaller, because the network hardware can be a limiting factor. The maximum size of a data packet on an ethernet is about fifteen hundred bytes. The size of a TCP/IP packet is limited to this amount when the data is sent over an ethernet. If more data is transferred, more data packets need to be sent by the operating system.

For the layers to serve their designated functions, additional information regarding each layer must be saved in the data packet. This takes place in the *header* of the packet. Every layer attaches a small block of data, called the protocol header, to the front of each emerging packet. A sample TCP/IP data packet traveling over an ethernet cable is illustrated in [Figure 19.2, "TCP/IP Ethernet Packet"](#) (page 218). The proof sum is

located at the end of the packet, not at the beginning. This simplifies things for the network hardware.

Figure 19.2 *TCP/IP Ethernet Packet*



When an application sends data over the network, the data passes through each layer, all implemented in the Linux kernel except the physical layer. Each layer is responsible for preparing the data so it can be passed to the next layer. The lowest layer is ultimately responsible for sending the data. The entire procedure is reversed when data is received. Like the layers of an onion, in each layer the protocol headers are removed from the transported data. Finally, the transport layer is responsible for making the data available for use by the applications at the destination. In this manner, one layer only communicates with the layer directly above or below it. For applications, it is irrelevant whether data is transmitted via a 100 Mbit/s FDDI network or via a 56-Kbit/s modem line. Likewise, it is irrelevant for the data line which kind of data is transmitted, as long as packets are in the correct format.

19.1 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 19.2, “IPv6—The Next Generation Internet”](#) (page 221).

19.1.1 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 19.1, “Writing IP Addresses”](#) (page 219).

Example 19.1 *Writing IP Addresses*

```
IP Address (binary): 11000000 10101000 00000000 00010100
IP Address (decimal): 192.      168.      0.      20
```

In decimal form, the four bytes are written in the decimal number system, separated by periods. The IP address is assigned to a host or a network interface. It cannot be used anywhere else in the world. There are exceptions to this rule, but these are not relevant in the following passages.

The points in IP addresses indicate the hierarchical system. Until the 1990s, IP addresses were strictly categorized in classes. However, this system has proven too inflexible and was discontinued. Now, *classless routing* (CIDR, classless interdomain routing) is used.

19.1.2 Netmasks and Routing

Netmasks are used to define the address range of a subnetwork. If two hosts are in the same subnetwork, they can reach each other directly, if they are not in the same subnetwork, they need the address of a gateway that handles all the traffic between the subnetwork and the rest of the world. To check if two IP addresses are in the same subnet, simply “AND” both addresses with the netmask. If the result is identical, both IP addresses are in the same local network. If there are differences, the remote IP address, and thus the remote interface, can only be reached over a gateway.

To understand how the netmask works, look at [Example 19.2, “Linking IP Addresses to the Netmask”](#) (page 220). The netmask consists of 32 bits that identify how much of an IP address belongs to the network. All those bits that are 1 mark the corresponding bit in the IP address as belonging to the network. All bits that are 0 mark bits inside the subnetwork. This means that the more bits are 1, the smaller the subnetwork is. Because the netmask always consists of several successive 1 bits, it is also possible to just count the number of bits in the netmask. In [Example 19.2, “Linking IP Addresses to the Netmask”](#) (page 220) the first net with 24 bits could also be written as 192.168.0.0/24.

Example 19.2 *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 19.2 *Specific Addresses*

Address Type	Description
Base Network Address	This is the netmask AND any address in the network, as shown in Example 19.2, “Linking IP Addresses to the Netmask” (page 220) under <code>Result</code> . This address cannot be assigned to any hosts.
Broadcast Address	This basically says, “Access all hosts in this subnetwork.” To generate this, the netmask is inverted in binary form and linked to the base network address with a logical OR. The above ex-

Address Type	Description
	ample therefore results in 192.168.0.255. This address cannot be assigned to any hosts.
Local Host	The address 127.0.0.1 is assigned to the “loopback device” on each host. A connection can be set up to your own machine with this address.

Because IP addresses must be unique all over the world, you cannot just select random addresses. There are three address domains to use if you want to set up a private IP-based network. These cannot get any connection from the rest of the Internet, because they cannot be transmitted over the Internet. These address domains are specified in RFC 1597 and listed in **Table 19.3, “Private IP Address Domains”** (page 221).

Table 19.3 *Private IP Address Domains*

Network/Netmask	Domain
10.0.0.0/255.0.0.0	10.x.x.x
172.16.0.0/255.240.0.0	172.16.x.x – 172.31.x.x
192.168.0.0/255.255.0.0	192.168.x.x

19.2 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.

19.2.1 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 19.2.2, “Address Types and Structure”](#) (page 224).

The following is a list of some other advantages of the new protocol:

Autoconfiguration

IPv6 makes the network “plug and play” capable, which means that a newly set up system integrates into the (local) network without any manual configuration. The new host uses its automatic configuration mechanism to derive its own address from the information made available by the neighboring routers, relying on a protocol called the *neighbor discovery* (ND) protocol. This method does not require

any intervention on the administrator's part and there is no need to maintain a central server for address allocation—an additional advantage over IPv4, where automatic address allocation requires a DHCP server.

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 19.2.3, “Coexistence of IPv4 and IPv6”](#) (page 228). Also, systems can rely on a *dual stack IP* technique to support both protocols at the same time, meaning that they have two network stacks that are completely separate, such that there is no interference between the two protocol versions.

Custom Tailored Services through Multicasting

With IPv4, some services, such as SMB, need to broadcast their packets to all hosts in the local network. IPv6 allows a much more fine-grained approach by enabling servers to address hosts through *multicasting*—by addressing a number of hosts as parts of a group (which is different from addressing all hosts through *broadcasting* or each host individually through *unicasting*). Which hosts are addressed as a group may depend on the concrete application. There are some predefined groups to address all name servers (the *all name servers multicast group*), for example, or all routers (the *all routers multicast group*).

19.2.2 Address Types and Structure

As mentioned, the current IP protocol is lacking in two important aspects: there is an increasing shortage of IP addresses and configuring the network and maintaining the routing tables is becoming a more complex and burdensome task. IPv6 solves the first problem by expanding the address space to 128 bits. The second one is countered by introducing a hierarchical address structure, combined with sophisticated techniques to allocate network addresses, as well as *multihoming* (the ability to assign several addresses to one device, giving access to several networks).

When dealing with IPv6, it is useful to know about three different types of addresses:

Unicast

Addresses of this type are associated with exactly one network interface. Packets with such an address are delivered to only one destination. Accordingly, unicast addresses are used to transfer packets to individual hosts on the local network or the Internet.

Multicast

Addresses of this type relate to a group of network interfaces. Packets with such an address are delivered to all destinations that belong to the group. Multicast addresses are mainly used by certain network services to communicate with certain groups of hosts in a well-directed manner.

Anycast

Addresses of this type are related to a group of interfaces. Packets with such an address are delivered to the member of the group that is closest to the sender, according to the principles of the underlying routing protocol. Anycast addresses are used to make it easier for hosts to find out about servers offering certain services in the given network area. All servers of the same type have the same anycast address. Whenever a host requests a service, it receives a reply from the server with the closest location, as determined by the routing protocol. If this server should fail for some reason, the protocol automatically selects the second closest server, then the third one, and so forth.

An IPv6 address is made up of eight four-digit fields, each representing 16 bits, written in hexadecimal notation. They are also separated by colons (:). Any leading zero bytes within a given field may be dropped, but zeros within the field or at its end may not. Another convention is that more than four consecutive zero bytes may be collapsed into a double colon. However, only one such :: is allowed per address. This kind of

shorthand notation is shown in [Example 19.3, “Sample IPv6 Address”](#) (page 225), where all three lines represent the same address.

Example 19.3 *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 19.4, “IPv6 Address Specifying the Prefix Length”](#) (page 225), contains the information that the first 64 bits form the network part of the address and the last 64 form its host part. In other words, the 64 means that the netmask is filled with 64 1-bit values from the left. Just like with IPv4, the IP address is combined with AND with the values from the netmask to determine whether the host is located in the same subnetwork or in another one.

Example 19.4 *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 19.4, “Various IPv6 Prefixes”](#) (page 225).

Table 19.4 *Various IPv6 Prefixes*

Prefix (hex)	Definition
00	IPv4 addresses and IPv4 over IPv6 compatibility addresses. These are used to maintain compatibility with IPv4. Their use still requires a router able to translate IPv6 packets into IPv4 packets. Several special addresses, such as the one for the loopback device, have this prefix as well.
2 or 3 as the first digit	Aggregatable global unicast addresses. As is the case with IPv4, an interface can be assigned to form part of a certain subnetwork. Currently, there are the following address spaces: 2001::/16 (production quality address space) and 2002::/16 (6to4 address space).

Prefix (hex)	Definition
<code>fe80::/10</code>	Link-local addresses. Addresses with this prefix should not be routed and should therefore only be reachable from within the same subnetwork.
<code>fec0::/10</code>	Site-local addresses. These may be routed, but only within the network of the organization to which they belong. In effect, they are the IPv6 equivalent of the current private network address space, such as <code>10.x.x.x</code> .
<code>ff</code>	These are multicast addresses.

A unicast address consists of three basic components:

Public Topology

The first part (which also contains one of the prefixes mentioned above) is used to route packets through the public Internet. It includes information about the company or institution that provides the Internet access.

Site Topology

The second part contains routing information about the subnetwork to which to deliver the packet.

Interface ID

The third part identifies the interface to which to deliver the packet. This also allows for the MAC to form part of the address. Given that the MAC is a globally unique, fixed identifier coded into the device by the hardware maker, the configuration procedure is substantially simplified. In fact, the first 64 address bits are consolidated to form the `EUI-64` token, with the last 48 bits taken from the MAC, and the remaining 24 bits containing special information about the token type. This also makes it possible to assign an `EUI-64` token to interfaces that do not have a MAC, such as those based on PPP or ISDN.

On top of this basic structure, IPv6 distinguishes between five different types of unicast addresses:

`::` (unspecified)

This address is used by the host as its source address when the interface is initialized for the first time—when the address cannot yet be determined by other means.

`::1` (loopback)

The address of the loopback device.

IPv4 Compatible Addresses

The IPv6 address is formed by the IPv4 address and a prefix consisting of 96 zero bits. This type of compatibility address is used for tunneling (see [Section 19.2.3, “Coexistence of IPv4 and IPv6”](#) (page 228)) to allow IPv4 and IPv6 hosts to communicate with others operating in a pure IPv4 environment.

IPv4 Addresses Mapped to IPv6

This type of address specifies a pure IPv4 address in IPv6 notation.

Local Addresses

There are two address types for local use:

link-local

This type of address can only be used in the local subnetwork. Packets with a source or target address of this type should not be routed to the Internet or other subnetworks. These addresses contain a special prefix (`fe80::/10`) and the interface ID of the network card, with the middle part consisting of zero bytes. Addresses of this type are used during automatic configuration to communicate with other hosts belonging to the same subnetwork.

site-local

Packets with this type of address may be routed to other subnetworks, but not to the wider Internet—they must remain inside the organization's own network. Such addresses are used for intranets and are an equivalent of the private address space defined by IPv4. They contain a special prefix (`fec0::/10`), the interface ID, and a 16 bit field specifying the subnetwork ID. Again, the rest is filled with zero bytes.

As a completely new feature introduced with IPv6, each network interface normally gets several IP addresses, with the advantage that several networks can be accessed through the same interface. One of these networks can be configured completely automatically using the MAC and a known prefix with the result that all hosts on the local network can be reached as soon as IPv6 is enabled (using the link-local address). With

the MAC forming part of it, any IP address used in the world is unique. The only variable parts of the address are those specifying the *site topology* and the *public topology*, depending on the actual network in which the host is currently operating.

For a host to go back and forth between different networks, it needs at least two addresses. One of them, the *home address*, not only contains the interface ID but also an identifier of the home network to which it normally belongs (and the corresponding prefix). The home address is a static address and, as such, it does not normally change. Still, all packets destined to the mobile host can be delivered to it, regardless of whether it operates in the home network or somewhere outside. This is made possible by the completely new features introduced with IPv6, such as *stateless autoconfiguration* and *neighbor discovery*. In addition to its home address, a mobile host gets one or more additional addresses that belong to the foreign networks where it is roaming. These are called *care-of* addresses. The home network has a facility that forwards any packets destined to the host when it is roaming outside. In an IPv6 environment, this task is performed by the *home agent*, which takes all packets destined to the home address and relays them through a tunnel. On the other hand, those packets destined to the care-of address are directly transferred to the mobile host without any special detours.

19.2.3 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 19.2.2, “Address Types and Structure”](#) (page 224)).

IPv6 hosts that are more or less isolated in the (worldwide) IPv4 network can communicate through tunnels: IPv6 packets are encapsulated as IPv4 packets to move them across an IPv4 network. Such a connection between two IPv4 hosts is called a *tunnel*. To achieve this, packets must include the IPv6 destination address (or the corresponding prefix) as well as the IPv4 address of the remote host at the receiving end of the tunnel. A basic tunnel can be configured manually according to an agreement between the hosts' administrators. This is also called *static tunneling*.

However, the configuration and maintenance of static tunnels is often too labor-intensive to use them for daily communication needs. Therefore, IPv6 provides for three different methods of *dynamic tunneling*:

6over4

IPv6 packets are automatically encapsulated as IPv4 packets and sent over an IPv4 network capable of multicasting. IPv6 is tricked into seeing the whole network (Internet) as a huge local area network (LAN). This makes it possible to determine the receiving end of the IPv4 tunnel automatically. However, this method does not scale very well and is also hampered by the fact that IP multicasting is far from widespread on the Internet. Therefore, it only provides a solution for smaller corporate or institutional networks where multicasting can be enabled. The specifications for this method are laid down in RFC 2529.

6to4

With this method, IPv4 addresses are automatically generated from IPv6 addresses, enabling isolated IPv6 hosts to communicate over an IPv4 network. However, a number of problems have been reported regarding the communication between those isolated IPv6 hosts and the Internet. The method is described in RFC 3056.

IPv6 Tunnel Broker

This method relies on special servers that provide dedicated tunnels for IPv6 hosts. It is described in RFC 3053.

19.2.4 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” (Chapter 3, *Installation with YaST*, ↑Deployment Guide). 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 IPv6 manually, enter `modprobe ipv6` as `root`.

Because of the autoconfiguration concept of IPv6, the network card is assigned an address in the *link-local* network. Normally, no routing table management takes place on a workstation. The network routers can be queried by the workstation, using the *router advertisement protocol*, for what prefix and gateways should be implemented. The `radvd` program can be used to set up an IPv6 router. This program informs the worksta-

tions 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.

19.2.5 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).

19.3 Name Resolution

DNS assists in assigning an IP address to one or more names and assigning a name to an IP address. In Linux, this conversion is usually carried out by a special type of software known as `bind`. The machine that takes care of this conversion is called a *name server*. The names make up a hierarchical system in which each name component is separated by dots. The name hierarchy is, however, independent of the IP address hierarchy described above.

Consider a complete name, such as `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 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 re-

quests. If you already use the `.local` domain in your nameserver configuration, you must switch this option off in `/etc/host.conf`. Also read the `host.conf` manual page.

If you want to switch off MDNS during installation, use `nomdns=1` as a boot parameter.

For more information on multicast DNS, see <http://www.multicastdns.org>.

19.4 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 19.6, “Configuring a Network Connection Manually”](#) (page 253).

On SUSE Linux Enterprise Desktop, where NetworkManager is active by default, all network cards are configured. 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 SUSE Linux Enterprise Desktop.

19.4.1 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*.

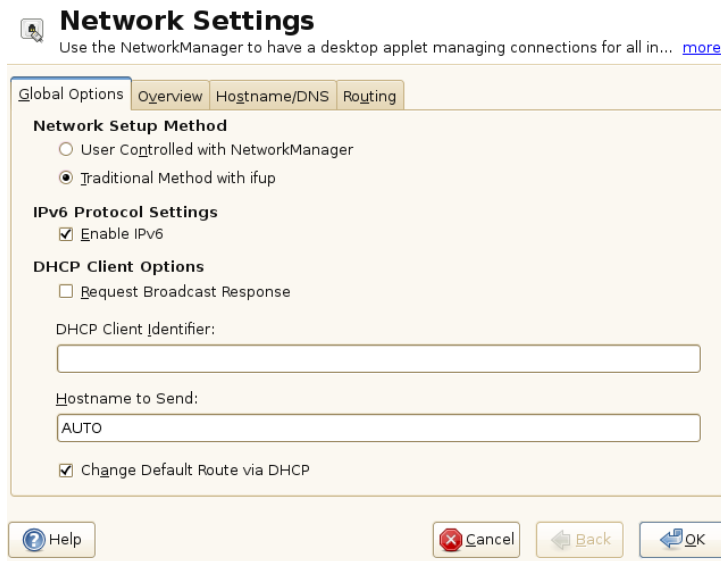
The *Global Options* tab allows to set general networking options such as the use of NetworkManager, IPv6 and general DHCP options. For more information, see [Section “Configuring Global Networking Options”](#) (page 233).

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 240). If you want to change the configuration of an already configured card, see [Section “Changing the Configuration of a Network Card”](#) (page 234).

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 241).

The *Routing* tab is used for the configuration of routing. See [Section “Configuring Routing”](#) (page 242) for more information.

Figure 19.3 *Configuring Network Settings*



Configuring Global Networking Options

The *Global Options* tab of the YaST *Network Settings* module allows 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 23, Using NetworkManager](#) (page 291).

In the *IPv6 Protocol Settings* choose whether you want to use the IPv6 protocol. It is possible to use IPv6 together with IPv4. By default, IPv6 is activated. However, in networks not using IPv6 protocol, response times can be faster with IPv6 protocol disabled. If you want to disable IPv6, uncheck the *Enable IPv6* option. This disables autoloading of the kernel module for IPv6. This will be applied after reboot.

In the *DHCP Client Options* configure options for the DHCP client. If you want the DHCP client to ask the server to always broadcast its responses, check *Request Broadcast Response*. It may be needed if your machine is moving between different networks. 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 20.1.2, “Configuration with YaST”](#) (page 275).

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* and/or *Zeroconf*.

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. On SUSE Linux Enterprise Desktop, 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. Specify whether the DHCP client should ask the server to always broadcast its responses in *Request Broadcast Response*. This option may be needed if your machine is a mobile client moving between networks. If you have a virtual host setup where different hosts communicate through the same interface, an *DHCP Client Identifier* is necessary to distinguish them.

DHCP is a good choice for client configuration but it is not ideal for server configuration. To set a static IP address, proceed as follows:

- 1 Select a card from the list of detected cards in the *Overview* tab of the YaST network card configuration module and click *Edit*.
- 2 In the *Address* tab, choose *Statically Assigned IP Address*.
- 3 Enter the *IP Address*. Both IPv4 and IPv6 addresses can be used. Enter the network mask in *Subnet Mask*. If the IPv6 address is used, use *Subnet Mask* for prefix length in format /64.

Optionally, you can enter a fully qualified *Hostname* for this address, which will be written to the `/etc/hosts` configuration file.

- 4 Click *Next*.
- 5 To activate the configuration, click *OK*.

If you use the static address, the name servers and default gateway are not configured automatically. To configure name servers, proceed as described in [Section “Configuring Hostname and DNS”](#) (page 241). To configure a gateway, proceed as described in [Section “Configuring Routing”](#) (page 242).

Configuring Aliases

One network device can have multiple IP addresses, called aliases. To set an alias for your network card, proceed as follows:

- 1 Select a card from the list of detected cards in the *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 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` or `KInternet`. Choose *Never* to not start the device at all. The *On NFSroot* is similar to *At Boot Time*, but the interface is 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 15, *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.
- 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 234).
- 5 If you selected *Wireless* as the device type of the interface, configure the wireless connection in the next dialog. Detailed information about wireless device configuration is available in [Section 20.1, “Wireless LAN”](#) (page 271).
- 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*.

If you are using DHCP to get an IP address, your hostname will be written to `/etc/hosts` by default and be resolvable as a `127.0.0.2` IP address. To disable this uncheck *Write Hostname to /etc/hosts* but note, that your hostname will not be resolvable without an 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 comma-separated list of interface names to be considered a valid source of settings. Except of complete interface names, also basic wildcards to match multiple interfaces are allowed. 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*. The default gateway matches every possible destination, but if any other entry exists that matches the required address, use this instead of the default route.
- 3 More entries can be entered in the *Routing Table*. Enter the *Destination* network IP address, *Gateway* IP address and the *Netmask*. Select the *Device* through which the traffic to the defined network will be routed (the minus sign stands for any

device). To omit any of these values, use the minus sign `-`. To enter a default gateway into the table, use `default` in the *Destination* field.

NOTE

If more default routes are used, it is possible to specify the metric option to determine which route has a higher priority. To specify the metric option, enter `- metric number` in *Options*. The route with the highest metric is used as default. If the network device is disconnected, its route will be removed and the next one will be used. However, the current kernel does not use metric in static routing, only routing daemons like `multipathd` do.

- 4 If the system is a router, enable the *IP Forwarding* option in the *Network Settings*.
- 5 To activate the configuration, click *OK*.

19.4.2 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 19.4 *Modem Configuration*

more'. The main area contains a 'Modem Device:' label followed by a text box containing '/dev/modem' and a dropdown arrow. Below that is a 'Dial Prefix (if needed):' label followed by an empty text box. There are two sections: 'Dial Mode' with radio buttons for 'Tone Dialing' (selected) and 'Pulse Dialing'; and 'Special Settings' with checkboxes for 'Speaker On' and 'Detect Dial Tone' (both checked). At the bottom center is a 'Details' button. At the very bottom are four buttons: 'Help' (with a question mark icon), 'Cancel' (with a red X icon), 'Back' (with a green left arrow icon), and 'Next' (with a green right arrow icon)." data-bbox="174 125 707 433"/>

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

[Details](#)

[Help](#) [Cancel](#) [Back](#) [Next](#)

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 option 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*.

19.4.3 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 19.5 *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 (EDSS1)*

☐ *1TR6*

☐ *Leased Line*

☐ *NI1*

Country: Code:

Area Code: Dial Prefix:

☒ Start ISDN Log

Activate device:

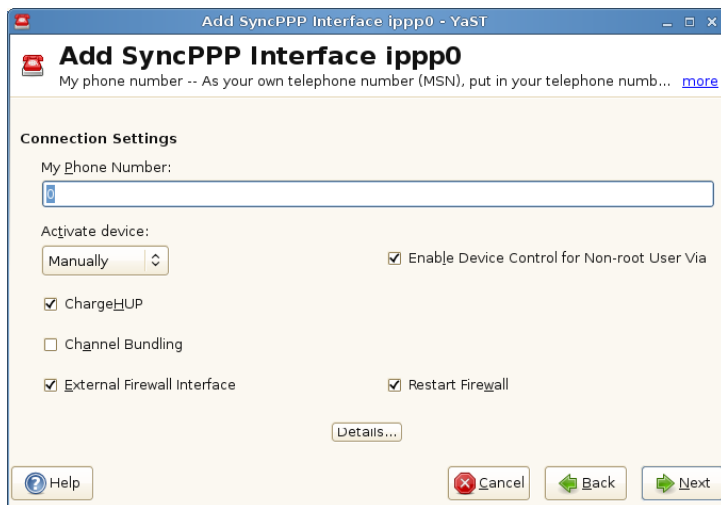
In the next dialog, shown in **Figure 19.5, “ISDN Configuration”** (page 246), select the protocol to use. The default is *Euro-ISDN (EDSS1)*, but for older or larger exchanges, select *1TR6*. If you are in the US, select *NI1*. Select your country in the relevant field. The corresponding country code then appears in the field next to it. Finally, provide your *Area Code* and the *Dial Prefix* if necessary. 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 19.6 *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 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, make IP address settings. If you have not been given a static IP by your provider, select *Dynamic IP Address*. Otherwise, use the fields provided to enter your host's local IP address and the remote IP address according to the specifications of your ISP. If the interface should be the default route to the Internet, select *Default Route*. Each host can only have one interface configured as the default route. Leave this dialog by selecting *Next*.

The following dialog allows you to set your country and select an ISP. The ISPs included in the list are call-by-call providers only. If your ISP is not in the list, select *New*. This opens the *Provider Parameters* dialog in which to enter all the details for your ISP. When entering the phone number, do not include any blanks or commas among the digits. Finally, enter your login and the password as provided by the ISP. When finished, select *Next*.

To use *Dial on Demand* on a stand-alone workstation, also specify the name server (DNS server). Most ISPs support dynamic DNS, which means the IP address of a name server is sent by the ISP each time you connect. For a single workstation, however, you still need to provide a placeholder address like 192.168.22.99. If your ISP does not support dynamic DNS, specify the name server IP addresses of the ISP. If desired,

specify a time-out for the connection—the period of network inactivity (in seconds) after which the connection should be automatically terminated. Confirm your settings with *Next*. YaST displays a summary of the configured interfaces. To activate these settings, select *OK*.

19.4.4 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.

For further information about the configuration of cable modems, read the Support Database article on the topic, which is available online at http://en.opensuse.org/SDB:Setting_Up_an_Internet_Connection_via_Cable_Modem_with_SuSE_Linux_8.0_or_Higher.

19.4.5 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 has already been set up in the correct way. If you have not done so yet, first configure the card by selecting *Configure Network Cards* (see [Section 19.4.1, “Configuring the Network Card with YaST”](#) (page 232)). 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.

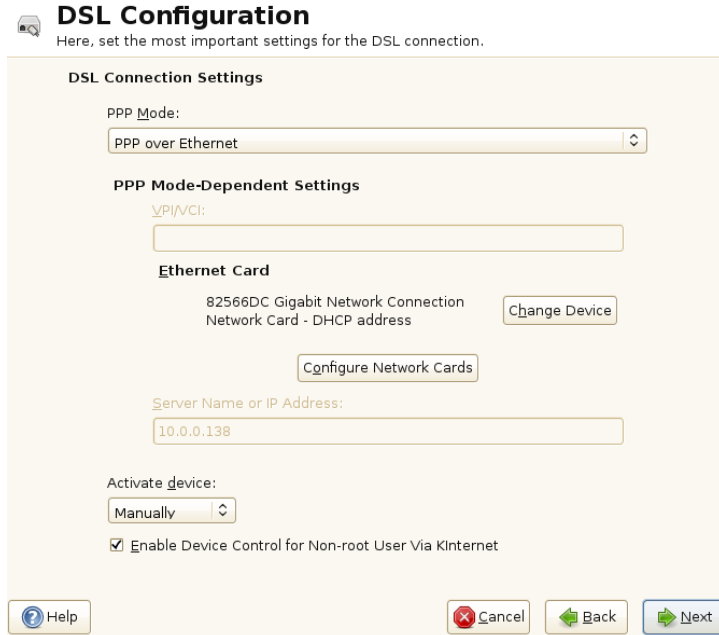
TIP

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 19.7, “DSL Configuration”](#) (page 251)), 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 19.7 DSL Configuration



DSL Configuration
Here, set the most important settings for the DSL connection.

DSL Connection Settings

PPP Mode:
PPP over Ethernet

PPP Mode-Dependent Settings

VPI/VCI:
[Empty text box]

Ethernet Card
82566DC Gigabit Network Connection
Network Card - DHCP address [Change Device]

[Configure Network Cards]

Server Name or IP Address:
10.0.0.138

Activate device:
Manually

☒ Enable Device Control for Non-root User Via KInternet

[Help] [Cancel] [Back] [Next]

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.

19.5 NetworkManager

NetworkManager is the ideal solution for a mobile workstation. With NetworkManager, you do not need to worry about configuring network interfaces and switching between networks when you are moving. NetworkManager can automatically connect to known WLAN networks. If you have two or more connection possibilities, it can connect to the faster one.

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 23.2, “Enabling NetworkManager”](#) (page 292) 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 23, Using NetworkManager](#) (page 291).

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, but it 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 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.

19.6 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. The problem is solved by `udev`. The `udev` persistent net generator (`/etc/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 19.5, “Manual Network Configuration Scripts” (page 254) summarizes the most important scripts involved in the network configuration.

Table 19.5 *Manual Network Configuration Scripts*

Command	Function
<code>if{up,down,status}</code>	The <code>if*</code> scripts start, stop network interfaces, or return the status of the specified interface. More information is available in the manual page of <code>ifup</code> .
<code>rcnetwork</code>	The <code>rcnetwork</code> script can be used to start, stop, or restart all network interfaces or just a specified one. Use <code>rcnetwork stop</code> to stop, <code>rcnetwork start</code> to start, and <code>rcnetwork restart</code> to restart network interfaces. If you want to stop, start or restart just one interface, use the command followed by the interface name, for example <code>rcnetwork restart eth0</code> . The <code>rcnetwork status</code> command displays the state of the interfaces, their IP addresses, and whether a DHCP client is running. With <code>rcnetwork stop-all-dhcp-clients</code> and <code>rcnetwork restart-all-dhcp-clients</code> you can stop or restart DHCP clients running on network interfaces.

More information about `udev` and persistent device names is available in [Chapter 13, *Dynamic Kernel Device Management with udev*](#) (page 145).

19.6.1 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, all variables from the files `dhcp`, `wireless`, and `config` can be used in the `ifcfg-*` files if a general setting should be used for only one interface.

/etc/sysconfig/network/{config, dhcp, wireless}

The file `config` contains general settings for the behavior of `ifup`, `ifdown`, and `ifstatus`. `dhcp` contains settings for DHCP and `wireless` for wireless LAN cards. The variables in all three configuration files are commented. Some of the variables from `/etc/sysconfig/network/config` can also be used in `ifcfg-*` files, where they are treated with 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,ifroute-*}

The static routing of TCP/IP packets is determined here. All the static routes required by the various system tasks can be entered in the `/etc/sysconfig/network/routes` file: routes to a host, routes to a host via a gateway, and routes to a network. For each interface that needs individual routing, define an additional configuration file: `/etc/sysconfig/network/ifroute-*`. Replace `*` with the name of the interface. The entries in the routing configuration files look like this:

# Destination	Dummy/Gateway	Netmask	Device
#			
127.0.0.0	0.0.0.0	255.255.255.0	lo
204.127.235.0	0.0.0.0	255.255.255.0	eth0
default	204.127.235.41	0.0.0.0	eth0
207.68.156.51	207.68.145.45	255.255.255.255	eth1
192.168.0.0	207.68.156.51	255.255.0.0	eth1

The route's destination is in the first column. This column may contain the IP address of a network or host or, in the case of *reachable* name servers, the fully qualified network or hostname.

The second column contains the default gateway or a gateway through which a host or network can be accessed. The third column contains the netmask for networks or hosts behind a gateway. For example, the mask is `255.255.255.255` for a host behind a gateway.

The fourth column is only relevant for networks connected to the local host such as loopback, Ethernet, ISDN, PPP, and dummy device. The device name must be entered here.

An (optional) fifth column can be used to specify the type of a route. Columns that are not needed should contain a minus sign – to ensure that the parser correctly interprets the command. For details, refer to the `routes(5)` man page.

/etc/resolv.conf

The domain to which the host belongs is specified in this file (keyword `search`). Also listed is the status of the name server address to access (keyword `nameserver`). Multiple domain names can be specified 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 with `#` signs. **Example 19.5**, “`/etc/resolv.conf`” (page 256) 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 19.5 */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 changed.

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 19.6, “/etc/hosts”](#) (page 258), 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 19.6 */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 19.7, “/etc/networks”](#) (page 258).

Example 19.7 */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 19.6,](#)

“Parameters for `/etc/host.conf`” (page 259) shows the parameters available. A sample `/etc/host.conf` is shown in [Example 19.8](#), “`/etc/host.conf`” (page 259).

Table 19.6 *Parameters for `/etc/host.conf`*

<code>order hosts, bind</code>	Specifies in which order the services are accessed for the name resolution. Available arguments are (separated by blank spaces or commas): <code>hosts</code> : searches the <code>/etc/hosts</code> file <code>bind</code> : accesses a name server <code>nis</code> : uses NIS
<code>multi on/off</code>	Defines if a host entered in <code>/etc/hosts</code> can have multiple IP addresses.
<code>nospoof on</code> <code>spoofalert on/off</code>	These parameters influence the name server <i>spoofing</i> but do not exert any influence on the network configuration.
<code>trim domainname</code>	The specified domain name is separated from the hostname after hostname resolution (as long as the hostname includes the domain name). This option is useful if only names from the local domain are in the <code>/etc/hosts</code> file, but should still be recognized with the attached domain names.

Example 19.8 *`/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 19.9, “/etc/nsswitch.conf”](#) (page 260). Comments are introduced by `#` signs. In this example, the entry under the `hosts` database means that a request is sent to `/etc/hosts` (files) via DNS.

Example 19.9 */etc/nsswitch.conf*

```
passwd:      compat
group:       compat

hosts:       files dns
networks:    files dns

services:    db files
protocols:   db files

netgroup:    files
automount:   files nis
```

The “databases” available over NSS are listed in [Table 19.7, “Databases Available via /etc/nsswitch.conf”](#) (page 260). In addition, `automount`, `bootparams`, `netmasks`, and `publickey` are expected in the near future. The configuration options for NSS databases are listed in [Table 19.8, “Configuration Options for NSS Databases”](#) (page 261).

Table 19.7 *Databases Available via /etc/nsswitch.conf*

<code>aliases</code>	Mail aliases implemented by <code>sendmail</code> ; see <code>man 5 aliases</code> .
<code>ethers</code>	Ethernet addresses.
<code>group</code>	For user groups used by <code>getgrent</code> . See also the man page for <code>group</code> .
<code>hosts</code>	For hostnames and IP addresses, used by <code>gethostbyname</code> and similar functions.
<code>netgroup</code>	Valid host and user lists in the network for the purpose of controlling access permissions; see the <code>netgroup(5)</code> man page.
<code>networks</code>	Network names and addresses, used by <code>getnetent</code> .

<code>passwd</code>	User passwords, used by <code>getpwent</code> ; see the <code>passwd(5)</code> man page.
<code>protocols</code>	Network protocols, used by <code>getprotoent</code> ; see the <code>protocols(5)</code> man page.
<code>rpc</code>	Remote procedure call names and addresses, used by <code>getrpcbyname</code> and similar functions.
<code>services</code>	Network services, used by <code>getservent</code> .
<code>shadow</code>	Shadow passwords of users, used by <code>getspnam</code> ; see the <code>shadow(5)</code> man page.

Table 19.8 *Configuration Options for NSS “Databases”*

<code>files</code>	directly access files, for example, <code>/etc/aliases</code>
<code>db</code>	access via a database
<code>nis, nisplus</code>	NIS, see also Chapter 3, <i>Using NIS</i> (↑Security Guide)
<code>dns</code>	can only be used as an extension for <code>hosts</code> and <code>networks</code>
<code>compat</code>	can only be used as an extension for <code>passwd</code> , <code>shadow</code> , and <code>group</code>

/etc/nscd.conf

This file is used to configure `nscd` (name service cache daemon). See 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 mecha-

nism in `nsd` 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 hostname without the domain name attached. This file is read by several scripts while the machine is booting. It may only contain one line in which the hostname is set.

19.6.2 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 routing, network devices, policy routing, and tunnels. It was designed as a replacement for the older tools `ifconfig` and `route`.

`ip` is a very complex tool. Its common syntax is `ip options object command`. You can work with the following objects:

`link`

This object represents a network device.

`address`

This object represents the IP address of device.

neighbour

This object represents a ARP or NDISC cache entry.

route

This object represents the routing table entry.

rule

This object represents a rule in the routing policy database.

maddress

This object represents a multicast address.

mrout

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` objects. If, for example, you want to read help for `ip addr`, enter `ip addr help`. Find the `ip` manual in `/usr/share/doc/packages/iproute2/ip-cref.pdf`.

Testing a Connection with ping

The `ping` command is the standard tool for testing whether a TCP/IP connection works. It uses the ICMP protocol to send a small data packet, `ECHO_REQUEST` datagram, to the destination host, requesting an immediate reply. If this works, `ping` displays a message to that effect, which indicates that the network link is basically functioning.

`ping` does more than test only the function of the connection between two computers: it also provides some basic information about the quality of the connection. In [Example 19.10, “Output of the Command ping”](#) (page 264), you can see an example of the `ping` output. The second-to-last line contains information about number of transmitted packets, packet loss, and total time of `ping` running.

As the destination, you can use a hostname or IP address, for example, `ping example.com` or `ping 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 19.10 *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 option `-i`. For example to increase `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.

Configuring the Network with ifconfig

`ifconfig` is a traditional network configuration tool. In contrast to `ip`, you can use it only for interface configuration. If you want to configure routing, use `route`.

NOTE: ifconfig and ip

The program `ifconfig` is obsolete. Use `ip` instead.

Without arguments, `ifconfig` displays the status of the currently active interfaces. As you can see in [Example 19.11, “Output of the ifconfig Command”](#) (page 266), `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 19.11 *Output of the ifconfig Command*

```
eth0      Link encap:Ethernet  HWaddr 00:08:74:98:ED:51
          inet6 addr: fe80::208:74ff:fe98:ed51/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:634735 errors:0 dropped:0 overruns:4 frame:0
          TX packets:154779 errors:0 dropped:0 overruns:0 carrier:1
          collisions:0 txqueuelen:1000
          RX bytes:162531992 (155.0 Mb)  TX bytes:49575995 (47.2 Mb)
          Interrupt:11 Base address:0xec80

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8559 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8559 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:533234 (520.7 Kb)  TX bytes:533234 (520.7 Kb)

wlan1     Link encap:Ethernet  HWaddr 00:0E:2E:52:3B:1D
          inet addr:192.168.2.4  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::20e:2eff:fe52:3b1d/64 Scope:Link
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:50828 errors:0 dropped:0 overruns:0 frame:0
          TX packets:43770 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:45978185 (43.8 Mb)  TX bytes:7526693 (7.1 MB)
```

For more options and information about using `ifconfig`, enter `ifconfig -h` or see the `ifconfig (8)` man page.

Configuring Routing with route

`route` is a program for manipulating the IP routing table. You can use it to view your routing configuration and add or remove of routes.

NOTE: route and ip

The program `route` is obsolete. Use `ip` instead.

`route` is especially useful if you need quick and comprehensible information about your routing configuration to determine problems with routing. To view your current routing configuration, enter `route -n` as `root`.

Example 19.12 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.

19.6.3 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 19.9, “Some Start-Up Scripts for Network Programs”](#) (page 267).

Table 19.9 *Some Start-Up Scripts for Network Programs*

<code>/etc/init.d/network</code>	This script handles the configuration of the network interfaces. If the <code>network</code> service was not started, no network interfaces are implemented.
<code>/etc/init.d/xinetd</code>	Starts xinetd. xinetd can be used to make server services available on the system. For example, it can start vsftpd whenever an FTP connection is initiated.
<code>/etc/init.d/portmap</code>	Starts the portmapper needed for the RPC server, such as an NFS server.
<code>/etc/init.d/nfsserver</code>	Starts the NFS server.
<code>/etc/init.d/postfix</code>	Controls the postfix process.
<code>/etc/init.d/ypserv</code>	Starts the NIS server.

19.7 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 `ipppd` or `pppd`. Basically, all that needs to be done to go online is to start these programs correctly.

If you have a flat-rate connection that does not generate any additional costs for the dial-up connection, simply start the respective daemon. Control the dial-up connection with a KDE applet or a command-line interface. If the Internet gateway is not the host you are using, you might want to control the dial-up connection by way of a network host.

This is where `smpppd` is involved. It provides a uniform interface for auxiliary programs and acts in two directions. First, it programs the required `pppd` or `ipppd` and controls its dial-up properties. Second, it makes various providers available to the user programs and transmits information about the current status of the connection. As `smpppd` can also be controlled by way of the network, it is suitable for controlling dial-up connections to the Internet from a workstation in a private subnetwork.

19.7.1 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, this option must be set to `yes`. The port on which `smpppd` listens is `3185`. If this parameter is set to `yes`, the parameters `bind-address`, `host-range`, and `password` should also be set accordingly.

`bind-address = ip address`

If a host has several IP addresses, use this parameter to determine at which IP address smpppd should accept connections. The default is to listen at all addresses.

`host-range = min 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.

19.7.2 Configuring KInternet and cinternet for Remote Use

KInternet and cinternet can be used to control a local or remote smpppd. cinternet is the command-line counterpart of the graphical KInternet. To prepare these utilities for use with a remote smpppd, edit the configuration file `/etc/smpppd-c.conf` manually or using KInternet. This file only uses four options:

`sites = list of sites`

Here, tell the front-ends where to search for smpppd. The front-ends test the options in the order specified here. The `local` option orders the establishment of a connection to the local smpppd. The `gateway` option points to an smpppd on the gateway. The `config-file` indicates, that the connection should be established to the smpppd specified in the `server` and `port` options in the `/etc/smpppd-c.conf` file. `slp` orders the front-ends to connect to an smpppd found via SLP.

`server = server`

Here, specify the host on which smpppd runs.

`port = port`

Here, specify the port on which smpppd runs.

`password = password`

Insert the password selected for smpppd.

If smpppd is active, you can now try to access it, for example, with `cinternet --verbose --interface-list`. If you experience difficulties at this point, refer to the `smpppd-c.conf(5)` and `cinternet(8)` man pages.

Wireless Communication

There are several possibilities for using your Linux system to communicate with other computers, cellular phones, or peripheral devices. WLAN (wireless LAN) can be used to network laptops. Bluetooth can be used to connect individual system components (mouse, keyboard), peripheral devices, cellular phones, PDAs, and individual computers with each other. IrDA is mostly used for communication with PDAs or cellular phones. Universal Mobile Telecommunications System (UMTS), also known as 3G, can offer several multimedia services such as browsing the Web or sending and receiving messages. This chapter introduces these technologies and their configuration.

20.1 Wireless LAN

Wireless LANs have become an indispensable aspect of mobile computing. Today, most laptops have built-in WLAN cards. The 802.11 standard for the wireless communication of WLAN cards was prepared by the IEEE organization. Originally, this standard provided for a maximum transmission rate of 2 Mbit/s. Meanwhile, several supplements have been added to increase the data rate. These supplements define details such as the modulation, transmission output, and transmission rates (see [Table 20.1, “Overview of Various WLAN Standards”](#) (page 272)). Additionally, a lot of companies implement hardware with proprietary or draft features.

Table 20.1 *Overview of Various WLAN Standards*

Name	Band (GHz)	Maximum Transmission Rate (Mbit/s)	Note
802.11 Legacy	2.4	2	Outdated; virtually no end devices available
802.11a	5	54	Less interference-prone
802.11b	2.4	11	Less common
802.11g	2.4	54	Widespread, backwards-compatible with 11b
802.11n draft	2.4 and/or 5	300	Common

802.11 Legacy cards are not supported by SUSE® Linux Enterprise Desktop. Most cards using 802.11a, 802.11b, 802.11g and 802.11n draft are supported. New cards usually comply with the 802.11n draft standard, but cards using 802.11g are still available.

20.1.1 Function

In wireless networking, various techniques and configurations are used to ensure fast, high-quality, and secure connections. Different operating types suit different setups. It can be difficult to choose the right authentication method. The available encryption methods have different advantages and pitfalls.

Basically, wireless networks can be classified as managed networks and ad-hoc networks. Managed networks have a managing element: the access point. In this mode (also referred to as infrastructure mode), all connections of the WLAN stations in the network run over the access point, which may also serve as a connection to an ethernet. Ad-hoc networks do not have an access point. The stations communicate directly with each other, 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. Therefore, an access point

is usually used. It is even possible to use a WLAN card as an access point. Some cards support this functionality.

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. However, because WEP has proven to be insecure (see [Section “Security”](#) (page 279)), the WLAN industry (joined under the name *Wi-Fi Alliance*) has defined a new extension called WPA, which is supposed to eliminate the weaknesses of WEP. The later IEEE 802.11i standard (also referred to as WPA2, because WPA is based on a draft version 802.11i) includes WPA and some other authentication and encryption methods.

To make sure that only authorized stations can connect, various authentication mechanisms are used in managed networks:

Open

An open system is a system that does not require authentication. Any station can join the network. Nevertheless, WEP encryption (see [Section “Encryption”](#) (page 274)) can be used.

Shared Key (according to IEEE 802.11)

In this procedure, the WEP key is used for the authentication. However, this procedure is not recommended, because it makes the WEP key more susceptible to attacks. All an attacker needs to do is to listen long enough to the communication between the station and the access point. During the authentication process, both sides exchange the same information, once in encrypted form and once in unencrypted form. This makes it possible for the key to be reconstructed with suitable tools. Because this method makes use of the WEP key for the authentication and for the encryption, it does not enhance the security of the network. A station that has the correct WEP key can authenticate, encrypt, and decrypt. A station that does not have the key cannot decrypt received packets. Accordingly, it cannot communicate, regardless of whether it had to authenticate itself.

WPA-PSK (according to IEEE 802.1x)

WPA-PSK (PSK stands for preshared key) works similarly to the Shared Key procedure. All participating stations as well as the access point need the same key. The key is 256 bits in length and is usually entered as a passphrase. This system

does not need a complex key management like WPA-EAP and is more suitable for private use. Therefore, WPA-PSK is sometimes referred to as WPA “Home”.

WPA-EAP (according to IEEE 802.1x)

Actually, WPA-EAP is not an authentication system but a protocol for transporting authentication information. WPA-EAP is used to protect wireless networks in 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: TLS (Transport Layer Security), TTLS (Tunneled Transport Layer Security), and PEAP (Protected Extensible Authentication Protocol). In a nutshell, these options work as follows:

EAP-TLS

TLS authentication relies on the mutual exchange of certificates both for server and client. First, the server presents its certificate to the client where it is evaluated. If the certificate is considered valid, the client in turn presents its certificate to the server. While TLS is secure, it requires a working certification management infrastructure in your network. This infrastructure is rarely found in private networks.

EAP-TTLS and PEAP

Both TTLS and PEAP are two-stage protocols. In the first stage, a secure connection is established and in the second one the client authentication data is exchanged. They require far less certification management overhead than TLS, if any.

Encryption

There are various encryption methods to ensure that no unauthorized person can read the data packets that are exchanged in a wireless network or gain access to the network:

WEP (defined in IEEE 802.11)

This standard makes use of the RC4 encryption algorithm, originally with a key length of 40 bits, later also with 104 bits. Often, the length is declared as 64 bits or 128 bits, depending on whether the 24 bits of the initialization vector are included. However, this standard has some weaknesses. Attacks against the keys generated by this system may be successful. Nevertheless, it is better to use WEP than not encrypt the network at all.

Some vendors have implemented the non-standard “Dynamic WEP”. It works exactly as WEP and shares the same weaknesses, except the fact 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 in vain. TKIP is used together with WPA-PSK.

CCMP (defined in IEEE 802.11i)

CCMP describes the key management. Usually, it is used in connection with WPA-EAP, but it can also be used with WPA-PSK. The encryption takes place according to AES and is stronger than the RC4 encryption of the WEP standard.

20.1.2 Configuration with YaST

To configure the wireless network card, select *Network Devices > Network Settings* in the YaST control center. The Network Settings dialog where you can configure general network settings opens. Please refer to [Section 19.4, “Configuring a Network Connection with YaST”](#) (page 232) for more information about the general network configuration. All network cards that have been detected by the system are listed under the *Overview* tab.

Choose your wireless card from the list and click *Edit* to open the Network Card Setup dialog. Configure whether to use a dynamic or a static IP address under the tab *Address*. You can also adjust *General* and *Hardware* settings such as *Device Activation* or *Firewall Zone* and driver settings. In most cases there is no need to change the preconfigured values.

Click *Next* to proceed to the wireless network card specific configuration dialog. If you are using NetworkManager (refer to [Section 19.5, “NetworkManager”](#) (page 252) for more information), there is no need to adjust the wireless device settings, since these will be set by NetworkManager on demand—proceed with *Next* and *Yes* to finish the configuration. If you are using your computer only in a specific wireless network, make the basic settings for WLAN operation here.

Figure 20.1 *YaST: Configuring the Wireless Network Card*

more'. The main content area is titled 'Wireless Device Settings' and contains several configuration options: 'Operating Mode' with a dropdown menu set to 'Managed'; 'Network Name (ESSID)' with a dropdown menu and a 'Scan Network' button; 'Authentication Mode' with a dropdown menu set to 'WEP - Open'; 'Key Input Type' with three radio buttons: 'Passphrase' (selected), 'ASCII', and 'Hexadecimal'; and 'Encryption Key' with a text input field. At the bottom of the main area are two buttons: 'Expert Settings' and 'WEP Keys'. The footer of the window contains four buttons: 'Help', 'Abort', 'Back', and 'Next'."/>

Wireless Network Card Configuration
Here, set the most important settings for wireless networking. [more](#)

Wireless Device Settings

Operating Mode:
Managed

Network Name (ESSID):
Scan Network

Authentication Mode:
WEP - Open

Key Input Type
☒ Passphrase ☐ ASCII ☐ Hexadecimal

Encryption Key:

Expert Settings WEP Keys

Help Abort Back Next

Operating Mode

A station can be integrated in a WLAN in three different modes. The suitable mode depends on the network in which to communicate: *Ad-hoc* (peer-to-peer network without access point), *Managed* (network is managed by an access point), or *Master* (your network card should be used as the access point). To use any of the WPA-PSK or WPA-EAP modes, the operating mode must be set to *Managed*.

Network Name (ESSID)

All stations in a wireless network need the same ESSID for communicating with each other. If nothing is specified, the card may automatically selects an access point, which may not be the one you intended to use. Use *Scan Network* for a list of available wireless networks.

Authentication Mode

Select a suitable authentication method for your network: *No Encryption*, *WEP-Open*, *WEP-Shared Key*, *WPA-EAP*, or *WPA-PSK*. If you select WPA authentication, a network name (ESSID) must be set.

Key Input Type

WEP and WPA-PSK authentication methods require to input a key. The key has to be entered as either a *Passphrase*, as an *ASCII* string, or *Hexadecimal* string.

WEP Keys

Either enter the default key here or click *WEP Keys* to enter the advanced key configuration dialog. Set the length of the key to *128 bit* or *64 bit*. The default setting is *128 bit*. In the list area at the bottom of the dialog, up to four different keys can be specified for your station to use for the encryption. Press *Set as Default* to define one of them as the default key. Unless you change this, YaST uses the first entered key as the default key. If the standard key is deleted, one of the other keys must be marked manually as the default key. Click *Edit* to modify existing list entries or create new keys. In this case, a pop-up window prompts you to select an input type (*Passphrase*, *ASCII*, or *Hexadecimal*). If you select *Passphrase*, enter a word or a character string from which a key is generated according to the length previously specified. *ASCII* requests an input of 5 characters for a 64-bit key and 13 characters for a 128-bit key. For *Hexadecimal*, enter 10 characters for a 64-bit key or 26 characters for a 128-bit key in hexadecimal notation.

WPA-PSK

To enter a key for WPA-PSK, select the input method *Passphrase* or *Hexadecimal*. In the *Passphrase* mode, the input must be 8 to 63 characters. In the *Hexadecimal* mode, enter 64 characters.

Expert Settings

This button opens a dialog for the detailed configuration of your WLAN connection. Usually there should be no need to change the preconfigured settings.

Channel

The specification of a channel on which the WLAN station should work is only needed in *Ad-hoc* and *Master* modes. In *Managed* mode, the card automatically searches the available channels for access points. In *Ad-hoc* mode, select one of the 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*.

Bit Rate

Depending on the performance of your network, you may want to set a certain bit rate for the transmission from one point to another. In the default setting *Auto*, the system tries to use the highest possible data transmission rate. Some WLAN cards do not support the setting of bit rates.

Access Point

In an environment with several access points, one of them can be preselected by specifying the MAC address.

Use Power Management

When you are on the road, use power saving technologies to maximize the operating time of your battery. More information about power management is available in [Chapter 17, *Power Management*](#) (page 191). Using power management may affect the connection quality and increase the network latency.

Click next to finish the setup. If you have chosen WPA-EAP authentication, another configuration step is needed before your station is ready for deployment in the WLAN. Enter the credentials you have been given by your network administrator. For TLS, provide *Identity*, *Client Certificate*, *Client Key*, and *Server Certificate*. TTLS and PEAP require *Identity* and *Password*. *Server Certificate* and *Anonymous Identity* are optional. YaST searches for any certificate under `/etc/cert`. Therefore, save the certificates given to you to this location and restrict access to these files to 0600 (owner read and write). Click *Details* to enter the advanced authentication dialog for your WPA-EAP setup. Select the authentication method for the second stage of EAP-TTLS or EAP-PEAP communication. If you selected TTLS in the previous dialog, choose any, MD5, GTC, CHAP, PAP, MSCHAPv1, or MSCHAPv2. If you selected PEAP, choose any, MD5, GTC, or MSCHAPv2. *PEAP version* can be used to force the use of a certain PEAP implementation if the automatically-determined setting does not work for you.

IMPORTANT: Security in Wireless Networks

Be sure to use one of the supported authentication and encryption methods to protect your network traffic. Unencrypted WLAN connections allow third parties to intercept all network data. Even a weak encryption (WEP) is better than none at all. Refer to [Section “Encryption”](#) (page 274) and [Section “Security”](#) (page 279) for information.

20.1.3 Utilities

The package `wireless-tools` contains utilities that allow to set wireless LAN specific parameters and get statistics. See http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html for more information.

20.1.4 Tips and Tricks for Setting Up a WLAN

These tips can help tweak speed and stability as well as security aspects of your WLAN.

Stability and Speed

The performance and reliability of a wireless network mainly depend on whether the participating stations receive a clean signal from the other stations. Obstructions like walls greatly weaken the signal. The more the signal strength sinks, the more the transmission slows down. During operation, check the signal strength with the `iwconfig` utility on the command line (`Link Quality` field) or with `NetworkManager` or `KNetworkManager`. If you have problems with the signal quality, try to set up the devices somewhere else or adjust the position of the antennas of your access points. Auxiliary antennas that substantially improve the reception are available for a number of PCMCIA WLAN cards. The rate specified by the manufacturer, such as 54 Mbit/s, is a nominal value that represents the theoretical maximum. In practice, the maximum data throughput is no more than half this value.

Security

If you want to set up a wireless network, remember that anybody within the transmission range can easily access it if no security measures are implemented. Therefore, be sure to activate an encryption method. All WLAN cards and access points support WEP encryption. Although this is not entirely safe, it does present an obstacle for a potential attacker. WEP is usually adequate for private use. WPA-PSK would be even better, but it is not implemented in older access points or routers with WLAN functionality. On some devices, WPA can be implemented by means of a firmware update. Furthermore, although Linux supports WPA on most hardware components, some drivers do not offer WPA support. 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.

20.1.5 Troubleshooting

If your WLAN card fails to respond, check if you have downloaded the needed firmware. Refer to `/usr/share/doc/packages/wireless-tools/README.firmware` for more information.

Multiple Network Devices

Modern laptops usually have a network card and a WLAN card. If you configured both devices with DHCP (automatic address assignment), you may encounter problems with the name resolution and the default gateway. This is evident from the fact that you can ping the router but cannot surf the Internet. The Support Database features an article on this subject at http://en.opensuse.org/SDB:Name_Resolution_Does_Not_Work_with_Several_Concurrent_DHCP_Clients.

Problems with Prism2 Cards

Several drivers are available for devices with Prism2 chips. The various cards work more or less smoothly with the various drivers. With these cards, WPA is only possible with the hostap driver. If such a card does not work properly or not at all or you want to use WPA, read `/usr/share/doc/packages/wireless-tools/README.prism2`.

20.1.6 For More Information

The Internet pages of Jean Tourrilhes, who developed the *Wireless Tools* for Linux, present a wealth of useful information about wireless networks. See http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html.

SLP Services in the Network

The *service location protocol* (SLP) was developed to simplify the configuration of networked clients within a local network. To configure a network client, including all required services, the administrator traditionally needs detailed knowledge of the servers available in the network. SLP makes the availability of selected services known to all clients in the local network. Applications that support SLP can use the information distributed and be configured automatically.

SUSE® Linux Enterprise Desktop supports installation using installation sources provided with SLP and contains many system services with integrated support for SLP. YaST and Konqueror both have appropriate front-ends for SLP. You can use SLP to provide networked clients with central functions, such as an installation server, file server, or print server on your system.

IMPORTANT: SLP Support in SUSE Linux Enterprise Desktop

Services that offer SLP support include cupsd, rsyncd, ypserv, openldap2, ksys-guardd, saned, kdm, vnc, login, smpppd, rpasswd, postfix, and sshd (via fish).

21.1 Installation

Only an SLP client and slptools are installed by default. If you want to provide services via SLP, install the package `openslp-server`. To install the package, start YaST and select *Software > Software Management*. Now choose *Filter > Patterns* and click *Misc. Server*. Select `openslp-server`. Confirm the installation of the required packages to finish the installation process.

21.2 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 SUSE Linux Enterprise Desktop, 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`. This includes slpd in the set of services to be started at boot time.

21.3 SLP Front-Ends in SUSE Linux Enterprise Desktop

To find services provided via SLP in your network, use an SLP front-end. SUSE Linux Enterprise Desktop contains several front-ends:

slptool

slptool is a simple command line program that can be used to announce SLP inquiries in the network or announce proprietary services. `slptool --help` lists all available options and functions. slptool can also be called from scripts that process SLP information. For example, to find all network time servers that announce themselves in the current network, run the command:

```
slptool findsrvs service:ntp
```

YaST

Within YaST there is also a SLP browser available. However, this browser is not available through the YaST Control Center. To start this YaST module, run `yast2 slp` as `root` user. Click on the different protocols on the lefthand side of the user interface to get more information about the respective service.

21.4 Providing Services via SLP

Many applications in SUSE Linux Enterprise Desktop 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. The following is an example of a file for registering a scanner service:

```
## Register a saned service on this system
## en means english language
## 65535 disables the timeout, so the service registration does
## not need refreshes
service:scanner.sane://$HOSTNAME:6566,en,65535
watch-port-tcp=6566
description=SANE scanner daemon
```

The most important line in this file is the *service URL*, which begins with `service:.` This contains the service type (`scanner.sane`) and the address under which the service is available on the server. `$HOSTNAME` is automatically replaced with the full hostname. The name of the TCP port on which the relevant service can be found follows, separated by a colon. Then enter the language in which the service should appear and the duration of registration in seconds. These should be separated from the service URL by commas. Set the value for the duration of registration between 0 and 65535. 0 prevents registration. 65535 removes all restrictions.

The registration file also contains the two variables `watch-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 should 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.

21.5 For More Information

The following sources provide further information about SLP:

RFC 2608, 2609, 2610

RFC 2608 generally deals with the definition of SLP. RFC 2609 deals with the syntax of the service URLs used in greater detail and RFC 2610 deals with DHCP via SLP.

<http://www.openslp.org/>

The home page of the OpenSLP project.

`/usr/share/doc/packages/openslp`

This directory contains all available documentation for SLP, including a README .SuSE containing the SUSE Linux Enterprise Desktop 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.

Time Synchronization with NTP

22

The NTP (network time protocol) mechanism is a protocol for synchronizing the system time over the network. First, a machine can obtain the time from a server that is a reliable time source. Second, a machine can itself act as a time source for other computers in the network. The goal is twofold—maintaining the absolute time and synchronizing the system time of all machines within a network.

Maintaining an exact system time is important in many situations. The built-in hardware (BIOS) clock does often not meet the requirements of applications like databases. Manual correction of the system time would lead to severe problems because, for example, a backward leap can cause malfunction of critical applications. Within a network, it is usually necessary to synchronize the system time of all machines, but manual time adjustment is a bad approach. ntp provides a mechanism to solve these problems. It continuously adjusts the system time with the help of reliable time servers in the network. It further enables the management of local reference clocks, such as radio-controlled clocks.

NOTE

To enable time synchronization by means of active directory, follow the instructions found at [Joining an AD Domain](#) (↑Security Guide).

22.1 Configuring an NTP Client with YaST

ntp is preset to use the local computer clock as a time reference. Using the (BIOS) clock, however, only serves as a fallback for the case that no time source of greater precision is available. YaST facilitates the configuration of an NTP client. For a system that is not running a firewall, use either the quick or advanced configuration. For a firewall-protected system, the advanced configuration can open the required ports in SuSEfirewall2.

22.1.1 Advanced NTP Client Configuration

You can either configure the NTP client manually or automatically to get a list of the NTP servers available in your network via DHCP. If you choose *Configure NTP Daemon via DHCP*, the manual options explained below are not available.

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:

Server

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. 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`.

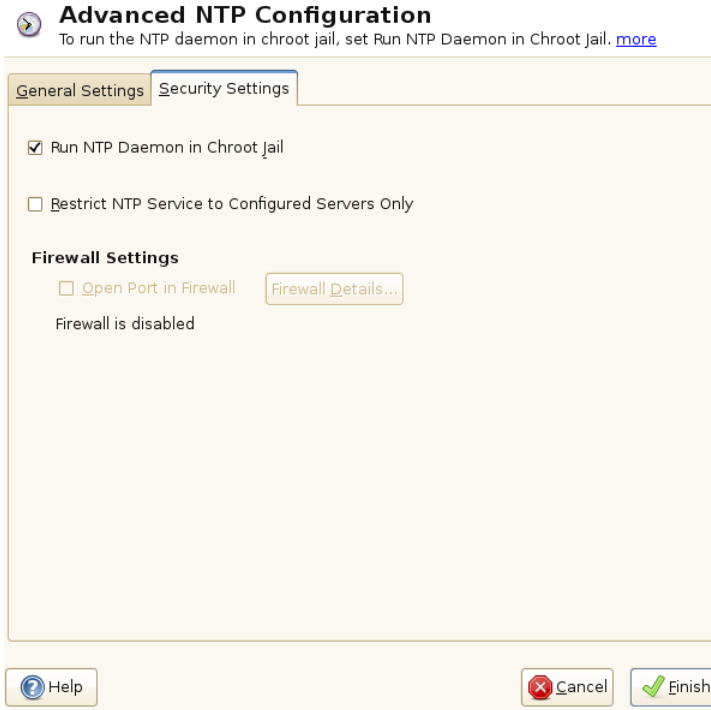
Outgoing Broadcast

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 22.1 *Advanced NTP Configuration: Security Settings*



In the *Security Settings* tab, 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`, because it prevents the attacker from compromising the entire system.

Restrict NTP Service to Configured Servers Only increases the security of your system by disallowing remote computers to view and modify NTP settings of your computer and to use the trap facility for remote event logging. Once enabled, these restrictions apply to all remote computers, unless you override the access control options for individual computers in the list of time sources in the *General Settings* tab. For all other remote computers, only querying for local time is allowed.

Enable *Open Port in Firewall* if `SuSEfirewall2` is active, which it is by default. If you leave the port closed, it is not possible to establish a connection to the time server.

22.2 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 `rcntpd start`, it takes about one hour until the time is stabilized and the drift file for correcting the local computer clock is created. With the drift file, the systematic error of the hardware clock can be computed as soon as the computer is powered on. The correction is used immediately, resulting in a higher stability of the system time.

There are two possible ways to use the NTP mechanism as a client: First, the client can query the time from a known server in regular intervals. With many clients, this approach can cause a high load on the server. Second, the client can wait for NTP broadcasts sent out by broadcast time servers in the network. This approach has the disadvantage that the quality of the server is unknown and a server sending out wrong information can cause severe problems.

If the time is obtained via broadcast, you do not need the server name. In this case, enter the line `broadcastclient` in the configuration file `/etc/ntp.conf`. To use one or more known time servers exclusively, enter their names in the line starting with `servers`.

22.3 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 net-

work. For this purpose, they are assigned special IP addresses in the form `127.127.t.u`. Here, *t* stands for the type of the clock and determines which driver is used and *u* for the unit, which determines the interface used.

Normally, the individual drivers have special parameters that describe configuration details. The file `/usr/share/doc/packages/ntp-doc/drivers/driverNN.html` (where *NN* is the number of the driver) provides information about the particular type of clock. For example, the “type 8” clock (radio clock over serial interface) requires an additional mode that specifies the clock more precisely. The Conrad DCF77 receiver module, for example, has mode 5. To use this clock as a preferred reference, specify the keyword `prefer`. The complete `server` line for a Conrad DCF77 receiver module would be:

```
server 127.127.8.0 mode 5 prefer
```

Other clocks follow the same pattern. Following the installation of the `ntp-doc` package, the documentation for `ntp` is available in the directory `/usr/share/doc/packages/ntp-doc`. The file `/usr/share/doc/packages/ntp-doc/refclock.html` provides links to the driver pages describing the driver parameters.

Using 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 wired or wireless networks when you are moving. NetworkManager can automatically connect to known wireless networks. It can also manage several network connections in parallel, the fastest connection is then used as default. Furthermore, you can switch between available networks manually and manage your network connection using an applet or widget in the system tray.

On laptop computers, NetworkManager is active by default. However it can be at any time activated or deactivated using YaST as described in [Section 23.2, “Enabling NetworkManager”](#) (page 292).

23.1 Use Cases for NetworkManager

NetworkManager provides a sophisticated and intuitive user interface which enables users easily to 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.

- You want to use SCPM for network configuration management. To use SCPM and NetworkManager at the same time, disable the network resource in SCPM configuration.

23.2 Enabling NetworkManager

If you want to manage your network connection with NetworkManager, enable NetworkManager in the YaST Network Settings module. To enable NetworkManager, proceed as follows:

- 1 Run YaST and go to *Network Devices > Network Settings*.
- 2 The *Network Settings* dialog opens. Go to the *Global Options* tab.
- 3 In the *Network Setup Method* field, activate *User Controlled with NetworkManager*.
- 4 Click *Finish*.
- 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 19.4, “Configuring a Network Connection with YaST”](#) (page 232) and [Section 20.1, “Wireless LAN”](#) (page 271).

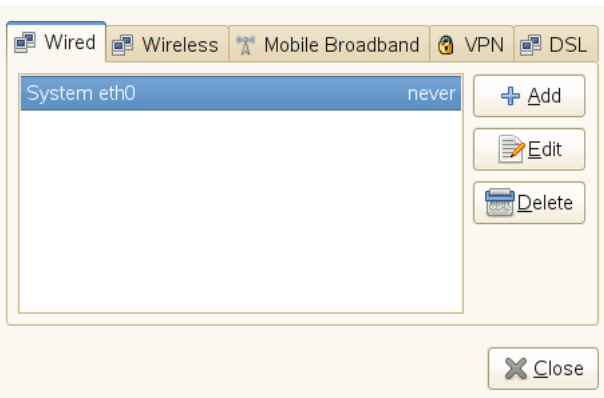
After having enabled NetworkManager, configure your network connections with the NetworkManager as described in [Section 23.3, “Configuring Network Connections”](#) (page 293).

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.

23.3 Configuring Network Connections

After having enabled NetworkManager in YaST, configure your network connections in a dialog available from the GNOME Control Center or from the Personal Settings in KDE 4. If you use GNOME, start the GNOME Control Center from the main menu, then select *System > Network Configurations* to open the *Network Configuration* dialog. If you use KDE, start the *Personal Settings* from the main menu by clicking *Configure Desktop*, then select *Advanced > Network Settings* to open the *Network Settings* dialog.

Figure 23.1 GNOME Network Configuration Dialog



Alternatively, you can start the configuration dialogs from the NetworkManager applet/widget in the system tray by clicking *Configure* (KDE) or by right-clicking the GNOME applet and selecting *Edit Connections*.

The GNOME and KDE 4 configuration dialog shows tabs for all types of network connections, such as wired, wireless, Mobile Broadband, DSL, and VPN connections. NetworkManager also supports connections to 802.1X protected networks.

To add a new connection, click the tab for the connection type you want to use and click *Add*. Enter a *Connection Name* and your connection details. 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), specify the *MAC address* (the Hardware Address) of the device in order to tie the connection to this device. Click *OK* or *Apply*

to confirm your settings. The newly configured network connection now appears in the list of available networks you get by left-clicking the NetworkManager applet or widget.

NOTE: Hidden Networks

To connect to a “hidden” network (a network that does not broadcast its service) you have to know the Extended Service Set Identifier (ESSID) of the network because it cannot be detected automatically. In this case, enter the ESSID and the encryption parameters, if necessary.

When editing each connection, you can also define if NetworkManager should automatically use this connection (activate *Connect Automatically*) or should use this connection systemwide (activate *Available to all users*). Such system connections can be shared by all users and are made available right after NetworkManager is started—before any users log in. To create and edit system connections, `root` permission is required.

23.4 Using KDE NetworkManager Widget

In KDE 4, the KNetworkManager applet for controlling NetworkManager has been replaced by the NetworkManager widget. Widgets are small applications that can be integrated into your desktop or your panel. If the network has been set up for NetworkManager control, the widget usually starts automatically with the desktop environment and is shown as icon in the system tray.

The NetworkManager widget shows the current network status as an icon and reports on changes using notifications. Use the widget to configure new network connections, to manually select another network connection, to disable the use of wireless networks, or to switch to offline mode altogether, if needed. The appearance of the icon depends on the type and state of the current network connection. Hold the mouse cursor over the icon to see details about the connection.

NetworkManager distinguishes two types of connections: trusted and untrusted. A trusted connection is any network that you explicitly selected. All others are untrusted. Right-click the connection icons to show a list of connections that you have already used at least once in the past. The currently used connection is marked in the menu.

Left-click any of the connection applets to choose another network connection at any time. Such a choice takes priority over automatically selected networks. The chosen network is used as long as it is available, meaning that plugging a network cable in does not switch to a wired network connection automatically.

23.5 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 GNOME NetworkManager applet, the applet is probably not started. Press `Alt + F2` and enter `nm-applet` to start it manually.

23.5.1 Connecting to Wired Networks

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 selected in the menu.
- 2 To switch to another network, choose it from the list.
- 3 To switch off all network connections, both wired and wireless, right-click the applet icon and uncheck *Enable Networking*.

23.5.2 Connecting to Wireless Networks

Available visible wireless networks are listed in the GNOME NetworkManager applet menu under *Wireless Networks*. The signal strength of each network is also shown in the menu. Encrypted wireless networks are marked with a shield icon.


Procedure 23.1 *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. Choose the type of *Wireless Security* the network uses and enter the appropriate *Password*.
- 3 To connect to a network that does not broadcast its service set identifier (ESSID) and therefore cannot be detected automatically, left-click the NetworkManager icon and choose *Connect to Other Wireless Network*.
- 4 In the dialog that opens, enter the ESSID and set encryption parameters if necessary.
- 5 To disable wireless networking, right-click the applet icon and uncheck *Enable Wireless*. This can be very useful if you are on a plane or in any other environment where wireless networking is not allowed.

23.5.3 Configuring Your Wireless Card as an Access Point

If your wireless card supports access point mode, you can use NetworkManager for configuration.

- 1 Click *Create New Wireless Network*.



New wireless network

Enter a name for the wireless network you wish to create.

Network Name:

Wireless Security:

Password:

☐ Show password

- 2 Add the network name and set the encryption in the *Wireless Security* dialog.

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 “Authentication”](#) (page 273).

23.6 NetworkManager and VPN

NetworkManager supports several Virtual Private Network (VPN) technologies:

- NovellVPN—package `NetworkManager-novellvpn`
- OpenVPN—package `NetworkManager-openvpn`
- `vpnc` (Cisco)—package `NetworkManager-vpnc`
- PPTP (Point-to-Point Tunneling Protocol)—package `NetworkManager-pptp`

To use VPN with NetworkManager, install the appropriate VPN packages first. You need two packages for each VPN technology: one of the packages above (providing the generic support for NetworkManager), and the respective desktop-specific package for your applet.

For KDE, choose one of the following:

- NovellVPN support for KNetworkManager—package `NetworkManager-novellvpn-kde4`
- OpenVPN support for KNetworkManager—package `NetworkManager-openvpn-kde4`
- `vpnc` (Cisco) support for KNetworkManager—package `NetworkManager-vpnc-kde4`

PPTP support for KDE is not available yet, but is being worked on.

For GNOME, choose one of the following:

- NovellVPN support for GNOME NetworkManager applet—package `NetworkManager-novellvpn-gnome`
- OpenVPN support for GNOME NetworkManager applet—package `NetworkManager-openvpn-gnome`
- vpnc (Cisco) support for GNOME NetworkManager applet—package `NetworkManager-vpnc-gnome`
- PPTP (Point-to-Point Tunneling Protocol) support for GNOME NetworkManager applet—package `NetworkManager-pptp-gnome`

After you have installed the packages, configure your VPN connection as described in [Section 23.3, “Configuring Network Connections”](#) (page 293).

23.7 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.

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. Con-

nections 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, refer to [Section 23.3, “Configuring Network Connections”](#) (page 293).

If you switch to offline mode from using a wireless connection, NetworkManager blanks the ESSID. This ensures that the card is disconnected.

23.7.1 Storing Passwords and Credentials

If you do not want to enter your credentials anew 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.

NetworkManager can also retrieve its certificates for secure connections (for example, encrypted wired, wireless or VPN connections) from the certificate store. For more information, refer to Chapter 12, *Certificate Store* (†Security Guide).

Another option is to use single sign-on with Novell CASA. Single Sign-on is a method of access control that enables users to authenticate once and thus gain access to the resources of multiple software systems. If Novell CASA is configured for your system, NetworkManager will not ask for an additional password to unlock GNOME Keyring Manager. Instead, the keyring will be unlocked automatically when the users logs in to the desktop. For more information about Novell CASA, refer to <http://developer.novell.com/wiki/index.php/Special:Downloads/casa>.

23.8 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 eth-

ernet cards), you can tie a connection to a certain device by explicitly specifying the hardware address (or MAC address) of the device.

Look up the MAC address of your device either in the *Connection Information*, available from the applet/widget, or use the output of command line tools like `nm-tool` or `ifconfig`. Then start the dialog for configuring network connections from the GNOME Control Center with *System > Network Configurations* or in KDE 4 from the *Personal Settings* with *Advanced > Network Settings*. Choose the connection you want to modify and click *Edit*. On the *Wired* or *Wireless* tab, enter the *MAC address* of the device and confirm your changes with *OK*.

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 from the GNOME Control Center with *System > Network Configurations* or in KDE 4 from the *Personal Settings* with *Advanced > Network Settings*. 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 from the GNOME Control Center with *System > Network Configurations* or in KDE 4 from the *Personal Settings* with *Advanced > Network Settings*. Choose the connection you want to modify and click *Edit*. Switch to the *IPv4 Settings* tab. From the *Method* drop-down list, choose *Shared to other computers*. 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 *DHCP Server* is not already shown as *Allowed Service*, select *DHCP 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 from the GNOME Control Center with *System > Network Configurations* or in KDE 4 from the *Personal Settings* with *Advanced > Network Settings*. Choose the connection you want to modify and click *Edit*. Switch to the *IPv4 Settings* 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. Click *Routes* to add additional routes or to override automatic 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 23.7, “NetworkManager and Security”](#) (page 298).

23.9 Troubleshooting

Connection problems can occur. Some common problems related to NetworkManager include the applet not starting, a missing VPN option, and issues with SCPM. Methods for resolving and preventing these problems depend on the tool used.

NetworkManager Desktop Applet/Widget Does Not Start

GNOME NetworkManager applet or KDE NetworkManager widget should start automatically if the network is set up for NetworkManager control. If the applet/widget does not start, check if NetworkManager is enabled in YaST as described in [Section 23.2, “Enabling NetworkManager”](#) (page 292). Then make sure that the appropriate package for your desktop environment is also installed. If you are using KDE 4, the package is `NetworkManager-kde4`. For GNOME users the package is `NetworkManager-gnome`.

If the GNOME desktop applet is installed but is not running for some reason (perhaps you quit it accidentally), start it manually with the command `nm-applet`.

If your KDE 4 system tray does not show any icon for network connections (as might be the case after switching from a static network configuration to user-controlled with NetworkManager in YaST), add the NetworkManager widget to the panel: right-click an empty patch on the panel, and select *Panel Options > Add Widgets*. (If your desktop objects are currently locked, you might need to click *Unlock Widgets* first before you can add any objects.) In the dialog box that appears, select *NetworkManager* and click *Add Widget*.

NetworkManager Applet/Widget Does Not Include the VPN Option

Support for NetworkManager, applets, and VPN for NetworkManager is distributed in separate packages. If your NetworkManager applet/widget does not include the VPN option, check if the packages with NetworkManager support for your VPN technology are installed. For more information, see [Section 23.6, “NetworkManager and VPN”](#) (page 297).

SCPM Does Not Switch the Network Configuration

You are probably using SCPM together with NetworkManager. NetworkManager is not currently able to work with SCPM profiles. Do not use NetworkManager together with SCPM when SCPM profiles also change network settings. To use SCPM and NetworkManager at the same time, disable the network resource in SCPM configuration.

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`.

23.10 For More Information

More information about NetworkManager can be found on the following Web sites and directories:

- <http://www.gnome.org/projects/NetworkManager/>—NetworkManager project page
- For more information on KDE NetworkManager widget refer to <http://userbase.kde.org/KNetworkManager>.

- Also check out the information in the following directories for the latest information about NetworkManager and the GNOME NetworkManager applet and the KDE NetworkManager widget: `/usr/share/doc/packages/NetworkManager/`, `/usr/share/doc/packages/NetworkManager-kde4/` and `/usr/share/doc/packages/NetworkManager-gnome/`

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.

24.1 Terminology

The following are some terms used in Samba documentation and in the YaST module.

SMB protocol

Samba uses the SMB (server message block) protocol that is based on the NetBIOS services. Microsoft released the protocol so other software manufacturers could establish connections to a Microsoft domain network. With Samba, the SMB protocol works on top of the TCP/IP protocol, so the TCP/IP protocol must be installed on all clients.

CIFS protocol

CIFS (common Internet file system) protocol is another protocol supported by Samba. CIFS defines a standard remote file system access protocol for use over the network, enabling groups of users to work together and share documents across the network.

NetBIOS

NetBIOS is a software interface (API) designed for communication between machines providing a name service. It enables machines connected to the network to

reserve names for themselves. After reservation, these machines can be addressed by name. There is no central process that checks names. Any machine on the network can reserve as many names as it wants as long as the names are not already in use. The NetBIOS interface can be implemented for different network architectures. An implementation that works relatively closely with network hardware is called NetBEUI, but this is often referred to as NetBIOS. Network protocols implemented with NetBIOS are IPX from Novell (NetBIOS via TCP/IP) and TCP/IP.

The NetBIOS names sent via TCP/IP have nothing in common with the names used in `/etc/hosts` or those defined by DNS. NetBIOS uses its own, completely independent naming convention. However, it is recommended to use names that correspond to DNS hostnames to make administration easier or use DNS natively. This is the default used by Samba.

Samba server

Samba server provides SMB/CIFS services and NetBIOS over IP naming services to clients. For Linux, there are three daemons for Samba server: `smnd` for SMB/CIFS services, `nmbd` for naming services, and `winbind` for authentication.

Samba client

Samba client is a system that uses Samba services from a Samba server over the SMB protocol. All common operating systems, such as Mac OS X, Windows, and OS/2, support the SMB protocol. The TCP/IP protocol must be installed on all computers. Samba provides a client for the different UNIX flavors. For Linux, there is a kernel module for SMB that allows the integration of SMB resources on the Linux system level. You do not need run any daemon for Samba client.

Shares

SMB servers provide 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.

24.2 Configuring a Samba Server

For configuring a Samba server, see the SUSE Linux Enterprise Server documentation.

24.3 Configuring Clients

Clients can only access the Samba server via TCP/IP. NetBEUI and NetBIOS via IPX cannot be used with Samba.

24.3.1 Configuring a Samba Client with YaST

Configure a Samba client to access resources (files or printers) on the Samba server. Enter the domain or workgroup in the dialog *Network Services > Windows Domain Membership*. If you activate *Also Use SMB Information for Linux Authentication*, the user authentication runs over the Samba server. After completing all settings, click *Finish* to finish the configuration.

24.4 Samba as Login Server

In networks where predominantly Windows clients are found, it is often preferable that users may only register with a valid account and password. In a Windows-based network, this task is handled by a primary domain controller (PDC). You can use a Windows NT server configured as PDC, but this task can also be done with the help of a Samba server. The entries that must be made in the `[global]` section of `smb.conf` are shown in [Example 24.1, “Global Section in smb.conf”](#) (page 307).

Example 24.1 *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:

Example 24.2 *Setting Up a Machine Account*

```
useradd hostname\$\n  
smbpasswd -a -m hostname
```

With the `useradd` command, a dollar sign is added. The command `smbpasswd` inserts this automatically when the parameter `-m` is used. The commented configuration example (`/usr/share/doc/packages/Samba/examples/smb.conf.SuSE`) contains settings that automate this task.

Example 24.3 *Automated Setup of a Machine Account*

```
add machine script = /usr/sbin/useradd -g nogroup -c "NT Machine Account" \n  
-s /bin/false %m\$\n
```

To make sure that Samba can execute this script correctly, choose a Samba user with the required administrator permissions. To do so, select one user and add it to the `ntadmin` group. After that, all users belonging to this Linux group can be assigned Domain Admin status with the command:

```
net groupmap add ntgroup="Domain Admins" unixgroup=ntadmin
```

More information about this topic is provided in Chapter 12 of the Samba HOWTO Collection, found in `/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf`.

24.5 For More Information

Detailed Samba information is available in the digital documentation. Enter `apropos samba` at the command line to display some manual pages or just browse the `/usr/share/doc/packages/samba` directory if Samba documentation is installed for more online documentation and examples. Find a commented example configuration (`smb.conf.SuSE`) in the `examples` subdirectory.

The Samba HOWTO Collection provided by the Samba team includes a section about troubleshooting. In addition to that, Part V of the document provides a step-by-step guide to checking your configuration. You can find Samba HOWTO Collection in `/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf` after installing the package `samba-doc`.

Also read the Samba page in the openSUSE wiki at <http://en.opensuse.org/Samba>.

Sharing File Systems with NFS

Distributing and sharing file systems over a network is a common task in corporate environments. NFS is a proven system that also works together with the yellow pages protocol NIS. For a more secure protocol that works together with LDAP and may also be kerberized, check NFSv4.

NFS works with NIS to make a network transparent to the user. With NFS, it is possible to distribute arbitrary file systems over the network. With an appropriate setup, users always find themselves in the same environment regardless of the terminal they currently use.

25.1 Installing the Required Software

To configure your host as an NFS client, you do not need to install additional software. All packages needed to configure an NFS client are installed by default.

25.2 Importing File Systems with YaST

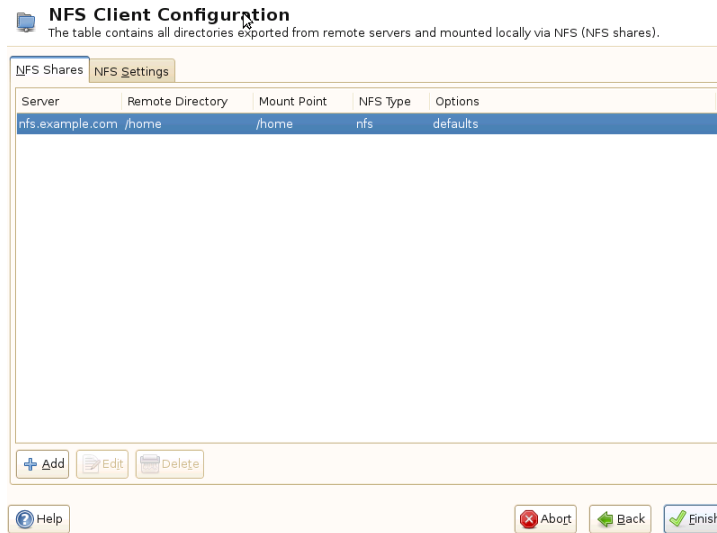
Users authorized to do so can mount NFS directories from an NFS server into their own file trees. This can be achieved using the YaST module *NFS Client*. Click on *Add* and enter the hostname of the NFS server, the directory to import, and the mount point at which to mount this directory locally. The changes will take effect after *Finish* is clicked in the first dialog.

In the *NFS Settings* tab, click *Open Port in Firewall* to open the firewall to allow access to the service from remote computers. The firewall status is displayed next to the check box. 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 *Finish* to save your changes. See **Figure 25.1, “NFS Client Configuration with YaST”** (page 312).

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 25.1 *NFS Client Configuration with YaST*



25.3 Importing File Systems Manually

File systems can also be imported manually from an NFS server. The prerequisite for this is a running RPC port mapper, which can be started by entering `rcrpcbind start` as `root`. Once this prerequisite is met, remote exported file systems can be

mounted in the file system just like local hard disks using the `mount` command in the following manner:

```
mount host:remote-path local-path
```

If user directories from the machine `nfs.example.com`, for example, should be imported, use the following command:

```
mount nfs.example.com:/home /home
```

25.3.1 Using the Automount Service

As well as the regular local device mounts, the `autofs` daemon can be used to mount remote file systems automatically too. To do this, add the following entry in the your `/etc/auto.master` file:

```
/nfsmounts /etc/auto.nfs
```

Now the `/nfsmounts` directory acts as a root for all the NFS mounts on the client if the `auto.nfs` file is completed appropriately. The name `auto.nfs` is chosen for sake of convenience—you can choose any name. In the selected file (create it if it does not exist), add entries for all the NFS mounts as in the following example:

```
localdata -fstype=nfs server1:/data  
nfs4mount -fstype=nfs4 server2:/
```

Activate the settings with `rcautofs start`. For this example, `/nfsmounts/localdata`, the `/data` directory of `server1`, is then mounted with NFS and `/nfsmounts/nfs4mount` from `server2` is mounted with NFSv4.

If the `/etc/auto.master` file is edited while the service `autofs` is running, the automounter must be restarted for the changes to take effect. Do this with `rcautofs restart`.

25.3.2 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 manually. For these mounts, use `nfs4` instead of `nfs` in the third column and make sure that the remote file system is given as `/` after the `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 command for mounting and it is only needed to provide the mount point as in:

```
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.

25.4 NFS with Kerberos

To use Kerberos authentication for NFS, GSS security must be enabled. To do so, select *Enable GSS Security* in the initial YaST dialog. Note, that you must have a working Kerberos server to use this feature. YaST does not set up the server but only 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. This means that they access the same KDC (Key Distribution Center) server and share their `krb5.keytab` file (the default location on any machine is `/etc/krb5.keytab`).
- 2 Start the `gssd` service on the client with `rcgssd start`.

For further information about configuring kerberized NFS, refer to the links in [Section 25.5, “For More Information”](#) (page 315).

25.5 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 any questions on NFSv4, refer to the Linux NFSv4 Frequently Asked Questions [<http://www.citi.umich.edu/projects/nfsv4/linux/faq/>] FAQ.

File Synchronization

Today, many people use several computers—one computer at home, one or several computers at the workplace, and possibly a laptop or PDA on the road. Many files are needed on all these computers. You may want to be able to work with all computers and modify the files and subsequently have the latest version of the data available on all computers.

26.1 Available Data Synchronization Software

Data synchronization is no problem for computers that are permanently linked by means of a fast network. In this case, use a network file system, like NFS, and store the files on a server, enabling all hosts to access the same data via the network. This approach is impossible if the network connection is poor or not permanent. When you are on the road with a laptop, copies of all needed files must be on the local hard disk. However, it is then necessary to synchronize modified files. When you modify a file on one computer, make sure a copy of the file is updated on all other computers. For occasional copies, this can be done manually with `scp` or `rsync`. However, if many files are involved, the procedure can be complicated and requires great care to avoid errors, such as overwriting a new file with an old file.

WARNING: Risk of Data Loss

Before you start managing your data with a synchronization system, you should be well acquainted with the program used and test its functionality. A backup is indispensable for important files.

The time-consuming and error-prone task of manually synchronizing data can be avoided by using one of the programs that use various methods to automate this job. The following summaries are merely intended to convey a general understanding of how these programs work and how they can be used. If you plan to use them, read the program documentation.

26.1.1 CVS

CVS, which is mostly used for managing program source versions, offers the possibility to keep copies of the files on multiple computers. Accordingly, it is also suitable for data synchronization. CVS maintains a central repository on the server in which the files and changes to files are saved. Changes that are performed locally are committed to the repository and can be retrieved from other computers by means of an update. Both procedures must be initiated by the user.

CVS is very resilient to errors when changes occur on several computers. The changes are merged and, if changes took place in the same lines, a conflict is reported. When a conflict occurs, the database remains in a consistent state. The conflict is only visible for resolution on the client host.

26.1.2 rsync

When no version control is needed but large directory structures need to be synchronized over slow network connections, the tool rsync offers well-developed mechanisms for transmitting only changes within files. This not only concerns text files, but also binary files. To detect the differences between files, rsync subdivides the files into blocks and computes checksums over them.

The effort put into the detection of the changes comes at a price. The systems to synchronize should be scaled generously for the usage of rsync. RAM is especially important.

26.2 Determining Factors for Selecting a Program

There are some important factors to consider when deciding which program to use.

26.2.1 Client-Server versus Peer-to-Peer

Two different models are commonly used for distributing data. In the first model, all clients synchronize their files with a central server. The server must be accessible by all clients at least occasionally. This model is used by CVS.

The other possibility is to let all networked hosts synchronize their data between each other as peers. rsync actually works in client mode, but any client can also act as a server.

26.2.2 Portability

CVS and rsync are also available for many other operating systems, including various Unix and Windows systems.

26.2.3 Interactive versus Automatic

In CVS, the data synchronization is started manually by the user. This allows fine control over the data to synchronize and easy conflict handling. However, if the synchronization intervals are too long, conflicts are more likely to occur.

26.2.4 Conflicts: Incidence and Solution

Conflicts only rarely occur in CVS, even when several people work on one large program project. This is because the documents are merged on the basis of individual lines. When a conflict occurs, only one client is affected. Usually conflicts in CVS can easily be resolved.

There is no conflict handling in rsync. The user is responsible for not accidentally overwriting files and manually resolving all possible conflicts. To be on safe side, a versioning system like RCS can be additionally employed.

26.2.5 Selecting and Adding Files

In CVS, new directories and files must be added explicitly using the command `cvs add`. This results in greater user control over the files to synchronize. On the other hand, new files are often overlooked, especially when the question marks in the output of `cvs update` are ignored due to the large number of files.

26.2.6 History

An additional feature of CVS is that old file versions can be reconstructed. A brief editing remark can be inserted for each change and the development of the files can easily be traced later based on the content and the remarks. This is a valuable aid for theses and program texts.

26.2.7 Data Volume and Hard Disk Requirements

A sufficient amount of free space for all distributed data is required on the hard disks of all involved hosts. CVS require additional space for the repository database on the server. The file history is also stored on the server, requiring even more space. When files in text format are changed, only the modified lines need to be saved. Binary files require additional space amounting to the size of the file every time the file is changed.

26.2.8 GUI

Experienced users normally run CVS from the command line. However, graphical user interfaces are available for Linux, such as cervisia, and for other operating systems, like wincvs. Many development tools, such as kdevelop, and text editors, such as Emacs, provide support for CVS. The resolution of conflicts is often much easier to perform with these front-ends.

26.2.9 User Friendliness

rsync is rather easy to use and is also suitable for newcomers. CVS is somewhat more difficult to operate. Users should understand the interaction between the repository and local data. Changes to the data should first be merged locally with the repository. This is done with the command `cvs update`. Then the data must be sent back to the repository with the command `cvs commit`. Once this procedure has been understood, newcomers are also able to use CVS with ease.

26.2.10 Security against Attacks

During transmission, the data should ideally be protected against interception and manipulation. CVS and rsync can easily be used via ssh (secure shell), providing security against attacks of this kind. Running CVS via rsh (remote shell) should be avoided. Accessing CVS with the *pserver* mechanism in insecure networks is likewise not advisable.

26.2.11 Protection against Data Loss

CVS has been used by developers for a long time to manage program projects and is extremely stable. Because the development history is saved, CVS even provides protection against certain user errors, such as unintentional deletion of a file.

Table 26.1 *Features of the File Synchronization Tools: -- = very poor, - = poor or not available, o = medium, + = good, ++ = excellent, x = available*

	CVS	rsync
Client/Server	C-S	C-S
Portability	Lin,Un*x,Win	Lin,Un*x,Win
Interactivity	x	x
Speed	o	+
Conflicts	++	o

	CVS	rsync
File Sel.	Sel./file, dir.	Dir.
History	x	-
Hard Disk Space	--	o
GUI	o	-
Difficulty	o	+
Attacks	+ (ssh)	+(ssh)
Data Loss	++	+

26.3 Introduction to CVS

CVS is suitable for synchronization purposes if individual files are edited frequently and are stored in a file format, such as ASCII text or program source text. The use of CVS for synchronizing data in other formats, such as JPEG files, is possible, but leads to large amounts of data, because all variants of a file are stored permanently on the CVS server. In such cases, most of the capabilities of CVS cannot be used. The use of CVS for synchronizing files is only possible if all workstations can access the same server.

26.3.1 Configuring a CVS Server

The *server* is the host on which all valid files are located, including the latest versions of all files. Any stationary workstation can be used as a server. If possible, the data of the CVS repository should be included in regular backups.

When configuring a CVS server, it might be a good idea to grant users access to the server via SSH. If the user is known to the server as `tux` and the CVS software is installed on the server as well as on the client, the following environment variables must be set on the client side:

```
CVS_RSH=ssh CVSROOT=tux@server:/serverdir
```

The command `cvs init` can be used to initialize the CVS server from the client side. This needs to be done only once.

Finally, the synchronization must be assigned a name. Select or create a directory on the client exclusively to contain files to manage with CVS (the directory can also be empty). The name of the directory is also the name of the synchronization. In this example, the directory is called `synchome`. Change to this directory and enter the following command to set the synchronization name to `synchome`:

```
cvs import synchome tux wilber
```

Many CVS commands require a comment. For this purpose, CVS starts an editor (the editor defined in the environment variable `$EDITOR` or `vi` if no editor was defined). The editor call can be circumvented by entering the comment in advance on the command line, such as in the following example:

```
cvs import -m 'this is a test' synchome tux wilber
```

26.3.2 Using CVS

The synchronization repository can now be checked out from all hosts with `cvs co synchome`. This creates a new subdirectory `synchome` on the client. To commit your changes to the server, change to the directory `synchome` (or one of its subdirectories) and enter `cvs commit`.

By default, all files (including subdirectories) are committed to the server. To commit only individual files or directories, specify them as in `cvs commit file1 directory1`. New files and directories must be added to the repository with a command like `cvs add file1 directory1` before they are committed to the server. Subsequently, commit the newly added files and directories with `cvs commit file1 directory1`.

If you change to another workstation, check out the synchronization repository if this has not been done during an earlier session at the same workstation.

Start the synchronization with the server with `cvs update`. Update individual files or directories as in `cvs update file1 directory1`. To see the difference between the current files and the versions stored on the server, use the command `cvs diff` or `cvs diff file1 directory1`. Use `cvs -nq update` to see which files would be affected by an update.

Here are some of the status symbols displayed during an update:

U

The local version was updated. This affects all files that are provided by the server and missing on the local system.

M

The local version was modified. If there were changes on the server, it was possible to merge the differences in the local copy.

P

The local version was patched with the version on the server.

C

The local file conflicts with current version in the repository.

?

This file does not exist in CVS.

The status M indicates a locally modified file. Either commit the local copy to the server or remove the local file and run the update again. In this case, the missing file is retrieved from the server. If you commit a locally modified file and the file was changed in the same line and committed, you might get a conflict, indicated with C.

In this case, look at the conflict marks (“>>” and “<<”) in the file and decide between the two versions. As this can be a rather unpleasant job, you might decide to abandon your changes, delete the local file, and enter `cvs up` to retrieve the current version from the server.

26.4 Introduction to rsync

rsync is useful when large amounts of data need to be transmitted regularly while not changing too much. This is, for example, often the case when creating backups. Another

application concerns staging servers. These are servers that store complete directory trees of Web servers that are regularly mirrored onto a Web server in a DMZ.

26.4.1 Configuration and Operation

rsync can be operated in two different modes. It can be used to archive or copy data. To accomplish this, only a remote shell, like ssh, is required on the target system. However, rsync can also be used as a daemon to provide directories to the network.

The basic mode of operation of rsync does not require any special configuration. rsync directly allows mirroring complete directories onto another system. As an example, the following command creates a backup of the home directory of tux on a backup server named sun:

```
rsync -baz -e ssh /home/tux/ tux@sun:backup
```

The following command is used to play the directory back:

```
rsync -az -e ssh tux@sun:backup /home/tux/
```

Up to this point, the handling does not differ much from that of a regular copying tool, like scp.

rsync should be operated in “rsync” mode to make all its features fully available. This is done by starting the rsyncd daemon on one of the systems. Configure it in the file `/etc/rsyncd.conf`. For example, to make the directory `/srv/ftp` available with rsync, use the following configuration:

```
gid = nobody
uid = nobody
read only = true
use chroot = no
transfer logging = true
log format = %h %o %f %l %b
log file = /var/log/rsyncd.log
```

[FTP]

```
path = /srv/ftp
comment = An Example
```

Then start rsyncd with `rcrsyncd start`. rsyncd can also be started automatically during the boot process. Set this up by activating this service in the runlevel editor provided by YaST or by manually entering the command `insserv rsyncd`. rsyncd

can alternatively be started by `xinetd`. This is, however, only recommended for servers that rarely use `rsyncd`.

The example also creates a log file listing all connections. This file is stored in `/var/log/rsyncd.log`.

It is then possible to test the transfer from a client system. Do this with the following command:

```
rsync -avz sun::FTP
```

This command lists all files present in the directory `/srv/ftp` of the server. This request is also logged in the log file `/var/log/rsyncd.log`. To start an actual transfer, provide a target directory. Use `.` for the current directory. For example:

```
rsync -avz sun::FTP .
```

By default, no files are deleted while synchronizing with `rsync`. If this should be forced, the additional option `--delete` must be stated. To ensure that no newer files are deleted, the option `--update` can be used instead. Any conflicts that arise must be resolved manually.

26.5 For More Information

CVS

Important information about CVS can be found in the homepage <http://www.cvshome.org>.

rsync

Important information about `rsync` is provided in the man pages `man rsync` and `man rsyncd.conf`. A technical reference about the operating principles of `rsync` is featured in `/usr/share/doc/packages/rsync/tech_report.ps`. Find the latest news about `rsync` on the project Web site at <http://rsync.samba.org/>.