

SUSE Linux Enterprise Server

11 SP1

www.novell.com

November 08, 2010

Administration Guide



Administration Guide

All content is copyright © 2006–2010 Novell, Inc. All rights reserved.

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	xi
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 Managing Software with Command Line Tools	21
4.1 Using Zypper	21
4.2 RPM—the Package Manager	30

5	Bash and Bash Scripts	43
5.1	What is “The Shell”?	43
5.2	Writing Shell Scripts	49
5.3	Redirecting Command Events	50
5.4	Using Aliases	51
5.5	Using Variables in Bash	51
5.6	Grouping And Combining Commands	54
5.7	Working with Common Flow Constructs	55
5.8	For More Information	56
Part II	System	57
6	32-Bit and 64-Bit Applications in a 64-Bit System Environment	59
6.1	Runtime Support	60
6.2	Software Development	61
6.3	Software Compilation on Biarch Platforms	61
6.4	Kernel Specifications	63
7	Bootng and Configuring a Linux System	65
7.1	The Linux Boot Process	65
7.2	The init Process	69
7.3	System Configuration via /etc/sysconfig	78
8	The Boot Loader GRUB	81
8.1	Bootng with GRUB	82
8.2	Configuring the Boot Loader with YaST	93
8.3	Uninstalling the Linux Boot Loader	98
8.4	Creating Boot CDs	99
8.5	The Graphical SUSE Screen	100
8.6	Troubleshooting	101
8.7	For More Information	102
9	Special System Features	103
9.1	Information about Special Software Packages	103
9.2	Virtual Consoles	110
9.3	Keyboard Mapping	111
9.4	Language and Country-Specific Settings	112

10	Printer Operation	117
10.1	The Workflow of the Printing System	119
10.2	Methods and Protocols for Connecting Printers	119
10.3	Installing the Software	120
10.4	Network Printers	121
10.5	Printing from the Command Line	123
10.6	Special Features in SUSE Linux Enterprise Server	124
10.7	Troubleshooting	126
11	Dynamic Kernel Device Management with udev	135
11.1	The /dev Directory	135
11.2	Kernel uevents and udev	136
11.3	Drivers, Kernel Modules and Devices	136
11.4	Booting and Initial Device Setup	137
11.5	Monitoring the Running udev Daemon	137
11.6	Influencing Kernel Device Event Handling with udev Rules	139
11.7	Persistent Device Naming	146
11.8	Files used by udev	146
11.9	For More Information	147
12	The X Window System	149
12.1	Manually Configuring the X Window System	149
12.2	Installing and Configuring Fonts	156
12.3	For More Information	162
13	Accessing File Systems with FUSE	163
13.1	Configuring FUSE	163
13.2	Available FUSE Plug-ins	163
13.3	For More Information	164
Part III	Mobile Computers	165
14	Mobile Computing with Linux	167
14.1	Laptops	167
14.2	Mobile Hardware	175
14.3	Cellular Phones and PDAs	176
14.4	For More Information	176

15	Wireless LAN	177
15.1	WLAN Standards	177
15.2	Operating Modes	178
15.3	Authentication	179
15.4	Encryption	180
15.5	Configuration with YaST	181
15.6	Utilities	186
15.7	Tips and Tricks for Setting Up a WLAN	186
15.8	Troubleshooting	187
15.9	For More Information	189
16	Power Management	191
16.1	Power Saving Functions	191
16.2	ACPI	192
16.3	Rest for the Hard Disk	196
16.4	Troubleshooting	198
16.5	For More Information	200
17	Using Tablet PCs	201
17.1	Installing Tablet PC Packages	202
17.2	Configuring Your Tablet Device	203
17.3	Using the Virtual Keyboard	204
17.4	Rotating Your Display	205
17.5	Using Gesture Recognition	205
17.6	Taking Notes and Sketching with the Pen	208
17.7	Troubleshooting	210
17.8	For More Information	211
Part IV	Services	213
18	Basic Networking	215
18.1	IP Addresses and Routing	218
18.2	IPv6—The Next Generation Internet	221
18.3	Name Resolution	231
18.4	Configuring a Network Connection with YaST	233
18.5	NetworkManager	256
18.6	Configuring a Network Connection Manually	258
18.7	smpppd as Dial-up Assistant	273

19	SLP Services in the Network	277
19.1	Installation	277
19.2	Activating SLP	278
19.3	SLP Front-Ends in SUSE Linux Enterprise Server	278
19.4	Installation over SLP	279
19.5	Providing Services via SLP	279
19.6	For More Information	280
20	Time Synchronization with NTP	281
20.1	Configuring an NTP Client with YaST	281
20.2	Manually Configuring ntp in the Network	285
20.3	Dynamic Time Synchronization at Runtime	285
20.4	Setting Up a Local Reference Clock	286
20.5	Clock Synchronization to an External Time Reference (ETR)	287
21	The Domain Name System	289
21.1	DNS Terminology	289
21.2	Installation	290
21.3	Configuration with YaST	290
21.4	Starting the Name Server BIND	300
21.5	The Configuration File /etc/named.conf	301
21.6	Zone Files	306
21.7	Dynamic Update of Zone Data	310
21.8	Secure Transactions	310
21.9	DNS Security	312
21.10	For More Information	312
22	DHCP	313
22.1	Configuring a DHCP Server with YaST	314
22.2	DHCP Software Packages	325
22.3	The DHCP Server dhcpd	326
22.4	For More Information	329
23	Using NetworkManager	331
23.1	Use Cases for NetworkManager	331
23.2	Enabling NetworkManager	332
23.3	Configuring Network Connections	333
23.4	Using KNetworkManager	336
23.5	Using GNOME NetworkManager Applet	340
23.6	NetworkManager and VPN	342
23.7	NetworkManager and Security	343

23.8	Frequently Asked Questions	344
23.9	Troubleshooting	346
23.10	For More Information	347
24	Samba	349
24.1	Terminology	349
24.2	Starting and Stopping Samba	351
24.3	Configuring a Samba Server	351
24.4	Configuring Clients	358
24.5	Samba as Login Server	359
24.6	Samba Server in the Network with Active Directory	360
24.7	For More Information	361
25	Sharing File Systems with NFS	363
25.1	Installing the Required Software	363
25.2	Importing File Systems with YaST	364
25.3	Importing File Systems Manually	365
25.4	Exporting File Systems with YaST	367
25.5	Exporting File Systems Manually	372
25.6	NFS with Kerberos	375
25.7	For More Information	375
26	File Synchronization	377
26.1	Available Data Synchronization Software	377
26.2	Determining Factors for Selecting a Program	379
26.3	Introduction to CVS	382
26.4	Introduction to rsync	384
26.5	For More Information	386
27	The Apache HTTP Server	387
27.1	Quick Start	387
27.2	Configuring Apache	389
27.3	Starting and Stopping Apache	404
27.4	Installing, Activating, and Configuring Modules	407
27.5	Getting CGI Scripts to Work	414
27.6	Setting Up a Secure Web Server with SSL	417
27.7	Avoiding Security Problems	423
27.8	Troubleshooting	425
27.9	For More Information	426

28	Setting up an FTP server with YaST	429
28.1	Starting the FTP server	430
28.2	FTP General Settings	431
28.3	FTP Performance Settings	432
28.4	Authentication	432
28.5	Expert Settings	433
28.6	For more information	433
29	The Squid Proxy Server	435
29.1	Some Facts about Proxy Caches	436
29.2	System Requirements	437
29.3	Starting Squid	439
29.4	The /etc/squid/squid.conf Configuration File	441
29.5	Configuring a Transparent Proxy	447
29.6	cachemgr.cgi	450
29.7	Cache Report Generation with Calamaris	452
29.8	For More Information	453
30	Web Based Enterprise Management using SFCB	455
30.1	Introduction and Basic Concept	455
30.2	Setting Up SFCB	457
30.3	SFCB CIMOM Configuration	462
30.4	Advanced SFCB Tasks	475
30.5	For More Details	483
Part V	Troubleshooting	485
31	Help and Documentation	487
31.1	Documentation Directory	488
31.2	Man Pages	490
31.3	Info Pages	491
32	Common Problems and Their Solutions	493
32.1	Finding and Gathering Information	493
32.2	Installation Problems	496
32.3	Boot Problems	506
32.4	Login Problems	509
32.5	Network Problems	516
32.6	Data Problems	521
32.7	IBM System z: Using initrd as a Rescue System	534

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. A breakdown of available device technologies, high availability configurations, and advanced administration possibilities introduces the system to the administrator.

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. It offers a wide range of network services, such as DNS, DHCP, Web, proxy, and authentication services, and 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:

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.

Storage Administration Guide

Provides information about how to manage storage devices on a SUSE Linux Enterprise Server.

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 Server 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 product manuals in your installed system under `/usr/share/doc/manual` or in the help centers of your desktop. Find the latest documentation updates at <http://www.novell.com/documentation> where you can download PDF or HTML versions of the manuals for your product.

2 Feedback

Several feedback channels are available:

Bugs and Enhancement Requests

For services and support options available for your product, refer to <http://www.novell.com/services/>.

To report bugs for a product component, please use <http://support.novell.com/additional/bugreport.html>.

Submit enhancement requests at <https://secure-www.novell.com/rms/rmsTool?action=ReqActions.viewAddPage&return=www>.

User Comments

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

3 Documentation Conventions

The following typographical conventions are used in this manual:

- `/etc/passwd`: directory names and filenames
- *placeholder*: replace *placeholder* with the actual value
- `PATH`: the environment variable `PATH`
- `ls`, `--help`: commands, options, and parameters
- `user`: users or groups
- `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
- ► **amd64 em64t ipf**: This paragraph is only relevant for the specified architectures. The arrows mark the beginning and the end of the text block. ◀
 - **ipseries zseries**: This paragraph is only relevant for the specified architectures. The arrows mark the beginning and the end of the text block. ◀
- *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 the update applet is used to keep your system up-to-date. Refer to Section “Keeping the System Up-to-date” (Chapter 9, *Installing or Removing Software*, ↑*Deployment Guide*) for further information on the update applet. This chapter covers the alternative tool for updating software packages: YaST Online Update.

The current patches for SUSE® Linux Enterprise Server 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 Server, 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 9, *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 Server 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/>.

Novell 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.

Procedure 1.1 *Installing Patches with YaST Online Update*

- 1** Run *Software > Online Update* in YaST
- 2** All new patches (except the optional ones) that are currently available for your system are already marked for installation. Click *Accept* or *Apply* to automatically install them.
- 3** Confirm with *Finish* after the installation has completed. Your system is now up-to-date.

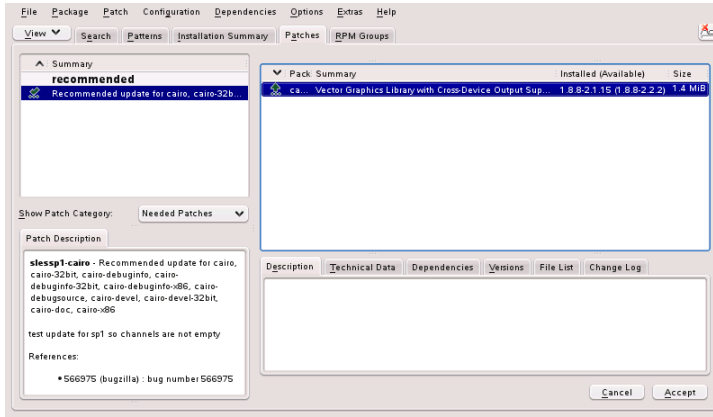
TIP: Disabling deltarpm

By default updates are downloaded as deltarpm. Since rebuilding rpm packages from deltarpm is a memory and CPU time consuming task, certain setups or hardware configurations might require you to disable the usage of deltarpm for performance sake. To disable the use of deltarpm edit the file `/etc/zypp/zypp.conf` and set `download.use_deltarpm` to `false`.

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.

Figure 1.1 *YaST Online Update*



The patch display lists the available patches for SUSE Linux Enterprise Server. 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)

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

Unneeded Patches

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

All Patches

All patches available for SUSE Linux Enterprise Server.

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* and *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

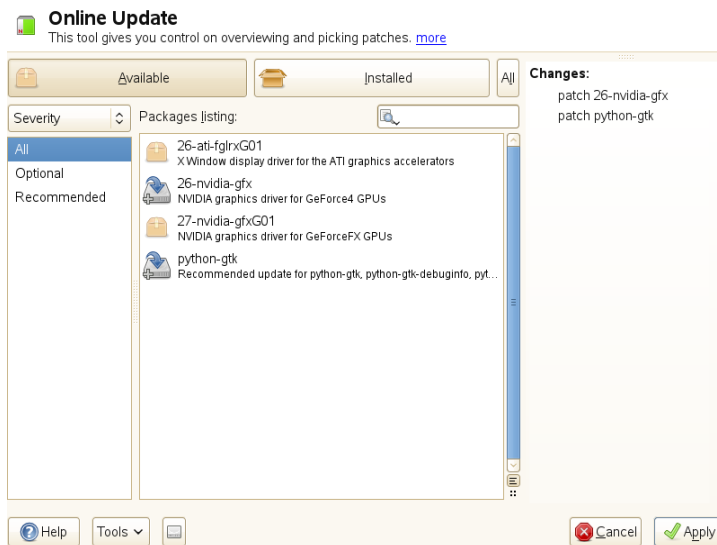
visible in the list until you mark it for installation. This will in fact not install the patch (because the package already is up-to-date), but mark the patch as having been installed.

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

Non-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 of the window. Here you can see a detailed patch description as well as the versions available. You can also choose to *Install* optional patches—security and recommended patches are already preselected for installation.

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. Such information can be, for example, the current kernel version being used, the hardware, RPM database, partitions and others. The result is used to help the Novell Support Center find your problem.

2.1 Novell Support Link Overview

Novell Support Link (NSL) is new to SUSE Linux Enterprise Server. 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 describe how to use `supportconfig` with YaST, with the command line and what other options 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 into the text field labeled *Novell 11 digit service request number*. Proceed with *Next*.
- 7 The information gathering begins. 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 the operation with *Next*.

2.2.2 Using Supportconfig Directly to Collect Information

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

- 1 Open a shell and become `root`.
- 2 Run `supportconfig` without any options. This gathers the default system information.
- 3 Wait for the tool to complete the operation.
- 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 the 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 tarball with the associated service request number. For example, if you have opened a service request with Novell and the tracking number is 12345678901, 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 tarball 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 tarball to your service request using the service request URL: <http://www.novell.com/center/eservice>.

5 Once the tarball is in the <ftp://ftp.novell.com/incoming> directory, it becomes automatically attached to your service request.

2.4 For More Information

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

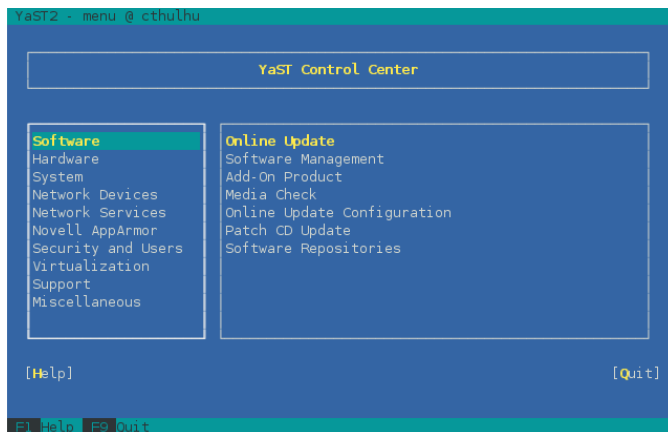
- `man supportconfig`—The manpage of supportconfig
- `man supportconfig.conf`—The manpage of the supportconfig configuration file
- <http://www.novell.com/communities/print/node/4097>—A Basic Server Health Check with Supportconfig
- <http://www.novell.com/communities/print/node/4827>—Create Your Own 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 the ncurses library to provide an easy pseudo-graphical user interface. The ncurses library is installed by default. The minimum supported size of the terminal emulator in which to run YaST is 80x25 characters.

Figure 3.1 *Main Window of YaST in Text Mode*



When you start YaST in text mode, the YaST Control Center appears (see Figure 3.1). The main window consists of three areas. The left frame features the categories to which the various modules belong. This frame is active when YaST is started and therefore

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

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

Various buttons or selection fields in the module contain a highlighted letter (yellow by default). Use Alt + highlighted_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 ← results in changing the active frame or the current selection list, as in the Control Center.

Buttons, Radio Buttons, and Check Boxes

To select buttons with empty square brackets (check boxes) or empty parentheses (radio buttons), press Space or Enter. Alternatively, radio buttons and check boxes can be selected directly with Alt + highlighted_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.

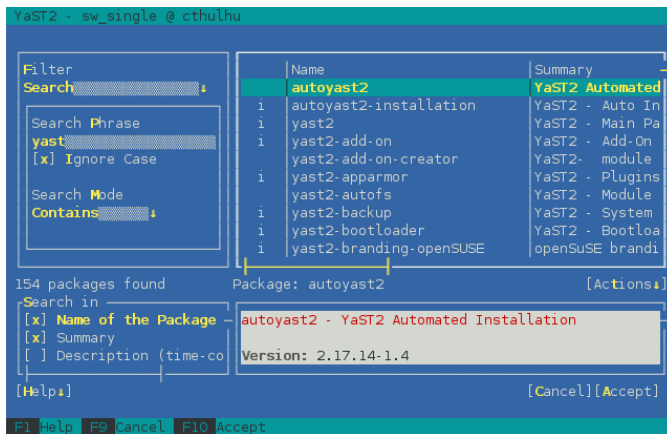
Function Keys

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

Using Navigation Tree in ncurses Mode

Some YaST modules use a navigation tree in the left part of the window to select configuration dialogs. Use the arrow keys (↑ and ↓) to navigate in the tree. Use Space to open or close tree items. In ncurses mode, Enter must be pressed after a selection in the navigation tree in order to show the selected dialog. This is an 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 the command line option `-i` to install the package:

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

or

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

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

If you need a command-line based software management utility with functionality beyond what YaST provides, consider using `zypper`. This new utility uses the same software management library that is also the foundation for the YaST package manager. The basic usage of `zypper` is covered in Section 4.1, “Using Zypper” (page 21).

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 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.
```


Managing Software with Command Line Tools

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

4.1 Using Zypper

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

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

4.1.1 General Usage

The general syntax of Zypper is:

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

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

```
zypper patch
```

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

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

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

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

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

```
zypper install mplayer
```

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

```
zypper search -t pattern
```

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

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

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

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

4.1.2 Installing and Removing Software with Zypper

To install or remove packages use the following commands:

```
zypper install package
zypper remove package
```

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

by the exact package name

```
zypper in MozillaFirefox
```

by repository alias and package name

```
zypper in mozilla:MozillaFirefox
```

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

by package name using wildcards

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

```
zypper in Moz*
```

by capability

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

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

by capability and/or architecture and/or version

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

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

by path

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

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

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

```
zypper install emacs -vim
```

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

```
zypper remove emacs +vim
```

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

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

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

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

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

WARNING: Do not Remove Mandatory System Packages

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

Installing Source Packages

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

```
zypper source-install package_name
```

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

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

Of course, this will only work if you have the repository with the source packages enabled in your repository list (it is added by default, but not enabled). See Section 4.1.4, “Managing Repositories with Zypper” (page 27) for details on repository management.

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

```
zypper search -t srcpackage
```

Utilities

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

```
zypper verify
```

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

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

4.1.3 Updating Software with Zypper

There are three different ways to update software using Zypper: by installing patches, by installing a new version of a package or by updating the entire distribution. The latter is achieved with the `zypper dist-upgrade` command which is discussed in Section “Distribution Upgrade with zypper” (Chapter 7, *Updating SUSE Linux Enterprise*, ↑*Deployment Guide*).

Installing Patches

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

```
zypper patch
```

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

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

```
zypper patch-check
```

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

```
~ # zypper patch-check  
Loading repository data...
```

```
Reading installed packages...
5 patches needed (1 security patch)
```

`zypper list-patches`

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

```
~ # zypper list-updates
Loading repository data...
Reading installed packages...
S | Repository | Name | Current | Available | Arch
--+-+-----+-----+-----+-----+-----+
v | Updates   | update-test-interactive | 0-2.35 | 0-9999.1.2 | noarch
v | Updates   | update-test-optional    | 0-2.35 | 0-9999.1.2 | noarch
v | Updates   | update-test-reboot-needed | 0-2.35 | 0-9999.1.2 | noarch
v | Updates   | update-test-relogin-suggested | 0-2.35 | 0-9999.1.2 | noarch
v | Updates   | update-test-security     | 0-2.35 | 0-9999.1.2 | noarch
```

`zypper patches`

Lists all patches available for SUSE Linux Enterprise Server, regardless of whether they are already installed or apply to your installation.

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

`-b`

Lists all needed patches for Bugzilla issues.

`--bugzilla [=number]`

Lists needed patches for the Bugzilla issue with the specified number.

To install a patch for a specific issue, use command:

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

Installing Updates

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

```
zypper update
```

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

```
zypper update package
zypper install package
```

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

```
zypper list-updates
```

NOTE: Differences between zypper update and zypper dist-upgrade

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

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

To upgrade your installation to a new product version use `zypper dist-upgrade` with the required repositories (see Section 4.1.4, “Managing Repositories with Zypper” (page 27) for details). This command ensures that all packages will be installed from the repositories currently enabled. This rule is enforced, so packages might change vendor or architecture or even might get downgraded. All packages that have unfulfilled dependencies after the upgrade will be uninstalled.

4.1.4 Managing Repositories with Zypper

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

zypper repos

The result will look similar to the following output:

```
# | Alias | Name
  | Enabled | Refresh
```

```

1 | SUSE-Linux-Enterprise-Server 11-0 | SUSE-Linux-Enterprise-Server 11-0
  | Yes          | No
2 | SLES-11-Updates                    | SLES 11 Online Updates
  | Yes          | Yes
3 | broadcomdrv                        | Broadcom Drivers
  | Yes          | No

```

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.

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

Adding Repositories

To add a repository, run

```
zypper addrepo URI Alias
```

URI can either be an Internet repository, a network resource, a directory or a CD or DVD (see <http://en.opensuse.org/Libzypp/URI> for details). The *Alias* is a shorthand and unique identifier of the repository. You can freely choose it, with the only exception that it has to be unique. Zypper will issue a warning if you specify an alias that is already in use. To make working with repositories more convenient, use short and easy-to-remember aliases.

Removing Repositories

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 3rd entry from the example, use the following command:

```
zypper removerepo 3
```

Modifying Repositories

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

```
zypper mr -er -p 20 'updates'
```

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

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

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

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

4.1.5 Querying Repositories and Packages with Zypper

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

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

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

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

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

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

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

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

The `what-provides package` is similar to `rpm -q --whatprovides package`, but `rpm` is only able to query the RPM database (that is the database of all installed packages). `Zypper`, on the other hand, will tell you about providers of the capability from any repository, not only those that are installed.

4.2 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`.

4.2.1 Verifying Package Authenticity

RPM packages have a GnuPG signature. The command `rpm --checksig package-1.2.3.rpm` can be used to verify the signature of an RPM package to determine whether it originates from SUSE or from another trustworthy facility. This is especially recommended for update packages from the Internet.

4.2.2 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, you risk compromising the integrity of the system and possibly jeopardize the ability to update the system.

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

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

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

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

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

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

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

4.2.3 RPM and Patches

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

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

Is the patch RPM suitable for my system?

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

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

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

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

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

Which files are replaced by the patch?

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

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

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

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

How can a patch RPM be installed in the system?

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

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

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

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

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

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

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

NOTE: Official Updates for SUSE Linux Enterprise Server

In order to make the download size of updates as small as possible, official updates for SUSE Linux Enterprise Server are not provided as Patch RPMs, but as Delta RPM packages (see Section 4.2.4, “Delta RPM Packages” (page 34) for details).

4.2.4 Delta RPM Packages

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

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

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

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

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

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

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

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

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

4.2.5 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 4.1, “The Most Important RPM Query Options” (page 35).

Table 4.1 *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 4.1, “`rpm -q -i wget`” (page 36).

Example 4.1 *rpm -q -i wget*

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

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

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

results in:

```
rpm-4.4.2.3-45.5
wget-1.11.4-1.70
```

If only part of the filename is known, use a shell script as shown in Example 4.2, “Script to Search for Packages” (page 36). Pass the partial filename to the script shown as a parameter when running it.

Example 4.2 *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.

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

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

Table 4.2 *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
M	Mode (permissions and file type)

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

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

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

4.2.6 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/rpmrc`):

SOURCES

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

SPECS

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

BUILD

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

RPMS

where the completed binary packages are stored

SRPMS

here are the source RPMs

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

WARNING

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

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

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

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

`-bp`

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

`-bc`

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

`-bi`

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

`-bb`

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

`-ba`

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

`--short-circuit`

Skip some steps.

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

4.2.7 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.

4.2.8 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 9, *Installing or Removing Software* (↑*Deployment Guide*)).

Bash and Bash Scripts

These days many people use computers with a graphical user interface (GUI) like KDE or GNOME. Although they offer lots of features, their use is limited when it comes to the execution of automatical tasks. Shells are a good addition to GUIs and this chapter gives you an overview of some aspects of shells, in this case Bash.

5.1 What is “The Shell”?

Traditionally, *the* shell is Bash (Bourne again Shell). When this chapter speaks about “the shell” it means Bash. There are actually more available shells than Bash (ash, csh, ksh, zsh, ...), each employing different features and characteristics. If you need further information about other shells, search for *shell* in YaST.

5.1.1 Knowing The Bash Configuration Files

A shell can be invoked as an:

1. interactive login shell. This is used when logging in to a machine, invoking Bash with the `--login` option or when logging in to a remote machine with SSH.
2. “ordinary” interactive shell. This is normally the case when starting xterm, konsole, gnome-terminal or similar tools.
3. non-interactive shell. This is used when invoking a shell script at the commandline.

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

Table 5.1 *Bash Configuration Files for Login Shells*

File	Description
<code>/etc/profile</code>	Do not modify this file, otherwise your modifications can be destroyed during your next update!
<code>/etc/profile.local</code>	Use this file if you extend <code>/etc/profile</code>
<code>/etc/profile.d/</code>	Contains system-wide configuration files for specific programs
<code>~/.profile</code>	Insert user specific configuration for login shells here

Table 5.2 *Bash Configuration Files for Non-Login Shells*

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

Additionally, Bash uses some more files:

Table 5.3 *Special Files for Bash*

File	Description
<code>~/.bash_history</code>	Contains a list of all commands you have been typing
<code>~/.bash_logout</code>	Executed when logging out

5.1.2 The Directory Structure

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

Table 5.4 *Overview of a Standard Directory Tree*

Directory	Contents
/	Root directory—the starting point of the directory tree.
/bin	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.
/boot	Static files of the boot loader.
/dev	Files needed to access host-specific devices.
/etc	Host-specific system configuration files.
/home	Holds the home directories of all users who have accounts on the system. However, <code>root</code> 's home directory is not located in <code>/home</code> but in <code>/root</code> .
/lib	Essential shared libraries and kernel modules.
/media	Mount points for removable media.
/mnt	Mount point for temporarily mounting a file system.
/opt	Add-on application software packages.
/root	Home directory for the superuser <code>root</code> .
/sbin	Essential system binaries.

Directory	Contents
<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 of which files and subdirectories can be found in the directories:

`/bin`

Contains the basic shell commands that may be used both by `root` and by other users. These commands include `ls`, `mkdir`, `cp`, `mv`, `rm` and `rmdir`. `/bin` also contains Bash, the default shell in SUSE Linux Enterprise Server.

`/boot`

Contains data required for booting, such as the boot loader, the kernel, and other data that is used before the kernel begins executing user-mode programs.

`/dev`

Holds device files that represent hardware components.

`/etc`

Contains local configuration files that control the operation of programs like the X Window System. The `/etc/init.d` subdirectory contains scripts that are executed during the boot process.

`/home/username`

Holds the private data of every user who has an account on the system. The files located here can only be modified by their owner or by the system administrator. By default, your e-mail directory and personal desktop configuration are located here in the form of hidden files and directories. KDE users find the personal confi-

guration data for their desktop in `.kde` or `.kde4`. GNOME users find it in `.gconf`.

NOTE: Home Directory in a Network Environment

If you are working in a network environment, your home directory may be mapped to a directory in the file system other than `/home`.

`/lib`

Contains the essential shared libraries needed to boot the system and to run the commands in the root file system. The Windows equivalent for shared libraries are DLL files.

`/media`

Contains mount points for removable media, such as CD-ROMs, USB sticks and digital cameras (if they use USB). `/media` generally holds any type of drive except the hard drive of your system. As soon as your removable medium has been inserted or connected to the system and has been mounted, you can access it from here.

`/mnt`

This directory provides a mount point for a temporarily mounted file system. `root` may mount file systems here.

`/opt`

Reserved for the installation of additional software. Optional software and larger add-on program packages can be found here. KDE3 is located here, whereas KDE4 and GNOME have moved to `/usr` now.

`/root`

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

`/sbin`

As the `s` indicates, this directory holds utilities for the superuser. `/sbin` contains the binaries essential for booting, restoring and recovering the system in addition to the binaries in `/bin`.

`/srv`

Holds data for services provided by the system, such as FTP and HTTP.

`/tmp`

This directory is used by programs that require temporary storage of files.

`/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 with the Filesystem Hierarchy Standard (FHS). This directory contains all application programs and establishes a secondary hierarchy in the file system. KDE4 and GNOME are also located here. `/usr` holds a number of subdirectories, such as `/usr/bin`, `/usr/sbin`, `/usr/local`, and `/usr/share/doc`.

`/usr/bin`

Contains generally accessible programs.

`/usr/sbin`

Contains 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 related to the setup and operation of Linux software.

`/var`

Whereas `/usr` holds static, read-only data, `/var` is for data which is written during system operation and thus is variable data, such as log files or spooling data. For

an overview of the most important log files you can find under `/var/log/`, refer to Table 32.1, “Log Files” (page 494).

5.2 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 5.1 *A Shell Script Printing a Text*

```
#!/bin/sh ❶  
# Output the following line: ❷  
echo "Hello World" ❸
```

- ❶ The first line begins with the *Shebang* characters (`#!`) which is an indicator that this file is a script. The script is executed with the specified interpreter after the Shebang, in this case `/bin/sh`.
- ❷ The second line is a comment beginning with the hash sign. It is recommended to comment difficult lines to remember what they do.
- ❸ The third line uses the built-in command `echo` to print the corresponding text.

Before you can run this script you need some prerequisites:

1. Every script should contain a Shebang line (this is already the case with our example above.) If a script does not have this line, you have to call the interpreter manually.
2. You can save the script wherever you want. However, it is a good idea to save it in a directory where the shell can find it. The search path in a shell is determined by the environment variable `PATH`. Usually a normal user does not have write access to `/usr/bin`. Therefore it is recommended to save your scripts in the users' directory `~/bin/`. The above example gets the name `hello.sh`.
3. The script needs executable permissions. Set the permissions with the following command:

```
chmod +x ~/bin/hello.sh
```

If you have fulfilled all of the above prerequisites, you can execute the script in the following ways:

1. **As Absolute Path** The script can be executed with an absolute path. In our case, it is `~/bin/hello.sh`.
2. **Everywhere** If the `PATH` environment variable contains the directory where the script is located, you can execute the script just with `hello.sh`.

5.3 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 the variable:

```
read a < foo
```

Command1 | Command2

Redirects the output of the left command as input for the right command. For example, the `cat` command outputs the content of the `/proc/cpuinfo` file. This output is used by `grep` to filter only those lines which contain `cpu`:

```
cat /proc/cpuinfo | grep cpu
```

Every channel has a *file descriptor*: 0 (zero) for standard input, 1 for standard output and 2 for standard error. It is allowed to insert this file descriptor before a `<` or `>` character. For example, the following line searches for a file starting with `foo`, but suppresses its errors by redirecting it to `/dev/null`:

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

5.4 Using Aliases

An alias is a 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`. Remove your alias with `unalias` and the corresponding alias name.

5.5 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 to know the value of a variable, insert the name of your variable as an argument:

```
printenv PATH
```

A variable, be it global or local, can also be viewed with `echo`:

```
echo $PATH
```

To set a local variable, use a variable name followed by the equal sign, followed by the value:

```
PROJECT="SLED"
```

Do not insert spaces around the equal sign, otherwise you get an error. To set an environment variable, use `export`:

```
export NAME="tux"
```

To remove a variable, use `unset`:

```
unset NAME
```

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

Table 5.5 *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

5.5.1 Using Argument Variables

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

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

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

5.5.2 Using Variable Substitution

Variable substitutions apply a pattern to the content of a variable either from the left or right side. The following list contains the possible syntax forms:

`${VAR#pattern}`
removes the shortest possible match from the left:

```
file=/home/tux/book/book.tar.bz2
echo ${file#*/}
home/tux/book/book.tar.bz2
```

`${VAR##pattern}`
removes the longest possible match from the left:

```
file=/home/tux/book/book.tar.bz2
echo ${file##*/}
book.tar.bz2
```

`${VAR%pattern}`
removes the shortest possible match from the right:

```
file=/home/tux/book/book.tar.bz2
echo ${file%.*}
/home/tux/book/book.tar
```

```
${VAR%%pattern}
```

removes the longest possible match from the right:

```
file=/home/tux/book/book.tar.bz2
echo ${file%%.*}
/home/tux/book/book
```

```
${VAR/pattern_1/pattern_2}
```

substitutes the content of *VAR* from the *pattern_1* with *pattern_2*:

```
file=/home/tux/book/book.tar.bz2
echo ${file/tux/wilber}
/home/wilber/book/book.tar.bz2
```

5.6 Grouping And Combining Commands

Shells allow you to concatenate and group commands for conditional execution. Each command returns an exit code which determines the success or failure of its operation. If it is 0 (zero) the command was successful, everything else marks an error which is specific to the command.

The following list shows, how commands can be grouped:

```
Command1 ; Command2
```

executes the commands in sequential order. The exit code is not checked. The following line displays the content of the file with `cat` and then prints its file properties with `ls` regardless of their exit codes:

```
cat filelist.txt ; ls -l filelist.txt
```

```
Command1 && Command2
```

runs the right command, if the left command was successful (logical AND). The following line displays the content of the file and prints its file properties only, when the previous command was successful (compare it with the previous entry in this list):

```
cat filelist.txt && ls -l filelist.txt
```

```
Command1 || Command2
```

runs the right command, when the left command has failed (logical OR). The following line creates only a directory in `/home/wilber/bar` when the creation of the directory in `/home/tux/foo` has failed:

```
mkdir /home/tux/foo || mkdir /home/wilber/bar
```

```
funcname() { ... }
```

creates a shell function. You can use the positional parameters to access its arguments. The following line defines the function `hello` to print a short message:

```
hello() { echo "Hello $1"; }
```

You can call this function like this:

```
hello Tux
```

which prints:

```
Hello Tux
```

5.7 Working with Common Flow Constructs

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

5.7.1 The `if` Control Command

The `if` command is used to check expressions. For example, the following code tests whether the current user is Tux:

```
if test $USER = "tux"; then
    echo "Hello Tux."
else
    echo "You are not Tux."
fi
```

The test expression can be as complex or simple as possible. The following expression checks if the file `foo.txt` exists:

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

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

The test expression can also be abbreviated in angled brackets:

```
if [ -e /tmp/foo.txt ] ; then
    echo "Found foo.txt"
fi
```

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

5.7.2 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
```

5.8 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

6

SUSE® Linux Enterprise Server is available for several 64-bit platforms. This does not necessarily mean that all the applications included have already been ported to 64-bit platforms. SUSE Linux Enterprise Server 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 Server 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 Server for the 64-bit platforms ia64, ppc64, System z and x86_64 is designed so that existing 32-bit applications run in the 64-bit environment “out-of-the-box.” The corresponding 32-bit platforms are x86 for ia64, ppc for ppc64, and x86 for x86_64. 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. The current ppc64 system runs most applications in 32-bit mode, but you can run 64-bit applications.

6.1 Runtime Support

IMPORTANT: Conflicts between Application Versions

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

An exception to this rule is PAM (pluggable authentication modules). SUSE Linux Enterprise Server 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 which contain data content that does not depend on the word size are not moved. This scheme conforms to LSB (Linux Standards Base) and FHS (File System Hierarchy Standard).

► **ipf:** The 64-bit libraries for `ia64` are located in the standard `lib` directories, there is neither a `lib64` directory nor a `lib32` directory. `ia64` executes 32-bit x86 code under an emulation. A set of basic libraries is installed in `/emul/ia32-linux/lib` and `/emul/ia32-linux/usr/lib`. ◀

6.2 Software Development

All 64-bit architectures support the development of 64-bit objects. The level of support for 32-bit compiling depends on the architecture. These are the various implementation options for the tool chain from GCC (GNU Compiler Collection) and binutils, which include the assembler `as` and the linker `ld`:

Biarch Compiler

Both 32-bit and 64-bit objects can be generated with a biarch development tool chain. The compilation of 64-bit objects is the default on almost all platforms. 32-bit objects can be generated if special flags are used. This special flag is `-m32` for GCC. The flags for the binutils are architecture-dependent, but GCC transfers the correct flags to linkers and assemblers. A biarch development tool chain currently exists for `amd64` (supports development for `x86` and `amd64` instructions), for `System z` and for `ppc64`. 32-bit objects are normally created on the `ppc64` platform. The `-m64` flag must be used to generate 64-bit objects.

No Support

SUSE Linux Enterprise Server does not support the direct development of 32-bit software on all platforms. To develop applications for `x86` under `ia64`, use the corresponding 32-bit version of SUSE Linux Enterprise Server.

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 Server environment is designed according to this principle. In the case of manually updated libraries, resolve these issues yourself.

6.3 Software Compilation on Biarch Platforms

To develop binaries for the other architecture on a biarch architecture, the respective libraries for the second architecture must additionally be installed. These packages are called `rpmname-32bit` or `rpmname-x86` (for `ia64`) if the second architecture is a 32-bit architecture or `rpmname-64bit` if the second architecture is a 64-bit architecture. 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` or `rpmname-devel-64bit`.

For example, to compile a program that uses `libaio` on a system whose second architecture is a 32-bit architecture (`x86_64` or System z), you need the following RPMs:

`libaio-32bit`

32-bit runtime package

`libaio-devel-32bit`

Headers and libraries for 32-bit development

`libaio`

64-bit runtime package

`libaio-devel`

64-bit development headers and libraries

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

The following example refers to an `x86_64` system with `x86` as the second architecture. Examples for `ppc64` with `ppc` as the second architecture would be similar. This example does not apply to `ia64` where you cannot build 32-bit packages.

1 Use the 32-bit compiler:

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

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

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

3 Set the assembler to generate 32-bit objects:

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

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

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

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

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

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

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

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

An example `configure` call to compile a native 32-bit application on `x86_64`, `ppc64` or `System z` could appear as follows:

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

6.4 Kernel Specifications

The 64-bit kernels for `x86_64`, `ppc64` and `System z` 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 on non-`ppc64` platforms as 64-bit programs to function properly. On `IBM System z`, not all `ioctl`s are available in the 32-bit kernel ABI.

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 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.

7.1 The Linux Boot Process

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

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

remaining part of the boot process. Therefore, the first 512 bytes on the first hard disk are referred to as the *Master Boot Record* (MBR). The boot loader then passes control to the actual operating system, in this case, the Linux kernel. More information about GRUB, the Linux boot loader, can be found in Chapter 8, *The Boot Loader GRUB* (page 81). For a network boot, the BIOS acts as the boot loader. It gets the image to start from the boot server then starts the system. This is completely independent of local hard disks.

3. **Kernel and initramfs** To pass system control, the boot loader loads both the kernel and an initial RAM-based file system (initramfs) into memory. The contents of the initramfs can be used by the kernel directly. initramfs contains a small executable called `init` that handles the mounting of the real root file system. If special hardware drivers are needed before the mass storage can be accessed, they must be in initramfs. For more information about initramfs, refer to Section 7.1.1, “initramfs” (page 66). If the system does not have a local hard disk, the initramfs must provide the root file system to the kernel. This can be done with the help of a network block device like iSCSI or SAN, but it is also possible to use NFS as the root device.
4. **init on initramfs** This program performs all actions needed to mount the proper root file system, like providing kernel functionality for the needed file system and device drivers for mass storage controllers with `udev`. After the root file system has been found, it is checked for errors and mounted. If this is successful, the initramfs is cleaned and the `init` program on the root file system is executed. For more information about `init`, refer to Section 7.1.2, “init on initramfs” (page 67). Find more information about `udev` in Chapter 11, *Dynamic Kernel Device Management with udev* (page 135).
5. **init** `init` handles the actual booting of the system through several different levels providing different functionality. `init` is described in Section 7.2, “The init Process” (page 69).

7.1.1 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` creates an `initrd`. In SUSE® Linux Enterprise Server, 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.

7.1.2 `init` on `initramfs`

The main purpose of `init` on `initramfs` is to prepare the mounting of and access to the real root file system. Depending on your system configuration, `init` 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 15, *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.

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

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

Finding the Installation Medium

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

Initiating Hardware Recognition and Loading Appropriate Kernel Modules

As mentioned in Section 7.1.1, “`initramfs`” (page 66), the boot process starts with a minimum set of drivers that can be used with most hardware configurations. `init` starts an initial hardware scanning process that determines the set of drivers suitable for your hardware configuration. The names of the modules needed for the boot

process are written to `INITRD_MODULES` in `/etc/sysconfig/kernel`. These names are used to generate a custom `initramfs` that is needed to boot the system. If the modules are not needed for boot but for coldplug, the modules are written to `/etc/sysconfig/hardware/hwconfig-*`. All devices that are described with configuration files in this directory are initialized in the boot process.

Loading the Installation System or Rescue System

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

Starting YaST

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

7.2 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 7.2.1, “Runlevels” (page 70)). 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 7.2.2, “Init Scripts” (page 72)).

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

7.2.1 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 7.1, “Available Runlevels” (page 70). 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 7.1 *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 Server standard installations. Users are prompted for login with a graphical interface or the default user is logged in automatically.

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

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

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

1. The administrator (`root`) requests `init` to change to a different runlevel by entering `telinit 5`.
2. `init` checks the current runlevel (`runlevel`) and determines it should start `/etc/init.d/rc` with the new runlevel as a parameter.
3. Now `rc` calls the stop scripts of the current runlevel for which there is no start script in the new runlevel. In this example, these are all the scripts that reside in `/etc/init.d/rc3.d` (the old runlevel was 3) and start with a `K`. The number following `K` specifies the order to run the scripts with the `stop` parameter, because there are some dependencies to consider.
4. The last things to start are the start scripts of the new runlevel. In this example, these are in `/etc/init.d/rc5.d` and begin with an `S`. Again, the number that follows the `S` determines the sequence in which the scripts are started.

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

7.2.2 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`). For IBM System z systems, this is the case only during the boot process or if an immediate system shutdown is initiated (power failure or via “signal quiesce”). 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 run-

level are called through symbolic links from one of the subdirectories (`/etc/init.d/rc0.d` to `/etc/init.d/rc6.d`). This is just for reasons of clarity and avoids duplicate scripts if they are used in several runlevels. Because every script can be executed as both a start and a stop script, these scripts must understand the parameters `start` and `stop`. The scripts also understand the `restart`, `reload`, `force-reload`, and `status` options. These different options are explained in Table 7.2, “Possible init Script Options” (page 73). Scripts that are run directly by `init` do not have these links. They are run independently from the runlevel when needed.

Table 7.2 *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 names that start with `S`. There, the file systems are checked and loop devices are configured if needed. The system time is also set. If an error occurs while automatically checking and repairing the file system, the system administrator can intervene after first entering the root password. The last executed script is `boot.local`.

`boot.local`

Here, enter additional commands to execute at boot before changing into a runlevel. It is similar 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 up. Edit such scripts with great care and, if possible, subject them to heavy testing in the multiuser environment. Find useful information about init scripts in Section 7.2.1, “Runlevels” (page 70).

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 7.1, “A Minimal INIT INFO Block” (page 75).

Example 7.1 *A Minimal INIT INFO Block*

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

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

To create the links from the runlevel directories (`/etc/init.d/rc?.d/`) to the corresponding scripts in `/etc/init.d/`, enter the command `insserv new-script-name`. The `insserv` program evaluates the `INIT INFO` header to create

the necessary links for start and stop scripts in the runlevel directories (`/etc/init.d/rc?.d/`). The program also takes care of the correct start and stop order for each runlevel by including the necessary numbers in the names of these links. If you prefer a graphical tool to create such links, use the runlevel editor provided by YaST, as described in Section 7.2.3, “Configuring System Services (Runlevel) with YaST” (page 76).

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.

7.2.3 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 Server system is runlevel 5 (full multiuser mode with network and X). A suitable alternative might be runlevel 3 (full multiuser mode with network).

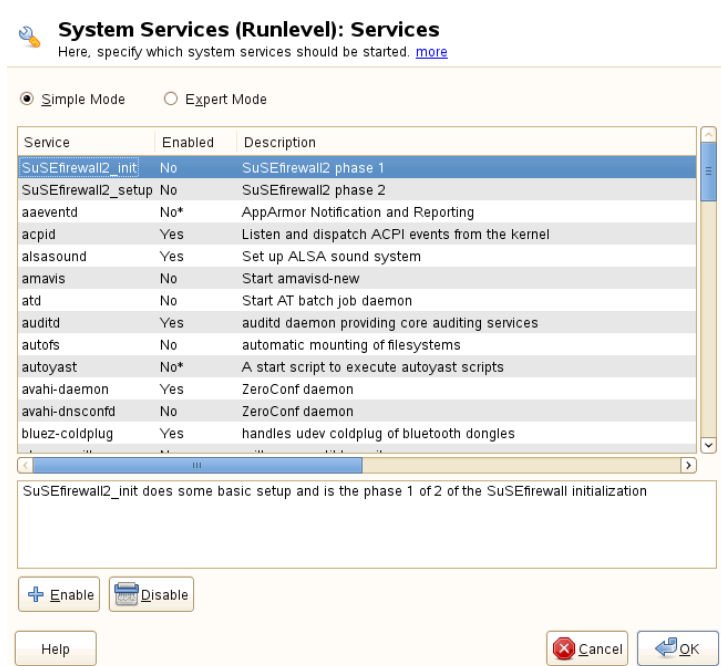
This YaST dialog allows the selection of one of the runlevels (as listed in Table 7.1, “Available Runlevels” (page 70)) as the new default. Additionally, use the table in this window to enable or disable individual services and daemons. The table lists the services

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 7.1 *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.

7.3 System Configuration via `/etc/sysconfig`

The main configuration of SUSE Linux Enterprise Server 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.

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

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

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

Do not modify the `/etc/sysconfig` files without 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 7.2 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.

7.3.2 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 Server. A special YaST module is available for configuring all settings. If you are not familiar with the subject of booting in Linux, read the following sections to acquire some background information. This chapter also describes some of the problems frequently encountered when booting with GRUB and their solutions.

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 7, *Booting and Configuring a Linux System* (page 65). A boot loader represents the interface between the machine (BIOS) and the operating system (SUSE Linux Enterprise Server). 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 basic important data of the file system. In contrast, the boot sectors of Linux partitions are initially empty after setting up a file system other than XFS. Therefore, a Linux partition is not bootable by itself, even if it contains a kernel and a valid root file system. A boot sector with valid code for booting the system has the same magic number as the MBR in its last two bytes (AA55).

8.1 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 four files that are described below:

`/boot/grub/menu.lst`

This file contains all information about partitions or operating systems that can be booted with GRUB. Without this information, the GRUB command line prompts the user for how to proceed (see Section “Editing Menu Entries during the Boot Procedure” (page 88) 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.

`/etc/sysconfig/bootloader`

This file is read by the perl-bootloader library which is used when configuring the bootloader with YaST and every time a new kernel is installed. It contains configuration options (such as kernel parameters) that will be added by default to the bootloader configuration file.

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

In GRUB, all boot parameters can be changed prior to booting. For example, errors made when editing the menu file can be corrected in this way. Boot commands can also be entered interactively at a kind of input prompt (see Section “Editing Menu Entries during the Boot Procedure” (page 88)). GRUB offers the possibility of determining the location of the kernel and the `initrd` prior to booting. In this way, you can even boot an installed operating system for which no entry exists in the boot loader configuration.

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

8.1.1 The File `//boot/grub/menu.lst`

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

Every time the system is booted, GRUB loads the menu file from the file system. For this reason, GRUB does not need to be reinstalled after every change to the file. Use the YaST boot loader to modify the GRUB configuration as described in Section 8.2, “Configuring the Boot Loader with YaST” (page 93).

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 85). 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 86).

Naming Conventions for Hard Disks and Partitions

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

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

```
(hd0,0)  first primary partition of the first hard disk
(hd0,1)  second primary partition
(hd0,2)  third primary partition
```

```
(hd0,3)    fourth primary partition (usually an extended partition)
(hd0,4)    first logical partition
(hd0,5)    second logical partition
```

Being dependent on BIOS devices, GRUB does not distinguish between 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 8.1.2, “The File `device.map`” (page 89).

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 Server. 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 88).

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 32.3, “US Keyboard Layout” (page 502).

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.

8.1.2 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/disk-by-id/DISK1 ID
(hd1)  /dev/disk-by-id/DISK2 ID
```

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.

NOTE: Maximum Number of Hard Disks

To address a hard disk, GRUB uses BIOS services. This is done via the software interrupt `Int13h`. Since `Int13h` is limited to handling a maximum number of eight disks, GRUB can only boot from the disks handled by `Int13h`, even if there are more disks present (which is often the case on multipath systems). The `device.map` file created during the installation will therefore only contain a maximum number of the eight disks handled by `Int13h`.

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
```

8.1.3 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.

8.1.4 The File `/etc/sysconfig/bootloader`

This configuration file is only used when configuring the bootloader with YaST and every time a new kernel is installed. It is evaluated by the `perl-bootloader` library which modifies the bootloader configuration file (for example `/boot/grub/menu.lst` for GRUB) accordingly. `/etc/sysconfig/bootloader` is not a GRUB specific configuration file - the values are applied to any bootloader installed on SUSE Linux Enterprise Server.

NOTE: Bootloader Configuration after a Kernel Update

Every time a new kernel is installed, the `perl bootloader` writes a new bootloader configuration file (for example `/boot/grub/menu.lst` for GRUB) using the defaults specified in `/etc/sysconfig/bootloadert`. If you are using a customized set of kernel parameters, make sure to adjust the relevant defaults in `/etc/sysconfig/bootloader` according to your needs.

`LOADER_TYPE`

Specifies the bootloader installed on the system (e.g. GRUB or LILO). Please do not modify, use YaST to change the bootloader— see Procedure 8.6, “Changing the Boot Loader Type” (page 97) for details.

DEFAULT_VGA / FAILSAFE_VGA / XEN_VGA

Screen resolution and color depth of the framebuffer used during booting are configured with the kernel parameter `vga`. These values define which resolution and color depth to use for the default boot entry, the failsafe and the XEN entry. The following values are valid:

Table 8.1 *Screen Resolution and Color Depth Reference*

	640x480	800x600	1024x768	1280x1024	1600x1200
8bit	0x301	0x303	0x305	0x307	0x31C
15bit	0x310	0x313	0x316	0x319	0x31D
16bit	0x311	0x314	0x317	0x31A	0x31E
24bit	0x312	0x315	0x318	0x31B	0x31F

DEFAULT_APPEND / FAILSAFE_APPEND / XEN_KERNEL_APPEND

Kernel parameters (other than `vga`) that are automatically appended to the default, failsafe and XEN boot entries in the bootloader configuration file.

CYCLE_DETECTION / CYCLE_NEXT_ENTRY

Configure whether to use boot cycle detection and if so, which alternative entry from `/boot/grub/menu.lst` (e.g. Failsafe) to boot in case of a reboot cycle. See `/usr/share/doc/packages/bootcycle/README` for detailed information.

8.1.5 Setting a Boot Password

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

IMPORTANT: Boot Password and Splash Screen

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

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

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

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

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

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

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

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

```
title linux
    kernel (hd0,4)/vmlinuz root=/dev/sda7 vga=791
    initrd (hd0,4)/initrd
    lock
```

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

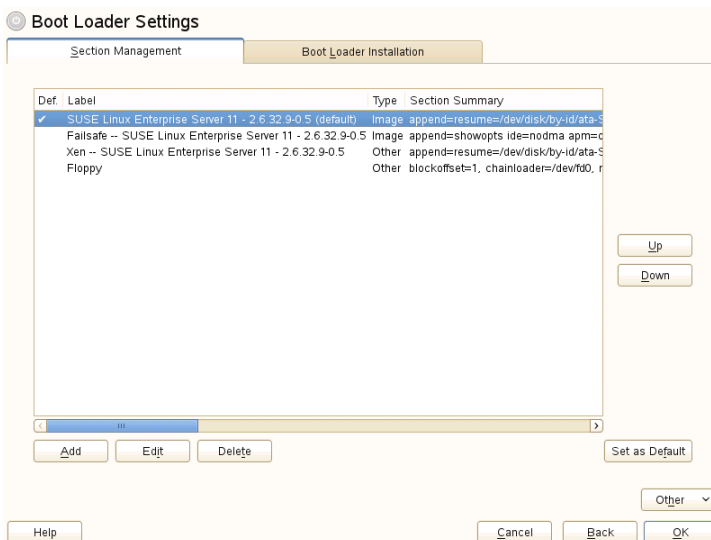
```
Error 32: Must be authenticated
```

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

8.2 Configuring the Boot Loader with YaST

The easiest way to configure the boot loader in your SUSE Linux Enterprise Server system is to use the YaST module. In the YaST Control Center, select *System > Boot Loader*. As in Figure 8.1, “Boot Loader Settings” (page 93), this shows the current boot loader configuration of your system and allows you to make changes.

Figure 8.1 *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 8.1, “Booting with GRUB” (page 82) 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 8.1, “Bootting with GRUB” (page 82) 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*.

8.2.1 Adjusting the Default Boot Entry

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

Procedure 8.1 *Setting the Default System*

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

8.2.2 Modifying the Boot Loader Location

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

Procedure 8.2 *Changing the Boot Loader Location*

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

Boot from Master Boot Record

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

Boot from Root Partition

This installs the boot loader in the boot sector of the / partition (this is the default).

Boot from Boot Partition

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

Boot from Extended Partition

This installs the boot loader in the extended partition container.

Custom Boot Partition

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

- 2 Click *OK* to apply your changes.

8.2.3 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 8.3 *Changing the Boot Loader Time-Out*

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

8.2.4 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 8.4 *Setting a Boot Loader Password*

- 1 Open the *Boot Loader Installation* tab.

- 2 Click *Boot Loader Options*.
- 3 Activate the *Protect Boot Loader with Password* option with a click and type in your *Password* twice.
- 4 Click *OK* twice to save the changes.

8.2.5 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 8.1.2, “The File device.map” (page 89)). To do so, proceed as follows:

Procedure 8.5 *Setting the Disk Order*

- 1 Open the *Boot Loader Installation* tab.
- 2 Click *Boot Loader Installation Details*.
- 3 If more than one disk is listed, select a disk and click *Up* or *Down* to reorder the displayed disks.
- 4 Click *OK* two times to save the changes.

8.2.6 Configuring Advanced Options

Advanced boot options can be configured via *Boot Loader Installation > Boot Loader Options*. Normally, it should not be necessary to change the default settings.

Set Active Flag in Partition Table for Boot Partition

Activates the partition that contains the boot loader. Some legacy operating systems (such as Windows 98) can only boot from an active partition.

Write Generic Boot Code to MBR

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

Debugging Flag

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

Hide Boot Menu

Hides the boot menu and boots the default entry.

Use Trusted GRUB

Starts the Trusted GRUB which supports trusted computing functionality.

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.

8.2.7 Changing Boot Loader Type

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

WARNING: LILO is unsupported

Using LILO is not recommended—it is unsupported on SUSE Linux Enterprise Server. Only use it in special cases.

Procedure 8.6 *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 Server.

Read Configuration Saved on Disk

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

4 Click *OK* two times to save the changes

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

NOTE: Custom Boot Loader

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

8.3 Uninstalling the Linux Boot Loader

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

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

8.4 Creating Boot CDs

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

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

Procedure 8.7 *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 a data file, but use the option for burning a CD image in your burning utility.

8.5 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 8, *The Boot Loader GRUB* (page 81) provides more information about this. However, if you prefer the text mode (the default in earlier versions) set `vga=normal`.

Completely Disabling the SUSE Screen

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

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.

8.6 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. In this case, update the BIOS.

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

System Containing Several Hard Disks Does Not Boot

During the installation, YaST may have incorrectly determined the boot sequence of the hard disks. For example, GRUB may regard the 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`.

8.7 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 concludes with a section about language and country-specific settings (I18N and L10N).

9.1 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.

9.1.1 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.

9.1.2 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 9.1, “Entry in `/etc/crontab`” (page 104), `root` is entered. Package-specific tables, located in `/etc/cron.d`, have the same format. See the `cron` man page (`man cron`).

Example 9.1 *Entry in `/etc/crontab`*

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

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

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

To run the `hourly`, `daily` or other periodic maintenance scripts at custom times, remove the time stamp files regularly using `/etc/crontab` entries (see Example 9.2, “`/etc/crontab: Remove Time Stamp Files`” (page 105), which removes the `hourly` one before every full hour, the `daily` one once a day at 2:14 a.m., etc.).

Example 9.2 */etc/crontab: Remove Time Stamp Files*

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

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

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

9.1.3 Log Files: Package `logrotate`

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

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

Example 9.3 *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.

9.1.4 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.

9.1.5 The ulimit Command

With the `ulimit` (*user limits*) command, it is possible to set limits for the use of system resources and to have these displayed. `ulimit` is especially useful for limiting available memory for applications. With this, an application can be prevented from co-opting too much of the system resources and slowing or even hanging up the operating system.

`ulimit` can be used with various options. To limit memory usage, use the options listed in Table 9.1, “`ulimit`: Setting Resources for the User” (page 107).

Table 9.1 *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 9.4 *ulimit: Settings in ~/.bashrc*

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

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

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

IMPORTANT

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

9.1.6 The `free` Command

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

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

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

9.1.7 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.

9.1.8 Selecting Man Pages Using the `man` Command

With `man man_page` you normally display a man page for instant reading. Now, if a man page with the same name exists in different sections, `man` prompts the user, the page from which section shall be made visible. The user is then expected to type the section as the answer.

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

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

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

The components of Emacs are divided into several packages:

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

9.2 Virtual Consoles

Linux is a multiuser and multitasking system. The advantages of these features can be appreciated even on a stand-alone PC system. In text mode, there are six virtual consoles available. Switch between them using `Alt + F1` through `Alt + F6`. The seventh console

is reserved for X and the tenth console shows kernel messages. More or fewer consoles can be assigned by modifying the file `/etc/inittab`.

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

9.3 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 the documents listed in `/usr/share/doc/packages/xkeyboard-config` (part of the `xkeyboard-config` package).

9.4 Language and Country-Specific Settings

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

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

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

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

`RC_LC_ALL`

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

`RC_LANG`

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

`ROOT_USES_LANG`

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

The variables can be set with the YaST `sysconfig` editor (see Section 7.3.1, “Changing the System Configuration Using the YaST `sysconfig` Editor” (page 78)). The value of such a variable contains the language code, country code, encoding and modifier. The individual components are connected by special characters:

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

9.4.1 Some Examples

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

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

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

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

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

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

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

```
LANG=en_IE@euro
```

The above example explicitly includes the Euro sign in a language setting. This setting is basically obsolete now, as UTF-8 also covers the Euro symbol. It is only useful if an application supports ISO-8859-15 and not UTF-8.

SuSEconfig reads the variables in `/etc/sysconfig/language` and writes the necessary changes to `/etc/SuSEconfig/profile` and `/etc/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 that messages are displayed in Spanish instead.

9.4.2 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
```

9.4.3 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.

9.4.4 For More Information

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

Printer Operation

SUSE® Linux Enterprise Server supports printing with many types of printers, including remote network printers. Printers can be configured manually or with YaST. For configuration instructions, refer to Section “Setting Up a Printer” (Chapter 8, *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 10.7, “Troubleshooting” (page 126).

CUPS is the standard print system in SUSE Linux Enterprise Server. 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 Server 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 an adequate print result. Linux may not be able to address some functions of extremely recent and high-end 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 10.7.1, “Printers without Standard Printer Language Support” (page 127) for more information.

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

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

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

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

The Ghostscript Web page.

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

10.1 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.

10.2 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.

► **zseries:** Printers and similar devices provided by the z/VM that connect locally with the IBM System z mainframes are not supported by CUPS or LPRng. On these platforms, printing is only possible over the network. The cabling for network printers must be installed according to the instructions of the printer manufacturer. ◀

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.

10.3 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 Server, 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 10.6.2, “PPD Files in Various Packages” (page 125) and Section 10.7.2, “No Suitable PPD File Available for a PostScript Printer” (page 127).

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 8, *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 Server 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.

10.4 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 ascertain the protocol. `nmap` checks a host for open ports. For example:

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

10.4.1 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(8)`.

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

1 First, list all options:

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

Example:

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

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

2 Change the option with `lpadmin`:

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

3 Check the new setting:

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

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

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

10.5 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`.

10.6 Special Features in SUSE Linux Enterprise Server

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

10.6.1 CUPS and Firewall

After having performed a default installation of SUSE Linux Enterprise Server, `SuSE-firewall2` 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.

10.6.2 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.

10.7 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.

10.7.1 Printers without Standard Printer Language Support

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

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

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

Instead of spending time trying to make a proprietary Linux driver work, it may be more cost-effective to purchase a 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.

10.7.2 No Suitable PPD File Available for a PostScript Printer

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

printer manufacturer or download a suitable PPD file from the Web page of the printer manufacturer.

If the PPD file is provided as a zip archive (.zip) or a self-extracting zip archive (.exe), unpack it with `unzip`. First, review the license terms of the PPD file. Then use the `cupstestppd` utility to check if the PPD file complies with “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.

10.7.3 Parallel Ports

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

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

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

If interrupt 7 is free, it can be activated with the entry shown in Example 10.1, “`/etc/modprobe.conf`: Interrupt Mode for the First Parallel Port” (page 129). 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 10.1 */etc/modprobe.conf: Interrupt Mode for the First Parallel Port*

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

10.7.4 Network Printer Connections

Identifying Network Problems

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

Checking the TCP/IP Network

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

Checking a Remote lpd

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

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

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

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

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

If `lpd` does not respond, it may not be active or there may be basic network problems. If `lpd` responds, the response should show why printing is not possible on the *queue* on *host*. If you receive a response like that shown in Example 10.2, “Error Message from `lpd`” (page 129), the problem is caused by the remote `lpd`.

Example 10.2 *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 10.3, “Broadcast from the CUPS Network Server” (page 130).

Example 10.3 *Broadcast from the CUPS Network Server*

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

► **zseries:** Take into account that IBM System z ethernet devices do not receive broadcasts by default. ◀

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

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

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

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

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

Troubleshooting a Network Printer or Print Server Box

Spoolers running in a print server box sometimes cause problems when they have to deal with multiple print jobs. Since this is caused by the spooler in the print server box, there no way to resolve this issue. As a work-around, circumvent the spooler in the print server box by addressing the printer connected to the print server box directly with the TCP socket. See Section 10.4, “Network Printers” (page 121).

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

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

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

Enter a command like

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

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

10.7.5 Defective Printouts without Error Message

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

10.7.6 Disabled Queues

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

10.7.7 CUPS Browsing: Deleting Print Jobs

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

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

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

10.7.8 Defective Print Jobs and Data Transfer Errors

If you switch the printer off or shut down the computer during the printing process, print jobs remain in the queue. Printing resumes when the computer (or the printer) is switched back on. Defective print jobs must be removed from the queue with `cancel`.

If a print job is defective or an error occurs in the communication between the host and the printer, the printer prints numerous sheets of paper with unintelligible characters,

because it is unable to process the data correctly. To rectify this situation, follow these steps:

- 1 To stop printing, remove all paper from ink jet printers or open the paper trays of laser printers. High-quality printers have a button for canceling the current 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.

10.7.9 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.

10.7.10 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

11

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

11.1 The `/dev` Directory

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

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

11.2 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 11.6, “Influencing Kernel Device Event Handling with udev Rules” (page 139).

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.

11.3 Drivers, Kernel Modules and Devices

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

```
MODALIAS=usb:v046DpC03Ed2000dc00dsc00dp00ic03isc01ip02
```

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

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

11.4 Booting and Initial Device Setup

All device events happening during the boot process before the `udev` daemon is running are lost, because the infrastructure to handle these events resides on the root file system and is not available at that time. To cover that loss, the kernel provides a `uevent` file located in the device directory of every device in the `sysfs` file system. By writing `add` to that file, the kernel resends the same event as the one lost during boot. A simple loop over all `uevent` files in `/sys` triggers all events again to create the device nodes and perform device setup.

As an example, a USB mouse present during boot may not be initialized by the early boot logic, because the driver is not available at that time. The event for the device discovery was lost and failed to find a kernel module for the device. Instead of manually searching for possibly connected devices, `udev` just requests all device events from the kernel after the root file system is available, so the event for the USB mouse device just runs again. Now it finds the kernel module on the mounted root file system and the USB mouse can be initialized.

From userspace, there is no visible difference between a device coldplug sequence and a device discovery during runtime. In both cases, the same rules are used to match and the same configured programs are run.

11.5 Monitoring the Running `udev` Daemon

The program `udevadm monitor` can be used to visualize the driver core events and the timing of the `udev` event processes.

```
UEVENT[1185238505.276660] add    /devices/pci0000:00/0000:00:1d.2/usb3/3-1
(usb)
UDEV   [1185238505.279198] add    /devices/pci0000:00/0000:00:1d.2/usb3/3-1
(usb)
```

```

UEVENT[1185238505.279527] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0 (usb)
UDEV [1185238505.285573] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0 (usb)
UEVENT[1185238505.298878] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10 (input)
UDEV [1185238505.305026] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10 (input)
UEVENT[1185238505.305442] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/mouse2 (input)
UEVENT[1185238505.306440] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/event4 (input)
UDEV [1185238505.325384] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/event4 (input)
UDEV [1185238505.342257] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/mouse2 (input)

```

The UEVENT lines show the events the kernel has sent over netlink. The UDEV lines show the finished udev event handlers. The timing is printed in microseconds. The time between UEVENT and UDEV is the time udev took to process this event or the udev daemon has delayed its execution to synchronize this event with related and already running events. For example, events for hard disk partitions always wait for the main disk device event to finish, because the partition events may rely on the data that the main disk event has queried from the hardware.

`udevadm monitor --env` shows the complete event environment:

```

ACTION=add
DEVPATH=/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10
SUBSYSTEM=input
SEQNUM=1181
NAME="Logitech USB-PS/2 Optical Mouse"
PHYS="usb-0000:00:1d.2-1/input0"
UNIQ=""
EV=7
KEY=70000 0 0 0 0
REL=103
MODALIAS=input:b0003v046DpC03Ee0110-e0,1,2,k110,111,112,r0,1,8,amlsfw

```

udev also sends messages to syslog. The default syslog priority that controls which messages are sent to syslog is specified in the udev configuration file `/etc/udev/udev.conf`. The log priority of the running daemon can be changed with `udevadm control log_priority=level/number`.

11.6 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 11.1 *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 considering. It consists of two match keys (`KERNEL` and `ATTRS`) and one assign key (`SYMLINK`). The `KERNEL` key searches for all devices of the `ttyUSB` type. Using the `*` wild card, this key matches several of these devices. The second match key, `ATTRS`, checks whether the `product` attribute file in `sysfs` for any `ttyUSB` device contains a certain string. The assign key (`SYMLINK`) triggers the addition of a symbolic link to this device under `/dev/pilot`. The operator used in this key (`+=`) tells `udev` to additionally perform this action, even if previous or later rules add other symbolic links. As this rule contains two match keys, it is only applied if both conditions are met.

The `printer` rule deals with USB printers and contains two match keys which must both apply to get the entire rule applied (`SUBSYSTEM` and `KERNEL`). Three assign keys deal with the naming for this device type (`NAME`), the creation of symbolic device links (`SYMLINK`) and the group membership for this device type (`GROUP`). Using the `*` wild card in the `KERNEL` key makes it match several `lp` printer devices. Substitutions are used in both, the `NAME` and the `SYMLINK` keys to extend these strings by the internal device name. For example, the symlink to the first `lp` USB printer would read `/dev/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.

11.6.1 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.

11.6.2 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.

11.6.3 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.

11.6.4 Using udev Assign Keys

In contrast to the match keys described above, assign keys do not describe conditions that must be met. They assign values, names and actions to the device nodes maintained by udev.

NAME

The name of the device node to be created. Once a rule has set a node name, all other rules with a `NAME` key for this node are ignored.

SYMLINK

The name of a symlink related to the node to be created. Multiple matching rules can add symlinks to be created with the device node. You can also specify multiple symlinks for one node in one rule using the space character to separate the symlink names.

OWNER, GROUP, MODE

The permissions for the new device node. Values specified here overwrite anything that has been compiled in.

ATTR{key}

Specify a value to be written to a sysfs attribute of the event device. If the `==` operator is used, this key is also used to match against the value of a sysfs attribute.

ENV{key}

Tell udev to export a variable to the environment. If the `==` operator is used, this key is also used to match against an environment variable.

RUN

Tell udev to add a program to the list of programs to be executed for this device. Keep in mind to restrict this to very short tasks to avoid blocking further events for this device.

LABEL

Add a label where a GOTO can jump to.

GOTO

Tell udev to skip a number of rules and continue with the one that carries the label referenced by the GOTO key.

IMPORT{*type*}

Load variables into the event environment such as the output of an external program. udev imports variables of several different types. If no type is specified, udev tries to determine the type itself based on the executable bit of the file permissions.

- `program` tells udev to execute an external program and import its output.
- `file` tells udev to import a text file.
- `parent` tells udev to import the stored keys from the parent device.

WAIT_FOR_SYSFS

Tells udev to wait for the specified sysfs file to be created for a certain device. For example, `WAIT_FOR_SYSFS="ioerr_cnt"` informs udev to wait until the `ioerr_cnt` file has been created.

OPTIONS

The `OPTION` key may have several possible values:

- `last_rule` tells udev to ignore all later rules.
- `ignore_device` tells udev to ignore this event completely.
- `ignore_remove` tells udev to ignore all later remove events for the device.
- `all_partitions` tells udev to create device nodes for all available partitions on a block device.

11.7 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
```

11.8 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.

11.9 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 Server.

TIP: IBM System z: Configuring the Graphical User Interface

IBM System z does not have any input or output devices supported by X.Org. Therefore, none of the configuration procedures described in this section apply. More relevant information for IBM System z can be found in Chapter 4, *Installation on IBM System z* (↑*Deployment Guide*).

12.1 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 8, *Setting Up Hardware Components with YaST*, ↑*Deployment Guide*). Alternatively it can be configured manually by editing 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 Server 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 Server's `SaX2` have failed. If your setup involves proprietary binary-only drivers, `X -configure` does not 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 12.1, “Sections in `/etc/X11/xorg.conf`” (page 150).

Table 12.1 *Sections in `/etc/X11/xorg.conf`*

Type	Meaning
Files	The paths used for fonts and the RGB color table.

Type	Meaning
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.
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

Type	Meaning
	information about the available options in the manual page <code>man 4 i810</code> .
Screen	<p>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.</p> <p>Note that some drivers demand that all of the used configurations must be present in the <code>Display</code> section at some place. For example, 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.</p>
ServerLayout	The layout of a single or multihead configuration. This section binds the input devices <code>InputDevice</code> and the display devices <code>Screen</code> .
DRI	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.

12.1.1 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 12.1, “Screen Section of the File `/etc/X11/xorg.conf`” (page 153).

Example 12.1 *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.

12.1.2 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/packages/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.

12.1.3 Monitor and Modes Section

Like the `Device` sections, the `Monitor` and `Modes` sections describe one monitor each. The configuration file `/etc/X11/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.

12.2 Installing and Configuring Fonts

The installation of additional fonts in SUSE Linux Enterprise Server is very easy. Simply copy the fonts to any directory located in the X11 font path (see Section 12.2.1, “X11 Core Fonts” (page 157)). To enable use of the fonts, the installation directory should be a subdirectory of the directories configured in `/etc/fonts/fonts.conf` (see Section 12.2.2, “Xft” (page 158)) or included into this file with `/etc/fonts/suse-font-dirs.conf`.

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

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

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

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

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

12.2.1 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 any 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 acquire `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 Server 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 Server contain at least the glyphs needed for European languages (formerly encoded as `iso-8859-*`).

12.2.2 Xft

From the outset, the programmers of Xft made sure that scalable fonts including antialiasing are well supported. 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 Server, 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
DejaVu Sans:style=Oblique:weight=80
Lucida Sans Typewriter:style=Regular:weight=80
DejaVu Sans:style=Book:weight=80
```

```
DejaVu Sans:style=Bold:weight=200
Lucida Sans:style=Regular:weight=80
```

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

Table 12.2 *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.

12.3 For More Information

Install the packages `xorg-x11-doc` and `howtoenh` to get more in-depth information about 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 `nv` driver, find more information about this driver in `man 4 nv`.

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

13

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.

13.1 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 available as separate packages. FUSE plugins are not shipped with SUSE Linux Enterprise.

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.

13.2 Available FUSE Plug-ins

FUSE is dependent on plug-ins. The following table lists common plug-ins. FUSE plugins are not shipped with SUSE Linux Enterprise.

Table 13.1 *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
<code>wdfs</code>	mount WebDAV file systems

13.3 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.

14.1 Laptops

The hardware of laptops differs from that of a normal desktop system. This is because criteria like exchangeability, space requirements and power consumption must be taken into account. The manufacturers of mobile hardware have developed standard interfaces like PCMCIA (Personal Computer Memory Card International Association), Mini PCI and Mini PCIe that can be used to extend the hardware of laptops. The standards cover memory cards, network interface cards, ISDN (and modem cards) and external hard disks.

TIP: SUSE Linux Enterprise Server and Tablet PCs

SUSE Linux Enterprise Server 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 the same way any other systems are. For a detailed introduction to the installation and configuration of Tablet PCs, refer to Chapter 17, *Using Tablet PCs* (page 201).

14.1.1 Power Conservation

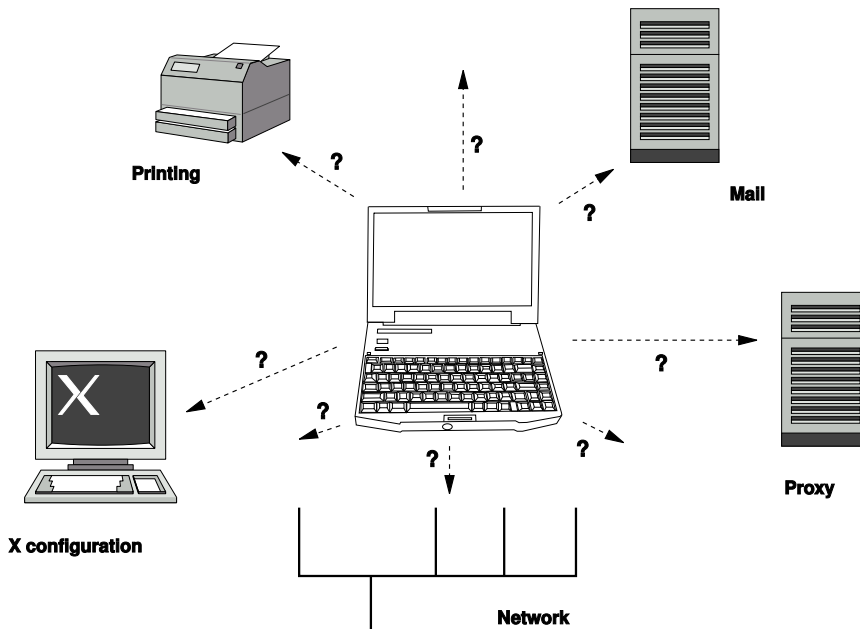
The inclusion of energy-optimized system components during laptop manufacturing contributes to their suitability for use without access to the electrical power grid. Their contribution towards conservation of power is at least as important as that of the operating system. SUSE® Linux Enterprise Server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

14.1.2 Integration in Changing Operating Environments

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

Figure 14.1 *Integrating a Mobile Computer in an Existing Environment*



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

Network

This includes IP address assignment, name resolution, Internet connectivity and connectivity to other networks.

Printing

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

E-Mail and Proxies

As with printing, the list of the corresponding servers must be current.

X (Graphical Environment)

If your laptop is temporarily connected to a projector or an external monitor, different display configurations must be available.

SUSE Linux Enterprise Server offers several ways of integrating laptops into existing operating environments:

NetworkManager

NetworkManager is especially tailored for mobile networking on laptops. It provides a means to easily and automatically switch between network environments and different types of networks, such as wireless LAN and ethernet. NetworkManager supports WEP and WPA/WPA2-PSK encryption in wireless LANs. It also supports dial-up connections (with smpppd), DSL connections, and option to configure virtual private networks (VPN). Both desktop environments (GNOME and KDE) include a front-end for NetworkManager. For more information about the desktop applets, see Section 23.4, “Using KNetworkManager” (page 336) and Section 23.5, “Using GNOME NetworkManager Applet” (page 340).

Table 14.1 *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 networking tools to configure networking whenever NetworkManager cannot handle network configuration correctly.

TIP: DNS configuration and various types of network connections

If you travel frequently with your laptop and change different types of network connections, NetworkManager works fine when all DNS addresses are assigned correctly assigned with DHCP. If some of your connections use static DNS address(es), add the to the `NETCONFIG_DNS_STATIC_SERVERS` in `/etc/sysconfig/network/config`.

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 operating 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 `scpm`. Enable SCPM using the YaST Profile Management module and configure the users that are 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 `scpm` command line tool. Refer to the `scpm` man page for details. Create profiles for all the different setups you want to use this system in. Switching between profiles can either be done on the running system via `scpm switch PROFILENAME` or at system boot time via the F3 key. When switching profiles, SCPM automatically adjusts your system configuration to the new environment described in the profile you have chosen.

SLP

The service location protocol (SLP) simplifies the connection of your 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 also be used to install a system, minimizing the effort of searching for a suitable installation source. Find detailed information about SLP in Chapter 19, *SLP Services in the Network* (page 277).

14.1.3 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 Server provides for each task.

System Monitoring

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

PowerDevil

PowerDevil is an application, that can be opened either from the *Kickoff Application Launcher (Configure Desktop > Advanced > Power Management)* or via the *Battery Monitor* tray icon. The applet icon adjusts to represent the type of power supply. When working on AC power, a small lightening icon is displayed. When working on batteries, the icon changes to a battery.

Click the PowerDevil tray icon to access options to configure its behavior. You can choose one of five displayed power profiles, according to your needs. For example, the *Presentation* scheme disables the screen saver and the power management in general, so that your presentation is not interrupted by system events. Click *More...* to open PowerDevil's configuration screen. Here you can edit individual profiles and set advanced power management options, such as what to do when the laptop lid has been closed, or when the battery is low.

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, memory usage, and other system parameters. 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. 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. One of the most widely used is a command-line tool called `rsync`. For more information, see its manual page (`man 1 rsync`).

Wireless Communication

As well as connecting to a home or office network with a cable, a laptop can also use wireless connection to access other computers, peripherals, cellular phones or PDAs. Linux supports three types of wireless communication:

WLAN

With the largest range of these wireless technologies, WLAN is the only one suitable for the operation of large and sometimes even spatially separate networks. Single machines can connect with each other to form an independent wireless network or access the Internet. Devices called *access points* act as base stations for WLAN-enabled devices and act as intermediaries for access to the Internet. A mobile user can switch among access points depending on location and which access point is offering the best connection. Like in cellular telephony, a large network is available to WLAN users without binding them to a specific location for accessing it. Find details about WLAN in Chapter 15, *Wireless LAN* (page 177).

Bluetooth

Bluetooth has the broadest application spectrum of all wireless technologies. It can be used for communication between computers (laptops) and PDAs or cellular phones, as can IrDA. It can also be used to connect various computers within range. Bluetooth is also used to connect wireless system components, like a keyboard or a mouse. The range of this technology is, however, not sufficient to connect remote systems to a network. WLAN is the technology of choice for communicating through physical obstacles like walls.

IrDA

IrDA is the wireless technology with the shortest range. Both communication parties must be within viewing distance of each other. Obstacles like walls cannot be overcome. One possible application of IrDA is the transmission of a file from a laptop to a cellular phone. The short path from the laptop to the cellular phone is then covered using IrDA. The long range transport of the file to the recipient of the file is handled by the mobile network. Another application of IrDA is the wireless transmission of printing jobs in the office.

14.1.4 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 Server 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 Server 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 15, *Wireless LAN* (page 177).

14.2 Mobile Hardware

SUSE Linux Enterprise Server 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.

External Hard Disks (USB and FireWire)

As soon as an external hard disk is correctly recognized by the system, its icon appears in the file manager. Clicking the icon displays the contents of the drive. It is possible to create folders and files here and edit or delete them. To rename a hard disk from the name it had been given by the system, select the corresponding menu item from the menu that opens when the icon is right-clicked. This name change is limited to display in the file manager. The descriptor by which the device is mounted in `/media` remains unaffected by this.

USB Flash Drives

These devices are handled by the system just like external hard disks. It is similarly possible to rename the entries in the file manager.

14.3 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 173). The configuration of these protocols on the cellular phones themselves is described in their manuals.

The support for synchronizing with handheld devices manufactured by Palm, Inc., is already built into Evolution and Kontact. Initial connection with the device is easily performed with the assistance of a wizard. Once the support for Palm Pilots is configured, it is necessary to determine which type of data should be synchronized (addresses, appointments, etc.).

14.4 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 Server. 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>.

Wireless LAN

Wireless LANs, or Wireless Local Area Network (WLANs), have become an indispensable aspect of mobile computing. Today, most laptops have built-in WLAN cards. This chapter describes how to set up a WLAN card with YaST, encrypt transmissions, and use tips and tricks.

15.1 WLAN Standards

WLAN cards communicate using the 802.11 standard, prepared by the IEEE organization. Originally, this standard provided for a maximum transmission rate of 2 Mbit/s. Meanwhile, several supplements have been added to increase the data rate. These supplements define details such as the modulation, transmission output, and transmission rates (see Table 15.1, “Overview of Various WLAN Standards” (page 177)). Additionally, many companies implement hardware with proprietary or draft features.

Table 15.1 *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

Name	Band (GHz)	Maximum Transmission Rate (Mbit/s)	Note
802.11b	2.4	11	Less common
802.11g	2.4	54	Widespread, backwards-compatible with 11b
802.11n (formerly 802.11n draft)	2.4 and/or 5	300	Common

802.11 Legacy cards are not supported by SUSE® Linux Enterprise Server. 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.

15.2 Operating Modes

In wireless networking, various techniques and configurations are used to ensure fast, high-quality, and secure connections. Different operating types suit different setups. It can be difficult to choose the right authentication method. The available encryption methods have different advantages and pitfalls.

Basically, wireless networks can be classified into three network modes:

Managed

Managed networks have a managing element: the access point. In this mode (also referred to as infrastructure mode), all connections of the WLAN stations in the network run through the access point, which may also serve as a connection to an ethernet.

Ad-hoc

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.

Master

In master mode your network card is used as the access point. It works only if your WLAN card supports this mode. Find out the details of your WLAN card on

<http://linux-wless.passsys.nl>.

15.3 Authentication

Because a wireless network is much easier to intercept and compromise than a wired network, the various standards include authentication and encryption methods. In the original version of the IEEE 802.11 standard, these are described under the term WEP (Wired Equivalent Privacy). However, because WEP has proven to be insecure (see Section 15.7.2, “Security” (page 187)), the WLAN industry (joined under the name *Wi-Fi Alliance*) has defined an extension called WPA, which is supposed to eliminate the weaknesses of WEP. The later IEEE 802.11i standard (also referred to as WPA2, because WPA is based on a draft version of 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:

None (Open)

An open system is a system that does not require authentication. Any station can join the network. Nevertheless, WEP encryption (see Section 15.4, “Encryption” (page 180)) 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, sometimes WPA-Personal (according to IEEE 802.1x)

WPA-PSK (PSK stands for preshared key) works similarly to the Shared Key procedure. All participating stations as well as the access point need the same key. The key is 256 bits in length and is usually entered as a passphrase. This system does not need a complex key management like WPA-EAP and is more suitable for private use. Therefore, WPA-PSK is sometimes referred to as WPA “Home”.

WPA-EAP, sometimes WPA-Enterprise (according to IEEE 802.1x)

Actually, WPA-EAP (Extensible Authentication Protocol) is not an authentication system but a protocol for transporting authentication information. WPA-EAP is used to protect wireless networks in enterprises. In private networks, it is scarcely used. For this reason, WPA-EAP is sometimes referred to as WPA “Enterprise”.

WPA-EAP needs a Radius server to authenticate users. EAP offers three different methods for connecting and authenticating to the server: 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 for both server and client. First, the server presents its certificate to the client where it is evaluated. If the certificate is considered valid, the client in turn presents its certificate to the server. While TLS is secure, it requires a working certification management infrastructure in your network. This infrastructure is rarely found in private networks.

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 the client authentication data is exchanged. They require far less certification management overhead than TLS, if any.

15.4 Encryption

There are various encryption methods to ensure that no unauthorized person can read the data packets that are exchanged in a wireless network or gain access to the network:

WEP (defined in IEEE 802.11)

This standard makes use of the RC4 encryption algorithm, originally with a key length of 40 bits, later also with 104 bits. Often, the length is declared as 64 bits or 128 bits, depending on whether the 24 bits of the initialization vector are included. However, this standard has some weaknesses. Attacks against the keys generated by this system may be successful. Nevertheless, it is better to use WEP than to not encrypt the network at all.

Some vendors have implemented the non-standard “Dynamic WEP”. It works exactly as WEP and shares the same weaknesses, except that the key is periodically changed by a key management service.

TKIP (defined in WPA/IEEE 802.11i)

This key management protocol defined in the WPA standard uses the same encryption algorithm as WEP, but eliminates its weakness. Because a new key is generated for every data packet, attacks against these keys are fruitless. TKIP is used together with WPA-PSK.

CCMP (defined in IEEE 802.11i)

CCMP describes the key management. Usually, it is used in connection with WPA-EAP, but it can also be used with WPA-PSK. The encryption takes place according to AES and is stronger than the RC4 encryption of the WEP standard.

15.5 Configuration with YaST

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 15.4, “Encryption” (page 180) and Section 15.7.2, “Security” (page 187) for information.

A WLAN card is usually detected during the installation. In case you need to configure it later, do the following:

- 1 Start YaST as user `root`.

- 2 Select *Network Devices > Network Settings* in the YaST control center. The Network Settings dialog opens. If your network is currently controlled by NetworkManager, it can not be edited by YaST and you see a warning message. Click *Ok* and the *Global Options* tab appears. Select *Traditional Method with ifup* to enable editing with YaST.
- 3 Switch to the *Overview* tab where all network cards are listed that have been detected by the system. If you need more information about general network configuration, refer to Section 18.4, “Configuring a Network Connection with YaST” (page 233).
- 4 Choose your wireless card from the list and click *Edit* to open the *Network Card Setup* dialog.
- 5 Configure whether to use a dynamic or a static IP address under the tab *Address*. Usually *Dynamic Address* is fine.
- 6 Click *Next* to proceed to the *Wireless Network Card Configuration* dialog.

Figure 15.1 YaST: Configuring the Wireless Network Card

Wireless Network Card Configuration
Here, set the most important settings for wireless networking. [more](#)

Wireless Device Settings

Operating Mode:
Managed

Network Name (ESSID):
[Dropdown] [Scan Network]

Authentication Mode:
WEP - Open

Key Input Type
☒ Passphrase
 ☐ ASCII
 ☐ Hexadecimal

Encryption Key:
[Text Input]

[Expert Settings] [WEP Keys]

[Help] [Abort] [Back] [Next]

- 7 Configure operating mode, network name (ESSID), and authentication mode:
 - 7a Choose the *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*.

7b Select a *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 select an access point, which may not be the one you intended to use. Use *Scan Network* for a list of available wireless networks.

7c Select an *Authentication Mode*.

Select a suitable authentication method for your network: *No Encryption* (not preferable), *WEP-Open*, *WEP-Shared Key*, *WPA-EAP (WPA version 1 or 2)*, or *WPA-PSK (WPA version 1 or 2)*. If you select WPA authentication, a network name (ESSID) must be set. 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. You have the following options for your key input type:

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.

- 7d** If you need detailed configuration of your WLAN connection, use the *Expert Settings* button. Usually there should be no need to change the preconfigured settings. You have the following options:

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 16, *Power Management* (page 191). Using power management may affect the connection quality and increase the network latency.

- 8** Click *Next* and finish with *Ok*.

- 9** If you have chosen WPA-EAP authentication, another configuration step is needed before your station is ready for deployment in the WLAN.
- 9a** 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).
- 9b** Click *Details* to enter the advanced authentication dialog for your WPA-EAP setup.
- 9c** 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.

15.5.1 Establishing an Ad-Hoc Network

In some cases it is useful to connect two computers equipped with a WLAN card. To establish an ad-hoc network with YaST, do the following:

- 1** Perform Step 1 (page 181) to Step 4 (page 182) as described in Section 15.5, “Configuration with YaST” (page 181).
- 2** Choose *Statically assigned IP Address* and enter the following data:
 - *IP Address*: 192.168.1.1. Change this address on the second computer to 192.168.1.2, for example.
 - *Subnet Mask*: /24
 - *Hostname*: Choose any name you like.
- 3** Proceed with *Next*.

- 4 Configure your operating mode, network name (ESSID), and authentication mode:
 - Choose from the *Operating Mode* popup menu the entry *Ad-hoc*.
 - Choose a *Network Name (ESSID)*. This can be any name, but it has to be used on every computer.
 - Choose from *Authentication Mode* the entry *No Encryption*.
- 5 Click *Next* and finish with *Ok*.
- 6 If you do not have `smpppd` installed, YaST asks you to do so.

15.6 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.

15.7 Tips and Tricks for Setting Up a WLAN

These tips can help tweak speed and stability as well as security aspects of your WLAN.

15.7.1 Stability and Speed

The performance and reliability of a wireless network mainly depend on whether the participating stations receive a clear signal from the other stations. Obstructions like walls greatly weaken the signal. The more the signal strength sinks, the more the transmission slows down. During operation, check the signal strength with the `iwconfig` utility on the command line (`Link Quality` field) or with `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 throughout is no more than half this value.

The useful `iwspy` command can displays WLAN statistics.

```
iwspy wlan0
wlan0      Statistics collected:
  00:AA:BB:CC:DD:EE : Quality:0   Signal level:0   Noise level:0
  Link/Cell/AP      : Quality:60/94 Signal level:-50 dBm  Noise level:-140
  dBm (updated)
  Typical/Reference : Quality:26/94 Signal level:-60 dBm  Noise level:-90
  dBm
```

15.7.2 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.

Use strong passwords for your authentication method. For example, the webpage <https://www.grc.com/passwords.htm> generates random 64 character passwords.

15.8 Troubleshooting

If your WLAN card fails to respond, check the following prerequisites:

1. Do you know your device name? Usually it is `wlan0`. Check with the tool `ifconfig`.

2. Have you checked your needed firmware? Refer to `/usr/share/doc/packages/wireless-tools/README.firmware` for more information.
3. Is your ESSID of your router broadcasted and visible (not hidden)?

15.8.1 Check your Status

The command `iwconfig` can give you important information about your wireless connection. For example, the following line displays the ESSID, the wireless mode, frequency, if you signal is encrypted, the link quality, and much more:

```
iwconfig wlan0
wlan0 IEEE 802.11abg ESSID:"guest"
      Mode:Managed   Frequency:5.22GHz   Access Point: 00:11:22:33:44:55
      Bit Rate:54 Mb/s   Tx-Power=13 dBm
      Retry min limit:7   RTS thr:off   Fragment thr:off
      Encryption key:off
      Power Management:off
      Link Quality:62/92   Signal level:-48 dBm   Noise level:-127 dBm
      Rx invalid nwid:0   Rx invalid crypt:0   Rx invalid frag:0
      Tx excessive retries:10   Invalid misc:0   Missed beacon:0
```

You can also get the previous information with the `iwlist` command. For example, the following line displays the current bit rate:

```
iwlist wlan0 rate
wlan0      unknown bit-rate information.
          Current Bit Rate=54 Mb/s
```

If you want an overview how many access points are available, it can also be done with the `iwlist` command. It gives you a list of “cells” which looks like this:

```
iwlist wlan0 scanning
wlan0   Scan completed:
        Cell 01 - Address: 00:11:22:33:44:55
                Channel:40
                Frequency:5.2 GHz (Channel 40)
                Quality=67/70   Signal level=-43 dBm
                Encryption key: off
                ESSID:"Guest"
                Bit Rates: 6 Mb/s; 9 Mb/s; 12 Mb/s; 18 Mb/s;
                           24 Mb/s; 36 Mb/s; 48 Mb/s
                Mode: Master
                Extra:tsf=0000111122223333
                Extra: Last beacon: 179ms ago
                IE: Unknown: ...
```

15.8.2 Multiple Network Devices

Modern laptops usually have a network card and a WLAN card. If you configured both devices with DHCP (automatic address assignment), you may encounter problems with the name resolution and the default gateway. This is evident from the fact that you can ping the router but cannot surf the Internet. The Support Database features an article on this subject at http://en.opensuse.org/SDB:Name_Resolution_Does_Not_Work_with_Several_Concurrent_DHCP_Clients.

15.8.3 Problems with Prism2 Cards

Several drivers are available for devices with Prism2 chips. The various cards work more or less smoothly with the various drivers. With these cards, WPA is only possible with the hostap driver. If such a card does not work properly or not at all or you want to use WPA, read `/usr/share/doc/packages/wireless-tools/README.prism2`.

15.9 For More Information

More information can be found on the following pages:

1. http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html—The Internet pages of Jean Tourrilhes, who developed the *Wireless Tools* for Linux, present a wealth of useful information about wireless networks.
2. tuxmobil.org—Useful hands-on information about mobile computers under Linux.
3. <http://www.linux-on-laptops.com>—More information about Linux on laptops.

Power Management

► **zseries:** The features and hardware described in this chapter do not exist on IBM System z, making this chapter irrelevant for these platforms. ◀

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.

16.1 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.

16.2 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 16.2.3, “Troubleshooting” (page 195) for more information about troubleshooting ACPI problems.

16.2.1 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 Server 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 must be controlled by a special process, as well. The system interface is `/proc/acpi/processor/*/throttling`.

Putting the Processor to Sleep

The operating system puts the processor to sleep whenever there is no activity. In this case, the operating system sends the CPU a `halt` command. There are three 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 run cool or be 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.

16.2.2 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, gtkacpiw) and tools for editing the ACPI tables in the BIOS (package `pmtools`).

16.2.3 Troubleshooting

There are two different types of problems. On one hand, the ACPI code of the kernel may contain bugs that were not detected in time. In this case, a solution will be made available for download. More often, the problems are caused by the BIOS. Sometimes, deviations from the ACPI specification are purposely integrated in the BIOS to circumvent errors in the ACPI implementation of other widespread operating systems. Hardware components that have serious errors in the ACPI implementation are recorded in a blacklist that prevents the Linux kernel from using ACPI for these components.

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

`pci=noacpi`

Do not use ACPI for configuring the PCI devices.

`acpi=ht`

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

`acpi=off`

Disable ACPI.

WARNING: Problems Booting without ACPI

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

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

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

In this case, the faulty DSDT of the BIOS is ignored. The procedure is described in Section 16.4, “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://tldp.org/HOWTO/ACPI-HOWTO/> (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://acpi.sourceforge.net/dsdt/index.php> (DSDT patches by Bruno Ducrot)

16.3 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` command 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.

16.4 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.

16.4.1 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.

16.4.2 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/*`.

16.4.3 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://en.opensuse.org/Pm-utils> and <http://en.opensuse.org/S2ram> to get more detailed information on how to modify the suspend and resume process.

16.5 For More Information

- <http://en.opensuse.org/S2ram>—How to get Suspend to RAM working
- <http://en.opensuse.org/Pm-utils>—How to modify the general suspend framework

Using Tablet PCs

SUSE® Linux Enterprise Server 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 and dragging and dropping objects
- Using gesture recognition in applications of the X Window System
- Drawing with GIMP
- Taking notes or sketching with applications like Jarnal or Xournal or editing larger amounts of text with Dasher

NOTE: Keyboard or Mouse Needed for Installation

During installation of SUSE Linux Enterprise Server, 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.

17.1 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.

17.2 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.

17.3 Using the Virtual Keyboard

To log in to the KDE or GNOME desktop or to unlock the screen, you can either enter your username and password as usual or via the virtual keyboard (`xvkbd`) displayed below the login field. To configure the keyboard or to access the integrated help, click the `xvkbd` field at the left lower corner and open the `xvkbd` main menu.

If your input is not visible (or is not transferred to the window where you need it), redirect the focus by clicking the *Focus* key in `xvkbd` and then clicking into the window that should get the keyboard events.

Figure 17.1 *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	Enter		
xvkbd	Caps Lock	Alt	Meta				Meta	Alt	←	→	↑	↓	0	Ins	.	Del		

If you want to use `xvkbd` after login, start it from the main menu or with `xvkbd` from a shell.

17.4 Rotating Your Display

Use KRandRTray (KDE) or `gnome-display-properties` (GNOME) to rotate or resize your display manually on the fly. Both KRandRTray and `gnome-display-properties` are applets for the RANDR extension of the X server.

Start KRandRTray or `gnome-display-properties` from the main menu, or enter `krandrtray` or `gnome-display-properties` to start the applet from a shell. After you have started the applet, the applet icon is usually added to your system tray. If the `gnome-display-properties` icon does not automatically appear in the system tray, make sure *Show Displays in Panel* is activated in the *Monitor Resolution Settings* dialog.

To rotate your display with KRandRTray, right-click the icon and select *Configure Display*. Select the desired orientation from the configuration dialog.

To rotate your display with `gnome-display-properties`, right-click the icon and select the desired orientation. Your display is immediately tilted to the new direction. The orientation of the graphics tablet changes also, so it can still interpret the movement of the pen correctly.

If you have problems changing the orientation of your desktop, refer to Section 17.7, “Troubleshooting” (page 210) for more information.

17.5 Using Gesture Recognition

SUSE Linux Enterprise Server 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.

17.5.1 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 17.1 *Training CellWriter*

- 1 Start CellWriter from the main menu or with `cellwriter` from the command line. On the first start, CellWriter automatically starts in the training mode. In training mode it shows a set of characters of the currently chosen key map.
- 2 Enter the gesture you would like to use for a character into the respective character's cell. With the first input, the background changes its color to white, whereas the character itself is shown in light 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 sent to another application until you click the *Enter* button, so you can correct or delete characters before you use them as input. Characters that have been recognized with a low degree of confidence will appear highlighted. To correct your input, use the context menu that appears on right-clicking a cell. To delete a character, either use your pen's eraser, or middle-click with the mouse to clear the cell. After finishing your input in CellWriter, define which application should receive the input by clicking into the application's window. Then send the input to the application by clicking *Enter*.

Figure 17.2 *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.

17.5.2 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.

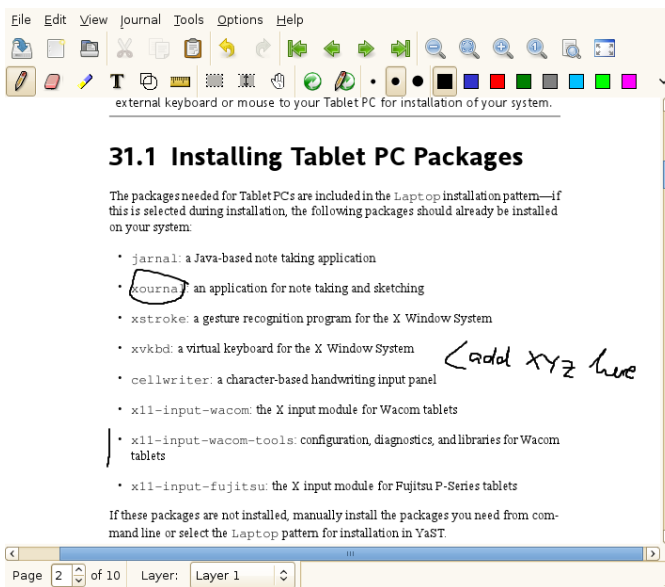
- 1 Start xstroke from the main menu or with `xstroke` from a shell. This adds a pencil icon to your system tray.
- 2 Start the application for which you want to create text input with the pen (for example, a terminal window, a text editor or an OpenOffice.org Writer).
- 3 To activate the gesture recognition 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.

17.6 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 17.3 *Annotating a PDF with Xournal*

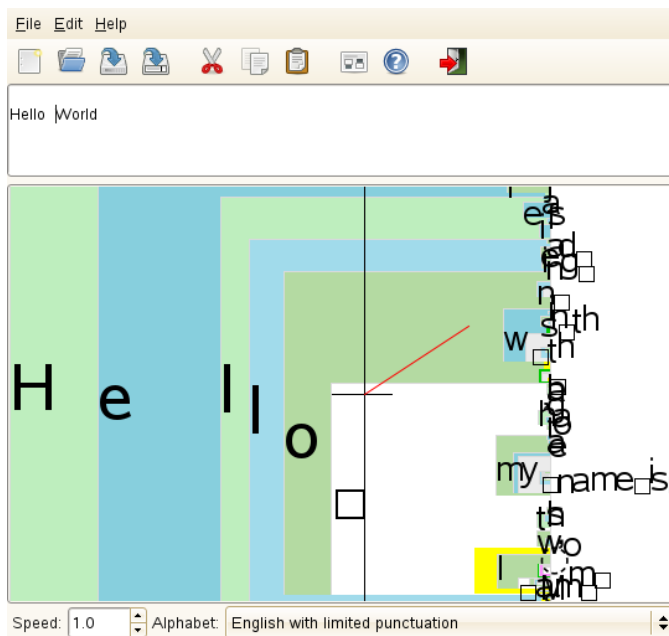


Dasher is another useful application. It was designed for situations where keyboard input is impractical or unavailable. With a bit of training, you can rapidly enter larger amounts

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 17.4 *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/>

17.7 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 17.2, “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.

17.8 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. 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 18.1, “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.”

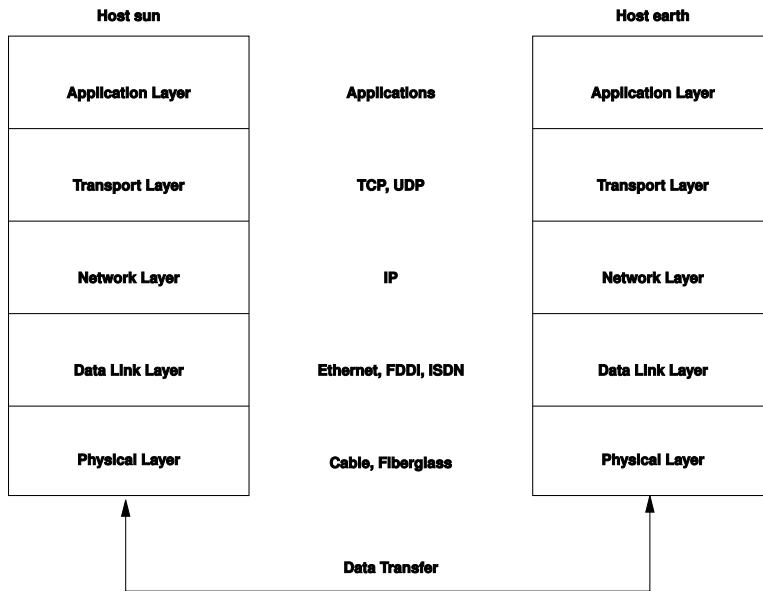
RFC stands for *Request for Comments*. RFCs are documents that describe various Internet protocols and implementation procedures for the operating system and its applications. The RFC documents describe the setup of Internet protocols. To expand your knowledge of any of the protocols, refer to the appropriate RFC documents. These are available at <http://www.ietf.org/rfc.html>.

Table 18.1 *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 or jumbled during the transmission. 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 18.1, “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 18.1 *Simplified Layer Model for TCP/IP*



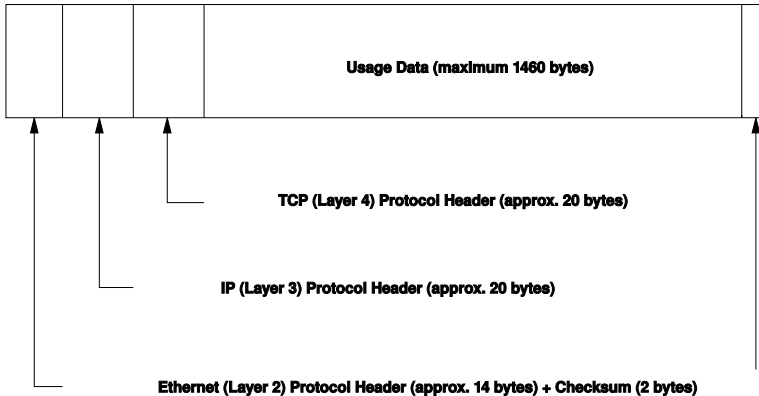
The diagram provides one or two examples for each layer. The layers are ordered according to *abstraction levels*. The lowest layer is very close to the hardware. The uppermost layer, however, is almost a complete abstraction from the hardware. Every layer has its own special function. The special functions of each layer are mostly implicit in their description. The data link and physical layers represent the physical network used, such as ethernet.

Almost all hardware protocols work on a packet-oriented basis. The data to transmit is collected into *packets* (it cannot be sent all at once). The maximum size of a TCP/IP packet is approximately 64 KB. Packets are normally quite smaller, as the network hardware can be a limiting factor. The maximum size of a data packet on an ethernet is about fifteen hundred bytes. The size of a TCP/IP packet is limited to this amount when the data is sent over an ethernet. If more data is transferred, more data packets need to be sent by the operating system.

For the layers to serve their designated functions, additional information regarding each layer must be saved in the data packet. This takes place in the *header* of the packet. Every layer attaches a small block of data, called the protocol header, to the front of each emerging packet. A sample TCP/IP data packet traveling over an ethernet cable is illustrated in Figure 18.2, “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 18.2 *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.

18.1 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 18.2, “IPv6—The Next Generation Internet” (page 221).

18.1.1 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 18.1, “Writing IP Addresses” (page 219).

Example 18.1 *Writing IP Addresses*

```
IP Address (binary): 11000000 10101000 00000000 00010100
IP Address (decimal): 192.      168.      0.      20
```

In decimal form, the four bytes are written in the decimal number system, separated by periods. The IP address is assigned to a host or a network interface. It can be used only once throughout the world. There are exceptions to this rule, but these are not relevant to the following passages.

The points in IP addresses indicate the hierarchical system. Until the 1990s, IP addresses were strictly categorized in classes. However, this system proved too inflexible and was discontinued. Now, *classless routing* (CIDR, classless interdomain routing) is used.

18.1.2 Netmasks and Routing

Netmasks are used to define the address range of a subnetwork. If two hosts are in the same subnetwork, they can reach each other directly. If they are not in the same subnetwork, they need the address of a gateway that handles all the traffic for the subnetwork. To check if two IP addresses are in the same subnet, simply “AND” both addresses with the netmask. If the result is identical, both IP addresses are in the same local network. If there are differences, the remote IP address, and thus the remote interface, can only be reached over a gateway.

To understand how the netmask works, look at Example 18.2, “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 18.2, “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 18.2 *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 18.2 *Specific Addresses*

Address Type	Description
Base Network Address	This is the netmask AND any address in the network, as shown in Example 18.2, “Linking IP Addresses to the Netmask” (page 220) under <i>Result</i> . 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 exam-

Address Type	Description
	ple therefore results in 192.168.0.255. This address cannot be assigned to any hosts.
Local Host	The address 127.0.0.1 is assigned to the “loopback device” on each host. A connection can be set up to your own machine with this address and with all addresses from the complete 127.0.0.0/8 loopback network as defined with IPv4. With IPv6 there is just one loopback address (: : 1).

Because IP addresses must be unique all over the world, you cannot just select random addresses. There are three address domains to use if you want to set up a private IP-based network. These cannot get any connection from the rest of the Internet, because they cannot be transmitted over the Internet. These address domains are specified in RFC 1597 and listed in Table 18.3, “Private IP Address Domains” (page 221).

Table 18.3 *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

18.2 IPv6—The Next Generation Internet

IMPORTANT: IBM System z: IPv6 Support

IPv6 is not supported by the CTC and IUCV network connections of the IBM System z hardware.

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.

18.2.1 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 18.2.2, “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.

Nevertheless if a router is connected to a switch, the router should send periodic advertisements with flags telling the hosts of a network how they should interact with each other. For more information, see RFC 2462 and the `radvd.conf(5)` manpage, and RFC 3315.

Mobility

IPv6 makes it possible to assign several addresses to one network interface at the same time. This allows users to access several networks easily, something that could be compared with the international roaming services offered by mobile phone companies: when you take your mobile phone abroad, the phone automatically logs in to a foreign service as soon as it enters the corresponding area, so you can be reached under the same number everywhere and are able to place an outgoing call just like in your home area.

Secure Communication

With IPv4, network security is an add-on function. IPv6 includes IPsec as one of its core features, allowing systems to communicate over a secure tunnel to avoid eavesdropping by outsiders on the Internet.

Backward Compatibility

Realistically, it would be impossible to switch the entire Internet from IPv4 to IPv6 at one time. Therefore, it is crucial that both protocols are able to coexist not only on the Internet, but also on one system. This is ensured by compatible addresses (IPv4 addresses can easily be translated into IPv6 addresses) and through the use of a number of tunnels. See Section 18.2.3, “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*).

18.2.2 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, ac-

According to the principles of the underlying routing protocol. Anycast addresses are used to make it easier for hosts to find out about servers offering certain services in the given network area. All servers of the same type have the same anycast address. Whenever a host requests a service, it receives a reply from the server with the closest location, as determined by the routing protocol. If this server should fail for some reason, the protocol automatically selects the second closest server, then the third one, and so forth.

An IPv6 address is made up of eight four-digit fields, each representing 16 bits, written in hexadecimal notation. They are separated by colons (:). Any leading zero bytes within a given field may be dropped, but zeros within the field or at its end may not. Another convention is that more than four consecutive zero bytes may be collapsed into a double colon. However, only one such :: is allowed per address. This kind of shorthand notation is shown in Example 18.3, “Sample IPv6 Address” (page 225), where all three lines represent the same address.

Example 18.3 *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 18.4, “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 18.4 *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 18.4, “Various IPv6 Prefixes” (page 226).

Table 18.4 *Various IPv6 Prefixes*

Prefix (hex)	Definition
00	IPv4 addresses and IPv4 over IPv6 compatibility addresses. These are used to maintain compatibility with IPv4. Their use still requires a router able to translate IPv6 packets into IPv4 packets. Several special addresses, such as the one for the loopback device, have this prefix as well.
2 or 3 as the first digit	Aggregatable global unicast addresses. As is the case with IPv4, an interface can be assigned to form part of a certain subnetwork. Currently, there are the following address spaces: 2001::/16 (production quality address space) and 2002::/16 (6to4 address space).
fe80::/10	Link-local addresses. Addresses with this prefix should not be routed and should therefore only be reachable from within the same subnetwork.
fec0::/10	Site-local addresses. These may be routed, but only within the network of the organization to which they belong. In effect, they are the IPv6 equivalent of the current private network address space, such as 10.x.x.x.
ff	These are multicast addresses.

A unicast address consists of three basic components:

Public Topology

The first part (which also contains one of the prefixes mentioned above) is used to route packets through the public Internet. It includes information about the company or institution that provides the Internet access.

Site Topology

The second part contains routing information about the subnetwork to which to deliver the packet.

Interface ID

The third part identifies the interface to which to deliver the packet. This also allows for the MAC to form part of the address. Given that the MAC is a globally unique, fixed identifier coded into the device by the hardware maker, the configuration procedure is substantially simplified. In fact, the first 64 address bits are consolidated to form the `EUI-64` token, with the last 48 bits taken from the MAC, and the remaining 24 bits containing special information about the token type. This also makes it possible to assign an `EUI-64` token to interfaces that do not have a MAC, such as those based on PPP or ISDN.

On top of this basic structure, IPv6 distinguishes between five different types of unicast addresses:

`::` (unspecified)

This address is used by the host as its source address when the interface is initialized for the first time—when the address cannot yet be determined by other means.

`::1` (loopback)

The address of the loopback device.

IPv4 Compatible Addresses

The IPv6 address is formed by the IPv4 address and a prefix consisting of 96 zero bits. This type of compatibility address is used for tunneling (see Section 18.2.3, “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.

18.2.3 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 18.2.2, “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.

18.2.4 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 6, *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. If you want to enable it temporarily until the next reboot, enter `modprobe -i ipv6` as `root`. It is basically impossible to unload the `ipv6` module once loaded.

Because of the autoconfiguration concept of IPv6, the network card is assigned an address in the *link-local* network. Normally, no routing table management takes place on a workstation. The network routers can be queried by the workstation, using the *router advertisement protocol*, for what prefix and gateways should be implemented. The `radvd` program can be used to set up an IPv6 router. This program informs the workstations which prefix to use for the IPv6 addresses and which routers. Alternatively, use `zebra/quagga` for automatic configuration of both addresses and routing.

Consult the `ifcfg-tunnel` (5) man page to get information about how to set up various types of tunnels using the `/etc/sysconfig/network` files.

18.2.5 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.

A book describing all the important aspects of the topic is *IPv6 Essentials* by Silvia Hagen (ISBN 0-596-00125-8).

18.3 Name Resolution

DNS assists in assigning an IP address to one or more names and assigning a name to an IP address. In Linux, this conversion is usually carried out by a special type of software known as *bind*. The machine that takes care of this conversion is called a *name server*. The names make up a hierarchical system in which each name component is separated by a period. The name hierarchy is, however, independent of the IP address hierarchy described above.

Consider a complete name, such as `jupiter.example.com`, written in the format `hostname.domain`. A full name, referred to as a *fully qualified domain name* (FQDN), consists of a hostname and a domain name (`example.com`). The latter also includes the *top level domain* or TLD (`com`).

TLD assignment has become quite confusing for historical reasons. Traditionally, three-letter domain names are used in the USA. In the rest of the world, the two-letter ISO national codes are the standard. In addition to that, longer TLDs were introduced in 2000 that represent certain spheres of activity (for example, `.info`, `.name`, `.museum`).

In the early days of the Internet (before 1990), the file `/etc/hosts` was used to store the names of all the machines represented over the Internet. This quickly proved to be impractical in the face of the rapidly growing number of computers connected to the Internet. For this reason, a decentralized database was developed to store the hostnames in a widely distributed manner. This database, similar to the name server, does not have the data pertaining to all hosts in the Internet readily available, but can dispatch requests to other name servers.

The top of the hierarchy is occupied by *root name servers*. These root name servers manage the top level domains and are run by the Network Information Center (NIC). Each root name server knows about the name servers responsible for a given top level domain. Information about top level domain NICs is available at <http://www.internic.net>.

DNS can do more than just resolve hostnames. The name server also knows which host is receiving e-mails for an entire domain—the *mail exchanger (MX)*.

For your machine to resolve an IP address, it must know about at least one name server and its IP address. Easily specify such a name server with the help of YaST. If you have a modem dial-up connection, you may not need to configure a name server manually at all. The dial-up protocol provides the name server address as the connection is made. The configuration of name server access with SUSE® Linux Enterprise Server is described in Section “Configuring Hostname and DNS” (page 242). Setting up your own name server is described in Chapter 21, *The Domain Name System* (page 289).

The protocol `whois` is closely related to DNS. With this program, quickly find out who is responsible for any given domain.

NOTE: MDNS and .local Domain Names

The `.local` top level domain is treated as link-local domain by the resolver. DNS requests are sent as multicast DNS requests instead of normal DNS requests. If you already use the `.local` domain in your nameserver configuration, you must switch this option off in `/etc/host.conf`. For more information, see the `host.conf` manual page.

If you want to switch off MDNS during installation, use `nomdns=1` as a boot parameter.

For more information on multicast DNS, see <http://www.multicastdns.org>.

18.4 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 18.6, “Configuring a Network Connection Manually” (page 258).

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 Server.

TIP: IBM System z: Hotpluggable Network Cards

On IBM System z platforms, hotpluggable network cards are supported, but not their automatic network integration via DHCP (as is the case on the PC). After detection, manually configure the interface.

18.4.1 Configuring the Network Card with YaST

To configure your wired or wireless network card in YaST, select *Network Devices > Network Settings*. After starting the module, YaST displays the *Network Settings* dialog with four tabs: *Global Options*, *Overview*, *Hostname/DNS* and *Routing*.

The *Global Options* tab allows you to set general networking options such as the use of NetworkManager, IPv6 and general DHCP options. For more information, see Section “Configuring Global Networking Options” (page 234).

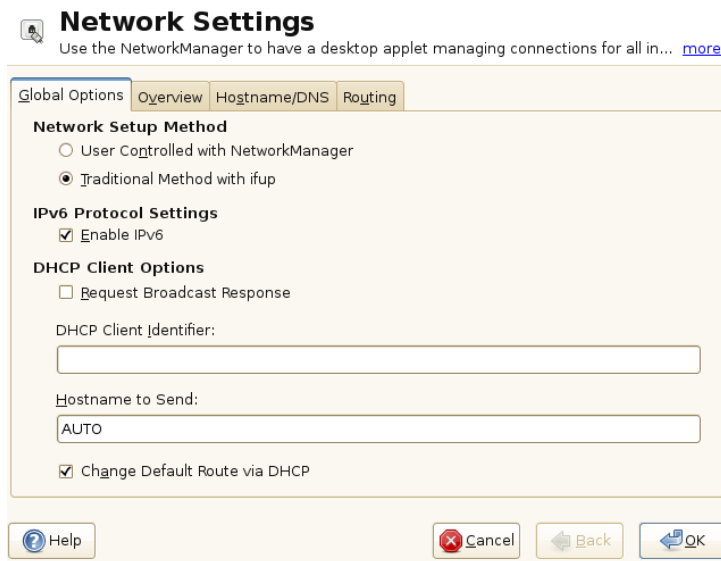
The *Overview* tab contains information about installed network interfaces and configurations. Any properly detected network card is listed with its name. You can manually configure new cards, remove or change their configuration in this dialog. If you want to manually configure a card that was not automatically detected, see Section “Config-

uring an Undetected Network Card” (page 241). If you want to change the configuration of an already configured card, see Section “Changing the Configuration of a Network Card” (page 235).

The *Hostname/DNS* tab allows to set the hostname of the machine and name the servers to be used. For more information, see Section “Configuring Hostname and DNS” (page 242).

The *Routing* tab is used for the configuration of routing. See Section “Configuring Routing” (page 244) for more information.

Figure 18.3 *Configuring Network Settings*



Configuring Global Networking Options

The *Global Options* tab of the YaST *Network Settings* module allows you to set important global networking options, such as the use of NetworkManager, IPv6 and DHCP client options. These settings are applicable for all network interfaces.

In the *Network Setup Method* choose the way network connections are managed. If you want a NetworkManager desktop applet to manage connections for all interfaces, choose *User Controlled with NetworkManager*. This option is well suited for switching between

multiple wired and wireless networks. If you do not run a desktop environment (GNOME or KDE), or if your computer is a Xen server, virtual system, or provides network services such as DHCP or DNS in your network, use the *Traditional Method with ifup*. If NetworkManager is used, `nm-applet` should be used to configure network options and the *Overview*, *Hostname/DNS* and *Routing* tabs of the *Network Settings* module are disabled. For more information on NetworkManager, see Chapter 23, *Using NetworkManager* (page 331).

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 15.5, “Configuration with YaST” (page 181).

Configuring IP Addresses

You can set the IP address of the network card or the way its IP address is determined in the *Address* tab of the *Network Card Setup* dialog. Both IPv4 and IPv6 addresses are supported. The network card can have *No IP Address* (which is useful for bonding devices), a *Statically Assigned IP Address* (IPv4 or IPv6) or a *Dynamic Address* assigned via *DHCP* or *Zeroconf* or both.

If using *Dynamic Address*, select whether to use *DHCP Version 4 Only* (for IPv4), *DHCP Version 6 Only* (for IPv6) or *DHCP Both Version 4 and 6*.

If possible, the first network card with link that is available during the installation is automatically configured to use automatic address setup via DHCP. On SUSE Linux Enterprise Desktop, where NetworkManager is active by default, all network cards are configured.

NOTE: IBM System z and DHCP

On IBM System z platforms, DHCP-based address configuration is only supported with network cards that have a MAC address. This is only the case with OSA and OSA Express cards.

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 242). To configure a gateway, proceed as described in Section “Configuring Routing” (page 244).

Configuring Aliases

One network device can have multiple IP addresses, called aliases.

NOTE: Aliases Are a Compatibility Feature

These so-called aliases resp. labels work with IPv4 only. With IPv6 they will be ignored. Using `iproute2` network interfaces can have one or more addresses.

Using YaST to set an alias for your network card, proceed as follows:

- 1 Select a card from the list of detected cards in the *Overview* tab of the YaST network card configuration module and click *Edit*.
- 2 In the *Address > Additional Addresses* tab, click *Add*.
- 3 Enter *Alias Name*, *IP Address*, and *Netmask*. Do not include the interface name in the alias name.
- 4 Click *OK*.
- 5 Click *Next*.
- 6 To activate the configuration, click *OK*.

Changing the Device Name and Udev Rules

It is possible to change the device name of the network card when it is used. It is also possible to determine whether the network card should be identified by udev via its hardware (MAC) address or via the bus ID. The later option is preferable in large servers to ease hot swapping of cards. To set these options with YaST, proceed as follows:

- 1 Select a card from the list of detected cards in the *Overview* tab of the YaST *Network Settings* module and click *Edit*.
- 2 Go to the *Hardware* tab. The current device name is shown in *Udev Rules*. Click *Change*.
- 3 Select whether udev should identify the card by its *MAC Address* or *Bus ID*. The current MAC address and bus ID of the card are shown in the dialog.
- 4 To change the device name, check the *Change Device Name* option and edit the name.
- 5 Click *OK* and *Next*.
- 6 To activate the configuration, click *OK*.

Changing Network Card Kernel Driver

For some network cards, several kernel drivers may be available. If the card is already configured, YaST allows you to select a kernel driver to be used from a list of available suitable drivers. It is also possible to specify options for the kernel driver. To set these options with YaST, proceed as follows:

- 1 Select a card from the list of detected cards in the *Overview* tab of the YaST *Network Settings* module and click *Edit*.
- 2 Go to the *Hardware* tab.
- 3 Select the kernel driver to be used in *Module Name*. Enter any options for the selected driver in *Options* in the form `option=value` . If more options are used, they should be space-separated.
- 4 Click *OK* and *Next*.

- 5 To activate the configuration, click *OK*.

Activating the Network Device

If you use the traditional method with `ifup`, you can configure your device to either start during boot, on cable connection, on card detection, manually or never. To change device start-up, proceed as follows:

- 1 In YaST select a card from the list of detected cards in *Network Devices > Network Settings* and click *Edit*.
- 2 In the *General* tab, select the desired entry from *Device Activation*.

Choose *At Boot Time* to start the device during the system boot. With *On Cable Connection*, the interface is watched for any existing physical connection. With *On Hotplug*, the interface is set as soon as available. It is similar to the *At Boot Time* option, and only differs in the fact that no error occurs if the interface is not present at boot time. Choose *Manually* to control the interface manually with `ifup`. Choose *Never* to not start the device at all. The *On NFSroot* is similar to *At Boot Time*, but the interface does not shut down with the `rcnetwork stop` command. Use this if you use an `nfs` or `iscsi` root file system.

- 3 Click *Next*.
- 4 To activate the configuration, click *OK*.

Usually, only the system administrator can activate and deactivate network interfaces. If you want any user to be able to activate this interface via KInternet, select *Enable Device Control for Non-root User via Kinternet*.

Setting Up Maximum Transfer Unit Size

You can set a maximum transmission unit (MTU) for the interface. MTU refers to the largest allowed packet size in bytes. A higher MTU brings higher bandwidth efficiency. However, large packets can block up a slow interface for some time, increasing the lag for further packets.

- 1 In YaST select a card from the list of detected cards in *Network Devices > Network Settings* and click *Edit*.

- 2 In the *General* tab, select the desired entry from the *Set MTU* list.
- 3 Click *Next*.
- 4 To activate the configuration, click *OK*.

Configuring the Firewall

Without having to enter the detailed firewall setup as described in Section “Configuring the Firewall with YaST” (Chapter 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.

In *Ethtool Options*, you can set `ethtool` options used by `ifup` for the interface. See the `ethtool` manual page for available options. If the option string starts with a `-` (for example `-K interface_name rx on`), the second word in the string is replaced with the current interface name. Otherwise (for example `autoneg off speed 10`) `ifup` prepends `-s interface_name`.

- 3 Click *Next*.

- 4 Configure any needed options, such as the IP address, device activation or firewall zone for the interface in the *General*, *Address*, and *Hardware* tabs. For more information about the configuration options, see Section “Changing the Configuration of a Network Card” (page 235).
- 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 Chapter 15, *Wireless LAN* (page 177).
- 6 Click *Next*.
- 7 To activate the new network configuration, click *OK*.

Configuring Hostname and DNS

If you did not change the network configuration during installation and the wired card was already available, a hostname was automatically generated for your computer and DHCP was activated. The same applies to the name service information your host needs to integrate into a network environment. If DHCP is used for network address setup, the list of domain name servers is automatically filled with the appropriate data. If a static setup is preferred, set these values manually.

To change the name of your computer and adjust the name server search list, proceed as follows:

- 1 Go to the *Network Settings > Hostname/DNS* tab in the *Network Devices* module in YaST.
- 2 Enter the *Hostname* and, if needed, the *Domain Name*. The domain is especially important if the machine is a mail server. Note that the hostname is global and applies to all set network interfaces.

If you are using DHCP to get an IP address, the hostname of your computer will be automatically set by the DHCP. You may want to disable this behavior if you connect to different networks, because they may assign different hostnames and changing the hostname at runtime may confuse the graphical desktop. To disable using DHCP to get an IP address uncheck *Change Hostname via DHCP*.

Assign Hostname to Loopback IP associates your hostname with 127.0.0.2 (loopback) IP address in `/etc/hosts`. This is an useful option if you want to have the hostname resolvable at all times, even without active network.

- 3 In *Modify DNS Configuration*, select the way the DNS configuration (name servers, search list, the content of the `/etc/resolv.conf` file) is modified.

If the *Use Default Policy* option is selected, the configuration is handled by the `netconfig` script which merges the data defined statically (with YaST or in the configuration files) with data obtained dynamically (from the DHCP client or NetworkManager). This default policy is sufficient in most cases.

If the *Only Manually* option is selected, `netconfig` is not allowed to modify the `/etc/resolv.conf` file. However, this file can be edited manually.

If the *Custom Policy* option is selected, a *Custom Policy Rule* string defining the merge policy should be specified. The string consists of a comma-separated list of interface names to be considered a valid source of settings. Except for complete interface names, basic wildcards to match multiple interfaces are allowed, as well. For example, `eth* ppp?` will first target all `eth` and then all `ppp0-ppp9` interfaces. There are two special policy values that indicate how to apply the static settings defined in the `/etc/sysconfig/network/config` file:

STATIC

The static settings have to be merged together with the dynamic settings.

STATIC_FALLBACK

The static settings are used only when no dynamic configuration is available.

For more information, see the `man 8 netconfig`.

- 4 Enter the *Name Servers* and fill in the *Domain Search* list. Name servers must be specified by IP addresses, such as 192.168.1.116, not by hostnames. Names specified in the *Domain Search* tab are domain names used for resolving hostnames without a specified domain. If more than one *Domain Search* is used, separate domains with commas or white space.
- 5 To activate the configuration, click *OK*.

Configuring Routing

To make your machine communicate with other machines and other networks, routing information must be given to make network traffic take the correct path. If DHCP is used, this information is automatically provided. If a static setup is used, this data must be added manually.

- 1 In YaST go to *Network Settings > Routing*.
- 2 Enter the IP address of the *Default Gateway* (IPv4 and IPv6 if necessary). The default gateway matches every possible destination, but if any other entry exists that matches the required address, use this instead of the default route.
- 3 More entries can be entered in the *Routing Table*. Enter the *Destination* network IP address, *Gateway* IP address and the *Netmask*. Select the *Device* through which the traffic to the defined network will be routed (the minus sign stands for any device). To omit any of these values, use the minus sign `-`. To enter a default gateway into the table, use `default` in the *Destination* field.

NOTE

If more default routes are used, it is possible to specify the metric option to determine which route has a higher priority. To specify the metric option, enter `- metric number` in *Options*. The route with the highest metric is used as default. If the network device is disconnected, its route will be removed and the next one will be used. However, the current kernel does not use metric in static routing, only routing daemons like `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*.

18.4.2 Modem

TIP: IBM System z: Modem

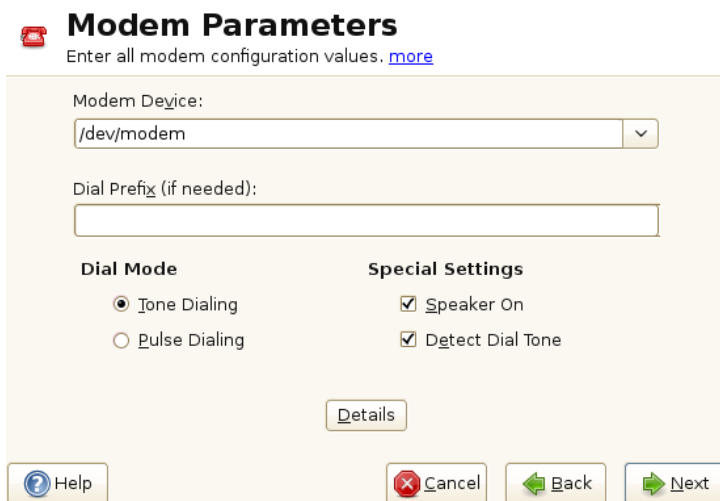
The configuration of this type of hardware is not supported on IBM System z platforms.

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 18.4 *Modem Configuration*



Modem Parameters
Enter all modem configuration values. [more](#)

Modem Device:

Dial Prefix (if needed):

Dial Mode	Special Settings
<input checked="" type="radio"/> Tone Dialing	<input checked="" type="checkbox"/> Speaker On
<input type="radio"/> Pulse Dialing	<input checked="" type="checkbox"/> Detect Dial Tone

If you are behind a private branch exchange (PBX), you may need to enter a dial prefix. This is often a zero. Consult the instructions that came with the PBX to find out. Also select whether to use tone or pulse dialing, whether the speaker should be on and whether

the modem should wait until it detects a dial tone. The last option should not be enabled if the modem is connected to an exchange.

Under *Details*, set the baud rate and the modem initialization strings. Only change these settings if your modem was not detected automatically or if it requires special settings for data transmission to work. This is mainly the case with ISDN terminal adapters. Leave this dialog by clicking *OK*. To delegate control over the modem to the normal user without root permissions, activate *Enable Device Control for Non-root User via Kinternet*. In this way, a user without administrator permissions can activate or deactivate an interface. Under *Dial Prefix Regular Expression*, specify a regular expression. The *Dial Prefix* in KInternet, which can be modified by the normal user, must match this regular expression. If this field is left empty, the user cannot set a different *Dial Prefix* without administrator permissions.

In the next dialog, select the ISP. To choose from a predefined list of ISPs operating in your country, select *Country*. Alternatively, click *New* to open a dialog in which to provide the data for your ISP. This includes a name for the dial-up connection and ISP as well as the login and password provided by your ISP. Enable *Always Ask for Password* to be prompted for the password each time you connect.

In the last dialog, specify additional connection options:

Dial on Demand

If you enable *Dial on Demand*, set at least one name server. Use this feature only if your Internet connection is inexpensive, because there are programs that periodically request data from the Internet.

Modify DNS when Connected

This option is enabled by default, with the effect that the name server address is updated each time you connect to the Internet.

Automatically Retrieve DNS

If the provider does not transmit its domain name server after connecting, disable this option and enter the DNS data manually.

Automatically Reconnect

If this options is enabled, the connection is automatically reestablished after failure.

Ignore Prompts

This option disables the detection of any prompts from the dial-up server. If the connection build-up is slow or does not work at all, try this option.

External Firewall Interface

Selecting this option activates the firewall and sets the interface as external. This way, you are protected from outside attacks for the duration of your Internet connection.

Idle Time-Out (seconds)

With this option, specify a period of network inactivity after which the modem disconnects automatically.

IP Details

This opens the address configuration dialog. If your ISP does not assign a dynamic IP address to your host, disable *Dynamic IP Address* then enter your host's local IP address and the remote IP address. Ask your ISP for this information. Leave *Default Route* enabled and close the dialog by selecting *OK*.

Selecting *Next* returns to the original dialog, which displays a summary of the modem configuration. Close this dialog with *OK*.


18.4.3 ISDN

TIP: IBM System z: ISDN

The configuration of this type of hardware is not supported on IBM System z platforms.

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 18.5 *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:

 Help  Cancel  Back  OK

In the next dialog, shown in Figure 18.5, “ISDN Configuration” (page 248), 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 18.6 ISDN Interface Configuration

Add SyncPPP Interface ippp0
My phone number -- As your own telephone number (MSN), put in your telephone numb... [more](#)

Connection Settings

My Phone Number:

Activate device:
 ☒ Enable Device Control for Non-root User Via

☒ ChargeUP

☐ Channel Bundling

☒ External Firewall Interface ☒ Restart Firewall

The number to enter for *My Phone Number* depends on your particular setup:

ISDN Card Directly Connected to Phone Outlet

A standard ISDN line provides three phone numbers (called multiple subscriber numbers, or MSNs). If the subscriber asked for more, there may be up to 10. One of these MSNs must be entered here, but without your area code. If you enter the wrong number, your phone operator automatically falls back to the first MSN assigned to your ISDN line.

ISDN Card Connected to a Private Branch Exchange

Again, the configuration may vary depending on the equipment installed:

1. Smaller private branch exchanges (PBX) built for home purposes mostly use the Euro-ISDN (EDSS1) protocol for internal calls. These exchanges have an internal S0 bus and use internal numbers for the equipment connected to them.

Use one of the internal numbers as your MSN. You should be able to use at least one of the exchange's MSNs that have been enabled for direct outward dialing. If this does not work, try a single zero. For further information, consult the documentation delivered with your phone exchange.

2. Larger phone exchanges designed for businesses normally use the 1TR6 protocol for internal calls. Their MSN is called EAZ and usually corresponds to the direct-

dial number. For the configuration under Linux, it should be sufficient to enter the last digit of the EAZ. As a last resort, try each of the digits from 1 to 9.

For the connection to be terminated just before the next charge unit is due, enable *ChargeHUP*. However, remember that may not work with every ISP. You can also enable channel bundling (multilink PPP) by selecting the corresponding option. Finally, you can enable the firewall for your link by selecting *External Firewall Interface* and *Restart Firewall*. To enable the normal user without administrator permissions to activate or deactivate the interface, select the *Enable Device Control for Non-root User via KInternet*.

Details opens a dialog in which to implement more complex connection schemes which are not relevant for normal home users. Leave the *Details* dialog by selecting *OK*.

In the next dialog, configure IP address settings. If you have not been given a static IP by your provider, select *Dynamic IP Address*. Otherwise, use the fields provided to enter your host's local IP address and the remote IP address according to the specifications of your ISP. If the interface should be the default route to the Internet, select *Default Route*. Each host can only have one interface configured as the default route. Leave this dialog by selecting *Next*.

The following dialog allows you to set your country and select an ISP. The ISPs included in the list are call-by-call providers only. If your ISP is not in the list, select *New*. This opens the *Provider Parameters* dialog in which to enter all the details for your ISP. When entering the phone number, do not include any blanks or commas among the digits. Finally, enter your login and the password as provided by the ISP. When finished, select *Next*.

To use *Dial on Demand* on a stand-alone workstation, specify the name server (DNS server) as well. Most ISPs support dynamic DNS, which means the IP address of a name server is sent by the ISP each time you connect. For a single workstation, however, you still need to provide a placeholder address like 192.168.22.99. If your ISP does not support dynamic DNS, specify the name server IP addresses of the ISP. If desired, specify a time-out for the connection—the period of network inactivity (in seconds) after which the connection should be automatically terminated. Confirm your settings with *Next*. YaST displays a summary of the configured interfaces. To activate these settings, select *OK*.

18.4.4 Cable Modem

TIP: IBM System z: Cable Modem

The configuration of this type of hardware is not supported on IBM System z platforms.

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.

18.4.5 DSL

TIP: IBM System z: DSL

The configuration of this type of hardware is not supported on IBM System z platforms.

To configure your DSL device, select the *DSL* module from the YaST *Network Devices* section. This YaST module consists of several dialogs in which to set the parameters of DSL links based on one of the following protocols:

- PPP over Ethernet (PPPoE)
- PPP over ATM (PPPoATM)
- CAPI for ADSL (Fritz Cards)

- Point-to-Point Tunneling Protocol (PPTP)—Austria

In the *DSL Devices* tab of the *DSL Configuration Overview* dialog, you will find a list of installed DSL devices. To change the configuration of a DSL device, select it in the list and click *Edit*. If you click *Add*, you can manually configure a new DSL device.

The configuration of a DSL connection based on PPPoE or PPTP requires that the corresponding network card be set up in the correct way. If you have not done so yet, first configure the card by selecting *Configure Network Cards* (see Section 18.4.1, “Configuring the Network Card with YaST” (page 233)). In the case of a DSL link, addresses may be assigned automatically but not via DHCP, which is why you should not enable the option *Dynamic Address*. Instead, enter a static dummy address for the interface, such as 192.168.22.1. In *Subnet Mask*, enter 255.255.255.0. If you are configuring a stand-alone workstation, leave *Default Gateway* empty.

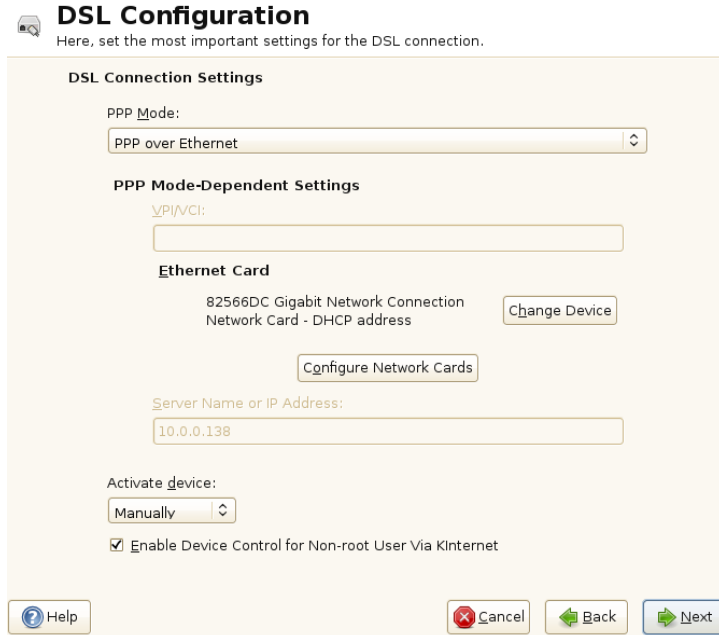
TIP

Values in *IP Address* and *Subnet Mask* are only placeholders. They are only needed to initialize the network card and do not represent the DSL link as such.

In the first DSL configuration dialog (see Figure 18.7, “DSL Configuration” (page 253)), 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 18.7 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:

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.

18.4.6 IBM System z: Configuring Network Devices

SUSE Linux Enterprise Server for IBM System z supports several different types of network interfaces. YaST can be used to configure all of them.

The qeth-hsi Device

To add a `qeth-hsi` (Hipersockets) interface to the installed system, start the *Network Devices > Network Settings* module in YaST. Select one of the devices marked *Hiper-socket* to use as the READ device address and click *Edit*. Enter a device number for the read channel, write channel and control channel (example device number format: 0.0.0600). Then click next. In the *Network Address Setup* dialog, specify the IP address and netmask for the new interface and leave the network configuration by pressing *Next* and *OK*.

The qeth-ethernet Device

To add a `qeth-ethernet` (IBM OSA Express Ethernet Card) interface to the installed system, start the *Network Devices > Network Settings* module in YaST. Select one of the devices marked *IBM OSA Express Ethernet Card* to use as the READ. Enter a device number for the read channel, write channel and control channel (example device number format: 0.0.0600).device address and click *Edit*. Enter the needed port name, port number (if applicable) and some additional options (see the *Linux for IBM System z: Device Drivers, Features, and Commands* manual for reference, http://www.ibm.com/developerworks/linux/linux390/documentation_novell_suse.html), your IP address, and an appropriate netmask. Leave the network configuration with *Next* and *OK*.

The ctc Device

To add a `ctc` (IBM parallel CTC Adapter) interface to the installed system, start the *Network Devices > Network Settings* module in YaST. Select one of the devices marked *IBM Parallel CTC Adapter* to use as your read channel and click *Configure*. Choose the *Device Settings* that fit your devices (usually this would be *Compatibility Mode*). Specify both your IP address and the IP address of the remote partner. If needed, adjust

the MTU size with *Advanced > Detailed Settings*. Leave the network configuration with *Next* and *OK*.

WARNING

The use of this interface is deprecated. This interface will not be supported in future versions of SUSE Linux Enterprise Server.

The lcs Device

To add an `lcs` (IBM OSA-2 Adapter) interface to the installed system, start the *Network Devices > Network Settings* module in YaST. Select one of the devices marked *IBM OSA-2 Adapter* and click *Configure*. Enter the needed port number, some additional options (see the *Linux for IBM System z: Device Drivers, Features, and Commands* manual for reference, http://www.ibm.com/developerworks/linux/linux390/documentation_novell_suse.html), your IP address and an appropriate netmask. Leave the network configuration with *Next* and *OK*.

The IUCV Device

To add an `iucv` (IUCV) interface to the installed system, start the *Network Devices > Network Settings* module in YaST. Select a device marked *IUCV* and click *Edit*. YaST prompts you for the name of your IUCV partner (*Peer*). Enter the name (this entry is case-sensitive) and select *Next*. Specify both the *IP Address* and the *Remote IP Address* of your partner. If needed, *Set MTU* size on *General* tab. Leave the network configuration with *Next* and *OK*.

WARNING

The use of this interface is deprecated. This interface will not be supported in future versions of SUSE Linux Enterprise Server.

18.5 NetworkManager

NetworkManager is the ideal solution for laptops and other portable computers. With NetworkManager, you do not need to worry about configuring network interfaces and switching between networks when you are moving.

18.5.1 NetworkManager and ifup

However, NetworkManager is not a suitable solution for all cases, so you can still choose between the traditional method for managing network connections (ifup) and NetworkManager. If you want to manage your network connection with NetworkManager, enable NetworkManager in the YaST Network Settings module as described in Section 23.2, “Enabling NetworkManager” (page 332) 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 331).

Some differences between ifup and NetworkManager include:

`root` Privileges

If you use NetworkManager for network setup, you can easily switch, stop or start your network connection at any time from within your desktop environment using an applet. NetworkManager also makes it possible to change and configure wireless card connections without requiring `root` privileges. For this reason, NetworkManager is the ideal solution for a mobile workstation.

Traditional configuration with ifup also provides some ways to switch, stop or start the connection with or without user intervention, like user-managed devices.

However, this always requires `root` privileges to change or configure a network device. This is often a problem for mobile computing, where it is not possible to preconfigure all the connection possibilities.

Types of Network Connections

Both traditional configuration and NetworkManager can handle network connections with a wireless network (with WEP, WPA-PSK, and WPA-Enterprise access), dial-up and wired networks using DHCP and static configuration. They also support connection through VPN.

NetworkManager tries to keep your computer connected at all times using the best connection available. If the network cable is accidentally disconnected, it tries to reconnect. It can find the network with the best signal strength from the list of your wireless connections and automatically use it to connect. To get the same functionality with `ifup`, a great deal of configuration effort is required.

18.5.2 NetworkManager Functionality and Configuration Files

The individual network connection settings created with NetworkManager are stored in configuration profiles. The *system* connections configured with either NetworkManager or YaST can be found in `/etc/sysconfig/network/ifcfg-*`. Any user-defined connections are stored in GConf for GNOME or `$HOME/.kde4/share/apps/networkmanagement/*` for KDE.

In case no profile is configured, NetworkManager automatically creates one and names it `Auto_${INTERFACE-NAME}`. That is made in an attempt to work without any configuration for as many cases as (securely) possible. If the automatically created profiles do not suit your needs, use the network connection configuration dialogs provided by KDE or GNOME to modify them as desired. For more information, refer to Section 23.3, “Configuring Network Connections” (page 333).

18.5.3 Controlling and Locking Down NetworkManager Features

On centrally administered machines, certain NetworkManager features can be controlled or disabled with PolicyKit, for example if a user is allowed to modify administrator defined connections or if a user is allowed to define his own network configurations. To view or change the respective NetworkManager policies, start the graphical *Authorizations* tool for PolicyKit. In the tree on the left side, find them below the *network-manager-settings* entry. For an introduction to PolicyKit and details on how to use it, refer to Chapter 9, *PolicyKit* (↑*Security Guide*).

18.6 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 (`/lib/udev/rules.d/75-persistent-net-generator.rules`) generates a rule matching the hardware (using its hardware address by default) and assigns a persistently unique interface for the hardware. The udev database of network interfaces is stored in the file `/etc/udev/rules.d/70-persistent-net.rules`. Every line in the file describes one network interface and specifies its persistent name. System administrators can change the assigned names by editing the `NAME=""` entries. The persistent rules can also be modified using YaST.

Table 18.5, “Manual Network Configuration Scripts” (page 258) summarizes the most important scripts involved in the network configuration.

Table 18.5 *Manual Network Configuration Scripts*

Command	Function
<code>ifup</code> , <code>ifdown</code> , <code>ifstatus</code>	The <code>if</code> scripts start or stop network interfaces, or return the status of the specified interface. For more information, see the <code>ifup</code> manual page.
<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</code>

Command	Function
	<p><code>stop</code> to stop, <code>rcnetwork start</code> to start and <code>rcnetwork restart</code> to restart network interfaces. If you want to stop, start or restart just one interface, use the command followed by the interface name, for example <code>rcnetwork restart eth0</code>. The <code>rcnetwork status</code> command displays the state of the interfaces, their IP addresses and whether a DHCP client is running. With <code>rcnetwork stop-all-dhcp-clients</code> and <code>rcnetwork restart-all-dhcp-clients</code> you can stop or restart DHCP clients running on network interfaces.</p>

For more information about udev and persistent device names, see Chapter 11, *Dynamic Kernel Device Management with udev* (page 135).

18.6.1 Configuration Files

This section provides an overview of the network configuration files and explains their purpose and the format used.

/etc/sysconfig/network/ifcfg-*

These files contain the configurations for network interfaces. They include information such as the start mode and the IP address. Possible parameters are described in the manual page of `ifup`. Additionally, most variables from the files `dhcp`, `wireless` and `config` can be used in the `ifcfg-*` files if a general setting should be used for only one interface.

For `ifcfg.template`, see Section “`/etc/sysconfig/network/config`, `/etc/sysconfig/network/dhcp`, and `/etc/sysconfig/network/wireless`” (page 260).

► **zseries:** IBM System z do not support USB. The names of the interface files and network aliases contain System z-specific elements like `qeth`. ◀

/etc/sysconfig/network/config, /etc/sysconfig/network/dhcp, and /etc/sysconfig/network/wireless

The file `config` contains general settings for the behavior of `ifup`, `ifdown` and `ifstatus`. `dhcp` contains settings for DHCP and `wireless` for wireless LAN cards. The variables in all three configuration files are commented. Some of the variables from `/etc/sysconfig/network/config` can also be used in `ifcfg-*` files, where they are given a higher priority. The `/etc/sysconfig/network/ifcfg.template` file lists variables that can be specified in a per interface scope. However, most of the `/etc/sysconfig/network/config` variables are global and cannot be overridden in `ifcfg`-files. For example `NETWORKMANAGER` or `NETCONFIG_*` variables are global.

/etc/sysconfig/network/routes and /etc/sysconfig/network/ifroute-*

The static routing of TCP/IP packets is determined here. All the static routes required by the various system tasks can be entered in the `/etc/sysconfig/network/routes` file: routes to a host, routes to a host via a gateway and routes to a network. For each interface that needs individual routing, define an additional configuration file: `/etc/sysconfig/network/ifroute-*`. Replace `*` with the name of the interface. The entries in the routing configuration files look like this:

# Destination	Dummy/Gateway	Netmask	Device
#			
127.0.0.0	0.0.0.0	255.255.255.0	lo
204.127.235.0	0.0.0.0	255.255.255.0	eth0
default	204.127.235.41	0.0.0.0	eth0
207.68.156.51	207.68.145.45	255.255.255.255	eth1
192.168.0.0	207.68.156.51	255.255.0.0	eth1

The route's destination is in the first column. This column may contain the IP address of a network or host or, in the case of *reachable* name servers, the fully qualified network or hostname.

The second column contains the default gateway or a gateway through which a host or network can be accessed. The third column contains the netmask for networks or hosts

behind a gateway. For example, the mask is 255.255.255.255 for a host behind a gateway.

The fourth column is only relevant for networks connected to the local host such as loopback, Ethernet, ISDN, PPP and dummy device. The device name must be entered here.

An (optional) fifth column can be used to specify the type of a route. Columns that are not needed should contain a minus sign – to ensure that the parser correctly interprets the command. For details, refer to the `routes(5)` man page.

/etc/resolv.conf

The domain to which the host belongs is specified in this file (keyword `search`). Also listed is the status of the name server address to access (keyword `nameserver`). Multiple domain names can be specified in the file. When resolving a name that is not fully qualified, an attempt is made to generate one by attaching the individual `search` entries. Multiple name servers can be specified in multiple lines, each beginning with `nameserver`. Comments are preceded by `#` signs. Example 18.5, “`/etc/resolv.conf`” (page 261) 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 18.5 */etc/resolv.conf*

```
# Our domain
search example.com
#
# We use dns.example.com (192.168.1.116) as nameserver
nameserver 192.168.1.116
```

/sbin/netconfig

`netconfig` is a modular tool to manage additional network configuration settings. It merges statically defined settings with settings provided by autoconfiguration mechanisms as `dhcp` or `ppp` according to a predefined policy. The required changes are applied to the system by calling the `netconfig` modules that are responsible for modifying a configuration file and restarting a service or a similar action.

`netconfig` recognizes three main actions. The `netconfig modify` and `netconfig remove` commands are used by daemons such as `dhcp` or `ppp` to provide or remove settings to `netconfig`. Only the `netconfig update` command is available for the user:

`modify`

The `netconfig modify` command modifies the current interface and service specific dynamic settings and updates the network configuration. `Netconfig` reads settings from standard input or from a file specified with the `--lease-file filename` option and internally stores them until a system reboot (or the next `modify` or `remove` action). Already existing settings for the same interface and service combination are overwritten. The interface is specified by the `-i interface_name` parameter. The service is specified by the `-s service_name` parameter.

`remove`

The `netconfig remove` command removes the dynamic settings provided by a modificatory action for the specified interface and service combination and updates the network configuration. The interface is specified by the `-i interface_name` parameter. The service is specified by the `-s service_name` parameter.

`update`

The `netconfig update` command updates the network configuration using current settings. This is useful when the policy or the static configuration has changed. Use the `-m module_type` parameter, if you want to update a specified service only (`dns`, `nis`, or `ntp`).

The `netconfig` policy and the static configuration settings are defined either manually or using YaST in the `/etc/sysconfig/network/config` file. The dynamic configuration settings provided by autoconfiguration tools as `dhcp` or `ppp` are delivered

directly by these tools with the `netconfig modify` and `netconfig remove` actions. NetworkManager also uses `netconfig modify` and `netconfig remove` actions. When NetworkManager is enabled, `netconfig` (in policy mode `auto`) uses only NetworkManager settings, ignoring settings from any other interfaces configured using the traditional `ifup` method. If NetworkManager does not provide any setting, static settings are used as a fallback. A mixed usage of NetworkManager and the traditional `ifup` method is not supported.

For more information about `netconfig`, see `man 8 netconfig`.

/etc/hosts

In this file, shown in Example 18.6, “`/etc/hosts`” (page 263), 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 18.6 */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 18.7, “`/etc/networks`” (page 263).

Example 18.7 */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 18.6, “Parameters for `/etc/host.conf`” (page 264) shows the parameters available. A sample `/etc/host.conf` is shown in Example 18.8, “`/etc/host.conf`” (page 264).

Table 18.6 *Parameters for `/etc/host.conf`*

<code>order hosts, bind</code>	Specifies in which order the services are accessed for the name resolution. Available arguments are (separated by blank spaces or commas): <i>hosts</i> : searches the <code>/etc/hosts</code> file <i>bind</i> : accesses a name server <i>nis</i> : 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 only if names from the local domain are in the <code>/etc/hosts</code> file, but should still be recognized with the attached domain names.

Example 18.8 *`/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 18.9, “`/etc/nsswitch.conf`” (page 265). Comments are preceded by `#` signs. In this example, the entry under the `hosts` database means that a request is sent to `/etc/hosts` (files) via DNS (see Chapter 21, *The Domain Name System* (page 289)).

Example 18.9 `/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 18.7, “Databases Available via `/etc/nsswitch.conf`” (page 265). In addition, `automount`, `bootparams`, `netmasks` and `publickey` are expected in the near future. The configuration options for NSS databases are listed in Table 18.8, “Configuration Options for NSS “Databases”” (page 266).

Table 18.7 *Databases Available via `/etc/nsswitch.conf`*

aliases	Mail aliases implemented by <code>sendmail</code> ; see <code>man 5 aliases</code> .
ethers	Ethernet addresses.
group	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 18.8 *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> (↑ <i>Security Guide</i>)
<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 mechanism in `nscd` to cache `hosts` makes the local system unable to trust forward and reverse lookup checks. Instead of asking `nscd` to cache names, set up a caching DNS server.

If the caching for `passwd` is activated, it usually takes about fifteen seconds until a newly added local user is recognized. Reduce this waiting time by restarting `nscd` with the command `rcnscd 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 must contain only one line (in which the hostname is set).

18.6.2 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.

`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.

mroute

This object represents a multicast routing cache entry.

tunnel

This object represents a tunnel over IP.

If no command is given, the default command is used (usually `list`).

Change the state of a device with the command `ip link`

`set device_name command`. For example, to deactivate device `eth0`, enter `ip link set eth0 down`. To activate it again, use `ip link set eth0 up`.

After activating a device, you can configure it. To set the IP address, use `ip addr add ip_address + dev device_name`. For example, to set the address of the interface `eth0` to `192.168.12.154/30` with standard broadcast (option `brd`), enter `ip addr add 192.168.12.154/30 brd + dev eth0`.

To have a working connection, you must also configure the default gateway. To set a gateway for your system, enter `ip route add gateway_ip_address`. To translate one IP address to another, use `nat:ip route add nat_ip_address via other_ip_address`.

To display all devices, use `ip link ls`. To display the running interfaces only, use `ip link ls up`. To print interface statistics for a device, enter `ip -s link ls device_name`. To view addresses of your devices, enter `ip addr`. In the output of the `ip addr`, also find information about MAC addresses of your devices. To show all routes, use `ip route show`.

For more information about using `ip`, enter `ip help` or see the `ip(8)` man page. The `help` option is also available for all `ip` 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 18.10, “Output of the Command `ping`” (page 270), 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 18.10 *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 18.11, “Output of the `ifconfig` Command” (page 271), `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 18.11 *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 18.12 *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.

18.6.3 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 18.9, “Some Start-Up Scripts for Network Programs” (page 272).

Table 18.9 *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 <code>xinetd</code> . <code>xinetd</code> can be used to make server services available on the system. For example, it can start <code>vsftpd</code> whenever an FTP connection is initiated.
<code>/etc/init.d/rpcbind</code>	Starts the <code>rpcbind</code> utility that converts RPC program numbers to universal addresses. It is needed for RPC services, such as an NFS server.
<code>/etc/init.d/nfsserver</code>	Starts the NFS server.
<code>/etc/init.d/postfix</code>	Controls the postfix process.
<code>/etc/init.d/ypserv</code>	Starts the NIS server.

18.7 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 desktop applet or a command-line interface. If the Internet gateway is not the host you are using, you might want to control the dial-up connection by way of a network host.

This is where `smpppd` 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.

18.7.1 Configuring smpppd

The connections provided by `smpppd` are automatically configured by YaST. The actual dial-up programs `KInternet` and `cinternet` are also preconfigured. Manual settings are only required to configure additional features of `smpppd` such as remote control.

The configuration file of `smpppd` is `/etc/smpppd.conf`. By default, it does not enable remote control. The most important options of this configuration file are:

`open-inet-socket = yes/no`

To control `smpppd` via the network, set this option to `yes`. `smpppd` listens on port 3185. If this parameter is set to `yes`, the parameters `bind-address`, `host-range` and `password` must be set accordingly.

`bind-address = ip address`

If a host has several IP addresses, use this parameter to determine at which IP address `smpppd` should accept connections. The default is to listen at all addresses.

`host-range = min ipmax ip`

The parameter `host-range` defines a network range. Hosts whose IP addresses are within this range are granted access to `smpppd`. All hosts not within this range are denied access.

`password = password`

By assigning a password, limit the clients to authorized hosts. As this is a plain-text password, you should not overrate the security it provides. If no password is assigned, all clients are permitted to access `smpppd`.

`slp-register = yes/no`

With this parameter, the `smpppd` service can be announced in the network via SLP.

More information about `smpppd` is available in the `smpppd(8)` and `smpppd.conf(5)` man pages.

18.7.2 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`

`list of sites` where the front-ends search for `smpppd`. The front-ends test the options in the order specified here. `local` orders the establishment of a connection to the local `smpppd`. `gateway` points to an `smpppd` on the gateway. `config-file` indicates that the connection should be established to the `smpppd` specified in the `server` and `port` options in `/etc/smpppd-c.conf`. `slp` orders the front-ends to connect to an `smpppd` found via SLP.

`server = server`

The host on which smpppd runs.

`port = port`

The port on which smpppd runs.

`password = password`

The password selected for smpppd.

If smpppd is active, try to access it. For example, with `cinternet --verbose --interface-list`. In case of difficulties at this point, refer to the `smpppd-c.conf(5)` and `cinternet(8)` man pages.

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 Server 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 Server

Services that offer SLP support include cupsd, rsyncd, ypserv, openldap2, ksys-guardd, saned, kdm, vnc, login, smpppd, rpasswd, postfix, and sshd (via fish).

19.1 Installation

All packages necessary are installed by default. However, if you want to provide services via SLP, check that the package `openslp-server` is installed.

19.2 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 Server, 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.

19.3 SLP Front-Ends in SUSE Linux Enterprise Server

To find services provided via SLP in your network, use an SLP front-end. SUSE Linux Enterprise Server 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.

19.4 Installation over SLP

If you offer an installation server with SUSE Linux Enterprise Server installation media within your network, this can be registered with SLP. For details, see Section “Setting Up the Server Holding the Installation Sources” (Chapter 14, *Remote Installation, ↑Deployment Guide*). If SLP installation is selected, linuxrc starts an SLP inquiry after the system has booted from the selected boot medium and displays the sources found.

19.5 Providing Services via SLP

Many applications in SUSE Linux Enterprise Server 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 sane 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.

TIP: YaST and SLP

Some services brokered by YaST, such as an installation server or YOU server, perform this registration automatically when you activate SLP in the module dialogs. YaST then creates registration files for these services.

Static Registration with `/etc/slp.reg`

The only difference between this method and the procedure with `/etc/slp.reg.d` is that all services are grouped within a central file.

Dynamic Registration with `slptool`

If a service needs to be registered dynamically without the need of configuration files, use the `slptool` command line utility. The same utility can also be used to deregister an existing service offering without restarting `slpd`.

19.6 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 Server 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

20

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.

20.1 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.

20.1.1 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`.

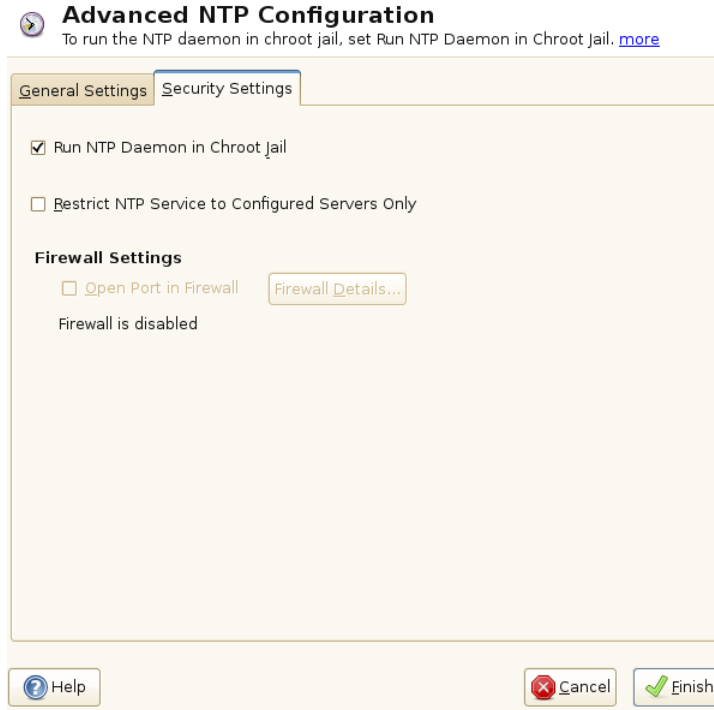
Outgoing Broadcast

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 20.1 *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`, as it prevents the attacker from compromising the entire system.

Restrict NTP Service to Configured Servers Only increases the security of your system by disallowing remote computers to view and modify NTP settings of your computer and to use the trap facility for remote event logging. Once enabled, these restrictions apply to all remote computers, unless you override the access control options for individual computers in the list of time sources in the *General Settings* tab. For all other remote computers, only querying for local time is allowed.

Enable *Open Port in Firewall* if `SuSEfirewall2` is active (which it is by default). If you leave the port closed, it is not possible to establish a connection to the time server.

20.2 Manually Configuring ntp in the Network

The easiest way to use a time server in the network is to set server parameters. For example, if a time server called `ntp.example.com` is reachable from the network, add its name to the file `/etc/ntp.conf` by adding the following line:

```
server ntp.example.com
```

To add more time servers, insert additional lines with the keyword `server`. After initializing `ntpd` with the command `rcntp start`, it takes about one hour until the time is stabilized and the drift file for correcting the local computer clock is created. With the drift file, the systematic error of the hardware clock can be computed as soon as the computer is powered on. The correction is used immediately, resulting in a higher stability of the system time.

There are two possible ways to use the NTP mechanism as a client: First, the client can query the time from a known server in regular intervals. With many clients, this approach can cause a high load on the server. Second, the client can wait for NTP broadcasts sent out by broadcast time servers in the network. This approach has the disadvantage that the quality of the server is unknown and a server sending out wrong information can cause severe problems.

If the time is obtained via broadcast, you do not need the server name. In this case, enter the line `broadcastclient` in the configuration file `/etc/ntp.conf`. To use one or more known time servers exclusively, enter their names in the line starting with `servers`.

20.3 Dynamic Time Synchronization at Runtime

If the system boots without network connection, `ntpd` starts up, but it cannot resolve DNS names of the time servers set in the configuration file. This can happen if you use Network Manager with an encrypted WLAN.

If you want `ntpd` to resolve DNS names at runtime, you must set the `dynamic` option. Then, when the network is established some time after booting, `ntpd` looks up the names again and can reach the time servers to get the time.

Manually edit `/etc/ntp.conf` and add `dynamic` to one or more `server` entries:

```
server ntp.example.com dynamic
```

Or use YaST and proceed as follows:

- 1 In YaST click *Network Services > NTP Configuration*.
- 2 Select the server you want to configure. Then click *Edit*.
- 3 Activate the *Options* field and add `dynamic`. Separate it with a space, if there are already other options entered.
- 4 Click *Ok* to close the edit dialog. Repeat the previous step to change all servers as wanted.
- 5 Finally click *Ok* to save the settings.

20.4 Setting Up a Local Reference Clock

The software package `ntp` contains drivers for connecting local reference clocks. A list of supported clocks is available in the `ntp-doc` package in the file `/usr/share/doc/packages/ntp-doc/refclock.html`. Every driver is associated with a number. In `ntp`, the actual configuration takes place by means of pseudo IP addresses. The clocks are entered in the file `/etc/ntp.conf` as though they existed in the network. For this purpose, they are assigned special IP addresses in the form `127.127.t.u`. Here, *t* stands for the type of the clock and determines which driver is used and *u* for the unit, which determines the interface used.

Normally, the individual drivers have special parameters that describe configuration details. The file `/usr/share/doc/packages/ntp-doc/drivers/driverNN.html` (where *NN* is the number of the driver) provides information about the particular type of clock. For example, the “type 8” clock (radio clock over serial interface) requires

an additional mode that specifies the clock more precisely. The Conrad DCF77 receiver module, for example, has mode 5. To use this clock as a preferred reference, specify the keyword `prefer`. The complete `server` line for a Conrad DCF77 receiver module would be:

```
server 127.127.8.0 mode 5 prefer
```

Other clocks follow the same pattern. Following the installation of the `ntp-doc` package, the documentation for `ntp` is available in the directory `/usr/share/doc/packages/ntp-doc`. The file `/usr/share/doc/packages/ntp-doc/refclock.html` provides links to the driver pages describing the driver parameters.

20.5 Clock Synchronization to an External Time Reference (ETR)

Support for clock synchronization to an external time reference (ETR) is available. The external time reference sends an oscillator signal and a synchronization signal every 2^{20} (2 to the power of 20) microseconds to keep TOD clocks of all connected servers in sync.

For availability two ETR units can be connected to a machine. If the clock deviates for more than the sync-check tolerance all CPUs get a machine check that indicates that the clock is out of sync. If this happens all DASD I/O to XRC enabled devices is stopped until the clock is synchronized again.

The ETR support is activated via two `sysfs` attributes; as `root` enter:

```
echo 1 > /sys/devices/system/etr/etr0/online
echo 1 > /sys/devices/system/etr/etr1/online
```


The Domain Name System

DNS (domain name system) is needed to resolve the domain names and hostnames into IP addresses. In this way, the IP address 192.168.2.100 is assigned to the hostname `jupiter`, for example. Before setting up your own name server, read the general information about DNS in Section 18.3, “Name Resolution” (page 231). The following configuration examples refer to BIND.

21.1 DNS Terminology

Zone

The domain namespace is divided into regions called zones. For instance, if you have `example.com`, you have the `example` section (or zone) of the `com` domain.

DNS server

The DNS server is a server that maintains the name and IP information for a domain. You can have a primary DNS server for master zone, a secondary server for slave zone, or a slave server without any zones for caching.

Master zone DNS server

The master zone includes all hosts from your network and a DNS server master zone stores up-to-date records for all the hosts in your domain.

Slave zone DNS server

A slave zone is a copy of the master zone. The slave zone DNS server obtains its zone data with zone transfer operations from its master server. The slave zone DNS server responds authoritatively for the zone as long as it has valid

(not expired) zone data. If the slave cannot obtain a new copy of the zone data, it stops responding for the zone.

Forwarder

Forwarders are DNS servers to which your DNS server should send queries it cannot answer. To enable different configuration sources in one configuration, `netconfig` is used (see also `man 8 netconfig`).

Record

The record is information about name and IP address. Supported records and their syntax are described in BIND documentation. Some special records are:

NS record

An NS record tells name servers which machines are in charge of a given domain zone.

MX record

The MX (mail exchange) records describe the machines to contact for directing mail across the Internet.

SOA record

SOA (Start of Authority) record is the first record in a zone file. The SOA record is used when using DNS to synchronize data between multiple computers.

21.2 Installation

To install a DNS server, start YaST and select *Software > Software Management*. Choose *Filter > Patterns* and select *DHCP and DNS Server*. Confirm the installation of the dependent packages to finish the installation process.

21.3 Configuration with YaST

You can use the DNS module of YaST to configure a DNS server for your local network. When starting the module for the first time, a wizard starts, prompting you to make just a few basic decisions concerning administration of the server. Completing this initial setup produces a very basic server configuration that should be functioning in its essential

aspects. The expert mode can be used to deal with more advanced configuration tasks, such as setting up ACLs, logging, TSIG keys, and other options.

21.3.1 Wizard Configuration

The wizard consists of three steps or dialogs. At the appropriate places in the dialogs, you are given the opportunity to enter the expert configuration mode.

- 1 When starting the module for the first time, the *Forwarder Settings* dialog, shown in Figure 21.1, “DNS Server Installation: Forwarder Settings” (page 291), opens. The *Netconfig DNS Policy* decides which devices should provide forwarders or whether you want to supply your own *Forwarder List*. For more information about netconfig, see `man 8 netconfig`.

Figure 21.1 *DNS Server Installation: Forwarder Settings*

DNS Server Installation: Forwarder Settings

Netconfig DNS policy Custom policy

auto auto

Add IP Address

IPv4 or IPv6 Address

192.168.27.1 Add

Forwarder List

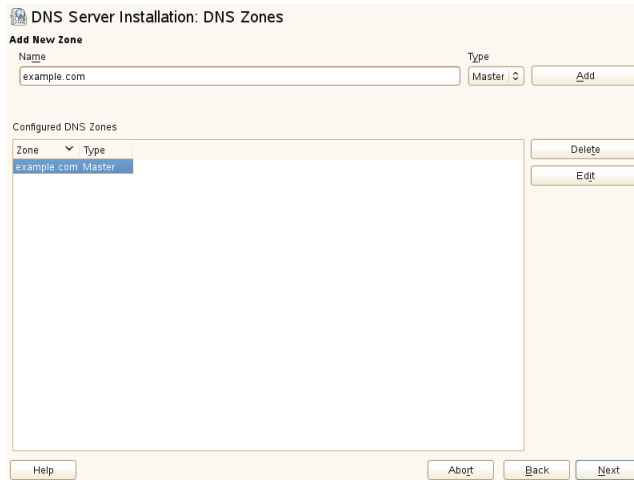
192.168.27.1 Delete

Help Cancel Back Next

- 2 The *DNS Zones* dialog consists of several parts and is responsible for the management of zone files, described in Section 21.6, “Zone Files” (page 306). For a new zone, provide a name for it in *Zone Name*. To add a reverse zone, the name must end in `.in-addr.arpa`. Finally, select the *Zone Type* (master or slave). See Figure 21.2, “DNS Server Installation: DNS Zones” (page 292). Click *Edit*

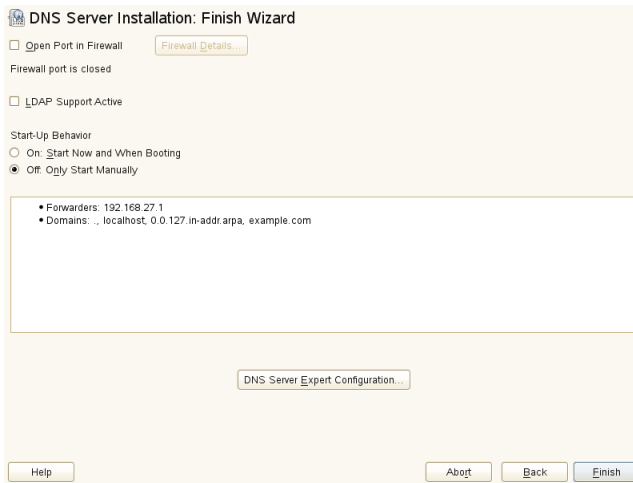
Zone to configure other settings of an existing zone. To remove a zone, click *Delete Zone*.

Figure 21.2 *DNS Server Installation: DNS Zones*



- 3 In the final dialog, you can open the DNS port in the firewall by clicking *Open Port in Firewall*. Then decide whether or not the DNS server should be started when booting (*On* or *Off*). You can also activate LDAP support. See Figure 21.3, “DNS Server Installation: Finish Wizard” (page 293).

Figure 21.3 *DNS Server Installation: Finish Wizard*



21.3.2 Expert Configuration

After starting the module, YaST opens a window displaying several configuration options. Completing it results in a DNS server configuration with the basic functions in place:

Start-Up

Under *Start-Up*, define whether the DNS server should be started on startup (during the booting the system) or manually. To start the DNS server immediately, select *Start DNS Server Now*. To stop the DNS server, select *Stop DNS Server Now*. To save the current settings, select *Save Settings and Restart DNS Server Now*. You can open the DNS port in the firewall with *Open Port in Firewall* and modify the firewall settings with *Firewall Details*.

By selecting *LDAP Support Active*, the zone files are managed by an LDAP database. Any changes to zone data written to the LDAP database are picked up by the DNS server as soon as it is restarted or prompted to reload its configuration.

Forwarders

If your local DNS server cannot answer a request, it tries to forward the request to a *Forwarder*, if configured so. This forwarder may be added manually to the *Forwarder List*. If the forwarder is not static like in dial-up connections, *netconfig* handles the configuration. For more information about *netconfig*, see `man 8 netconfig`.

Basic Options

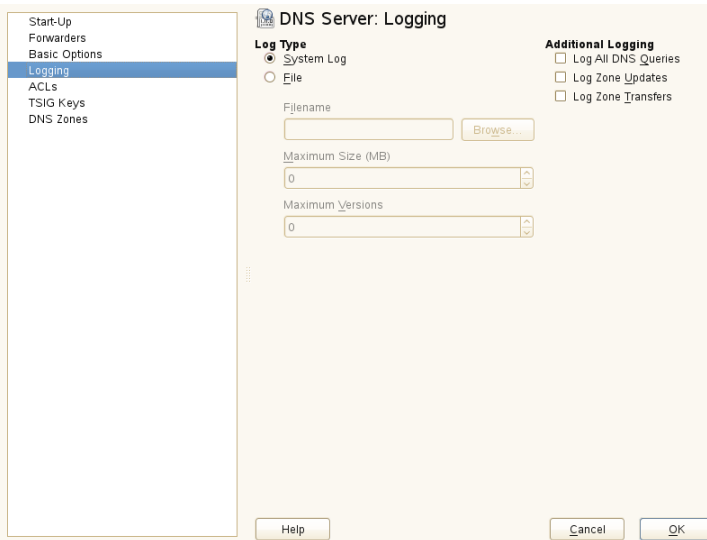
In this section, set basic server options. From the *Option* menu, select the desired item then specify the value in the corresponding entry field. Include the new entry by selecting *Add*.

Logging

To set what the DNS server should log and how, select *Logging*. Under *Log Type*, specify where the DNS server should write the log data. Use the systemwide log file `/var/log/messages` by selecting *System Log* or specify a different file by selecting *File*. In the latter case, additionally specify a name, the maximum file size in megabytes and the number of logfile versions to store.

Further options are available under *Additional Logging*. Enabling *Log All DNS Queries* causes *every* query to be logged, in which case the log file could grow extremely large. For this reason, it is not a good idea to enable this option for other than debugging purposes. To log the data traffic during zone updates between DHCP and DNS server, enable *Log Zone Updates*. To log the data traffic during a zone transfer from master to slave, enable *Log Zone Transfer*. See Figure 21.4, “DNS Server: Logging” (page 295).

Figure 21.4 *DNS Server: Logging*



Using ACLs

Use this window to define ACLs (access control lists) to enforce access restrictions. After providing a distinct name under *Name*, specify an IP address (with or without netmask) under *Value* in the following fashion:

```
{ 192.168.1/24; }
```

The syntax of the configuration file requires that the address ends with a semicolon and is put into curly braces.

TSIG Keys

The main purpose of TSIGs (transaction signatures) is to secure communications between DHCP and DNS servers. They are described in Section 21.8, “Secure Transactions” (page 310).

To generate a TSIG key, enter a distinctive name in the field labeled *Key ID* and specify the file where the key should be stored (*Filename*). Confirm your choices with *Add*.

To use a previously created key, leave the *Key ID* field blank and select the file where it is stored under *File Name*. After that, confirm with *Add*.

Adding a Slave Zone

To add a slave zone, select *DNS Zones*, choose the zone type *Slave*, write the name of the new zone, and click *Add*.

In the *Zone Editor* under *Master DNS Server IP*, specify the master from which the slave should pull its data. To limit access to the server, select one of the ACLs from the list.

Adding a Master Zone

To add a master zone, select *DNS Zones*, choose the zone type *Master*, write the name of the new zone, and click *Add*. When adding a master zone, a reverse zone is also needed. For example, when adding the zone `example.com` that points to hosts in a subnet `192.168.1.0/24`, you should also add a reverse zone for the IP-address range covered. By definition, this should be named `1.168.192.in-addr.arpa`.

Editing a Master Zone

To edit a master zone, select *DNS Zones*, select the master zone from the table, and click *Edit*. The dialog consists of several pages: *Basics* (the one opened first), *NS Records*, *MX Records*, *SOA*, and *Records*.

The basic dialog, shown in Figure 21.5, “DNS Server: Zone Editor (Basic)” (page 297), lets you define settings for dynamic DNS and access options for zone transfers to clients and slave name servers. To permit the dynamic updating of zones, select *Allow Dynamic Updates* as well as the corresponding TSIG key. The key must have been defined before the update action starts. To enable zone transfers, select the corresponding ACLs. ACLs must have been defined already.

In the basic dialog, select whether to enable zone transfers. Use the listed ACLs to define who can download zones.

Figure 21.5 *DNS Server: Zone Editor (Basic)*

The screenshot shows the 'Zone Editor' dialog box with the 'Basics' tab selected. The 'Settings for Zone' field contains 'example.com'. Below the tabs, the 'Allow Dynamic Updates' checkbox is checked, and the 'TSIG Key' field contains 'asdfg'. The 'Enable Zone Transport' checkbox is also checked, and the 'ACLs' section has the 'any' checkbox checked. At the bottom, there are 'Help', 'Cancel', 'Back', and 'OK' buttons.

Zone Editor

Settings for Zone:

Basics | NS Records | MX Records | SOA | Records

☒ **Allow Dynamic Updates**

TSIG Key:

☒ **Enable Zone Transport**

ACLs

- ☒ any
- ☐ localhost
- ☐ localnets

Zone Editor (NS Records)

This dialog allows you to define alternative name servers for the zones specified. Make sure that your own name server is included in the list. To add a record, enter its name under *Name Server to Add* then confirm with *Add*. See Figure 21.6, “DNS Server: Zone Editor (NS Records)” (page 298).

Figure 21.6 DNS Server: Zone Editor (NS Records)

The screenshot shows the 'Zone Editor' window for 'example.com'. The 'NS Records' tab is selected. It features a 'Name Server to Add' input field with an 'Add' button. Below is a 'Name Server List' table with a 'Delete' button. The bottom of the window has 'Help', 'Cancel', 'Back', and 'OK' buttons.

Name Server List	
------------------	--

Zone Editor (MX Records)

To add a mail server for the current zone to the existing list, enter the corresponding address and priority value. After doing so, confirm by selecting *Add*. See Figure 21.7, “DNS Server: Zone Editor (MX Records)” (page 298).

Figure 21.7 DNS Server: Zone Editor (MX Records)

The screenshot shows the 'Zone Editor' window for 'example.com' with the 'MX Records' tab selected. It includes a 'Mail Server to Add' section with 'Address' and 'Priority' input fields, an 'Add' button, and a 'Mail Relay List' table with 'Mail Server' and 'Priority' columns, a 'Delete' button, and a 'Help' button at the bottom.

Mail Relay List	
Mail Server	Priority

Zone Editor (SOA)

This page allows you to create SOA (start of authority) records. For an explanation of the individual options, refer to Example 21.6, “File /var/lib/named/example.com.zone” (page 306). Changing SOA records is not supported for dynamic zones managed via LDAP.

Figure 21.8 DNS Server: Zone Editor (SOA)

The screenshot shows the 'Zone Editor' window with the 'SOA' tab selected. The 'Settings for Zone' field at the top contains 'example.com'. The 'SOA' tab contains the following fields and values:

Field	Value	Unit
Serial	2010032500	
TTL	2	Days
Refresh	3	Hours
Retry	1	Hours
Expiration	1	Weeks
Minimum	1	Days

At the bottom of the dialog are buttons for 'Help', 'Cancel', 'Back', and 'OK'.

Zone Editor (Records)

This dialog manages name resolution. In *Record Key*, enter the hostname then select its type. *A-Record* represents the main entry. The value for this should be an IP address. *CNAME* is an alias. Use the types *NS* and *MX* for detailed or partial records that expand on the information provided in the *NS Records* and *MX Records* tabs. These three types resolve to an existing *A* record. *PTR* is for reverse zones. It is the opposite of an *A* record, for example:

```
hostname.example.com. IN A 192.168.0.1
1.0.168.192.in-addr.arpa IN PTR hostname.example.com.
```

NOTE: Editing the Reverse Zone

After adding a forward zone, go back to the main menu and select the reverse zone for editing. There in the tab *Basics* activate the checkbox *Automatically*

Generate Records From and select your forward zone. That way, all changes to the forward zone are automatically updated in the reverse zone.

21.4 Starting the Name Server BIND

On a SUSE® Linux Enterprise Server system, the name server BIND (*Berkeley Internet Name Domain*) comes preconfigured so it can be started right after installation without any problem. If you already have a functioning Internet connection and have entered 127.0.0.1 as the name server address for `localhost` in `/etc/resolv.conf`, you normally already have a working name resolution without needing to know the DNS of the provider. BIND carries out name resolution via the root name server, a notably slower process. Normally, the DNS of the provider should be entered with its IP address in the configuration file `/etc/named.conf` under `forwarders` to ensure effective and secure name resolution. If this works so far, the name server runs as a pure *caching-only* name server. Only when you configure its own zones will it become a proper DNS. A simple example of this is included in the documentation in `/usr/share/doc/packages/bind/config`.

TIP: Automatic Adaptation of the Name Server Information

Depending on the type of Internet connection or the network connection, the name server information can automatically be adapted to the current conditions. To do this, set the variable `MODIFY_NAMED_CONF_DYNAMICALY` in the `/etc/sysconfig/network/config` file to `yes`.

However, do not set up any official domains until assigned one by the responsible institution. Even if you have your own domain and it is managed by the provider, you are better off not using it, because BIND would otherwise not forward requests for this domain. The Web server at the provider, for example, would not be accessible for this domain.

To start the name server, enter the command `rcnamed start` as root. If “done” appears to the right in green then `named` (as the name server process is called) has been started successfully. Test the name server immediately on the local system with the `host` or `dig` programs, which should return `localhost` as the default server with the address 127.0.0.1. If this is not the case, `/etc/resolv.conf` probably contains an incorrect name server entry or the file does not exist at all. For the first test,

enter `host 127.0.0.1`, which should always work. If you get an error message, use `rndns status` to see whether the server is actually running. If the name server does not start or behaves unexpectedly, you can usually find the cause in the log file `/var/log/messages`.

To use the name server of the provider (or one already running on your network) as the forwarder, enter the corresponding IP address or addresses in the `options` section under `forwarders`. The addresses included in Example 21.1, “Forwarding Options in `named.conf`” (page 301) are just examples. Adjust these entries to your own setup.

Example 21.1 *Forwarding Options in `named.conf`*

```
options {  
    directory "/var/lib/named";  
    forwarders { 10.11.12.13; 10.11.12.14; };  
    listen-on { 127.0.0.1; 192.168.1.116; };  
    allow-query { 127/8; 192.168/16 };  
    notify no;  
};
```

The `options` entry is followed by entries for the zone, `localhost`, and `0.0.127.in-addr.arpa`. The `type hint` entry under “.” should always be present. The corresponding files do not need to be modified and should work as they are. Also make sure that each entry is closed with a “;” and that the curly braces are in the correct places. After changing the configuration file `/etc/named.conf` or the zone files, tell BIND to reread them with `rndns reload`. Achieve the same by stopping and restarting the name server with `rndns restart`. Stop the server at any time by entering `rndns stop`.

21.5 The Configuration File `/etc/named.conf`

All the settings for the BIND name server itself are stored in the `/etc/named.conf` file. However, the zone data for the domains to handle (consisting of the hostnames, IP addresses, and so on) are stored in separate files in the `/var/lib/named` directory. The details of this are described later.

`/etc/named.conf` is roughly divided into two areas. One is the `options` section for general settings and the other consists of `zone` entries for the individual domains.

A logging section and `acl` (access control list) entries are optional. Comment lines begin with a `#` sign or `//`. A minimal `/etc/named.conf` is shown in Example 21.2, “A Basic `/etc/named.conf`” (page 302).

Example 21.2 A Basic `/etc/named.conf`

```
options {
    directory "/var/lib/named";
    forwarders { 10.0.0.1; };
    notify no;
};

zone "localhost" in {
    type master;
    file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" in {
    type master;
    file "127.0.0.zone";
};

zone "." in {
    type hint;
    file "root.hint";
};
```

21.5.1 Important Configuration Options

`directory "filename";`

Specifies the directory in which BIND can find the files containing the zone data. Usually, this is `/var/lib/named`.

`forwarders { ip-address; };`

Specifies the name servers (mostly of the provider) to which DNS requests should be forwarded if they cannot be resolved directly. Replace *ip-address* with an IP address like `192.168.1.116`.

`forward first;`

Causes DNS requests to be forwarded before an attempt is made to resolve them via the root name servers. Instead of `forward first`, `forward only` can be written to have all requests forwarded and none sent to the root name servers. This makes sense for firewall configurations.

`listen-on port 53 { 127.0.0.1; ip-address; };`

Tells BIND on which network interfaces and port to accept client queries. `port 53` does not need to be specified explicitly, because `53` is the default port. Enter `127.0.0.1` to permit requests from the local host. If you omit this entry entirely, all interfaces are used by default.

`listen-on-v6 port 53 {any; };`

Tells BIND on which port it should listen for IPv6 client requests. The only alternative to `any` is `none`. As far as IPv6 is concerned, the server only accepts wild card addresses.

`query-source address * port 53;`

This entry is necessary if a firewall is blocking outgoing DNS requests. This tells BIND to post requests externally from port 53 and not from any of the high ports above 1024.

`query-source-v6 address * port 53;`

Tells BIND which port to use for IPv6 queries.

`allow-query { 127.0.0.1; net; };`

Defines the networks from which clients can post DNS requests. Replace *net* with address information like `192.168.2.0/24`. The `/24` at the end is an abbreviated expression for the netmask (in this case `255.255.255.0`).

`allow-transfer ! *;;`

Controls which hosts can request zone transfers. In the example, such requests are completely denied with `! *`. Without this entry, zone transfers can be requested from anywhere without restrictions.

`statistics-interval 0;`

In the absence of this entry, BIND generates several lines of statistical information per hour in `/var/log/messages`. Set it to 0 to suppress these statistics completely or set an interval in minutes.

`cleaning-interval 720;`

This option defines at which time intervals BIND clears its cache. This triggers an entry in `/var/log/messages` each time it occurs. The time specification is in minutes. The default is 60 minutes.

`interface-interval 0;`

BIND regularly searches the network interfaces for new or nonexistent interfaces. If this value is set to 0, this is not done and BIND only listens at the interfaces detected at start-up. Otherwise, the interval can be defined in minutes. The default is sixty minutes.

`notify no;`

`no` prevents other name servers from being informed when changes are made to the zone data or when the name server is restarted.

For a list of available options, read the manual page `man 5 named.conf`.

21.5.2 Logging

What, how, and where logging takes place can be extensively configured in BIND. Normally, the default settings should be sufficient. Example 21.3, “Entry to Disable Logging” (page 304), shows the simplest form of such an entry and completely suppresses any logging.

Example 21.3 *Entry to Disable Logging*

```
logging {  
    category default { null; };  
};
```

21.5.3 Zone Entries

Example 21.4 *Zone Entry for example.com*

```
zone "example.com" in {  
    type master;  
    file "example.com.zone";  
    notify no;  
};
```

After `zone`, specify the name of the domain to administer (`example.com`) followed by `in` and a block of relevant options enclosed in curly braces, as shown in Example 21.4, “Zone Entry for example.com” (page 304). To define a *slave zone*, switch the `type` to `slave` and specify a name server that administers this zone as `master` (which, in turn, may be a slave of another master), as shown in Example 21.5, “Zone Entry for example.net” (page 305).

Example 21.5 *Zone Entry for example.net*

```
zone "example.net" in {  
    type slave;  
    file "slave/example.net.zone";  
    masters { 10.0.0.1; };  
};
```

The zone options:

`type master;`

By specifying `master`, tell BIND that the zone is handled by the local name server. This assumes that a zone file has been created in the correct format.

`type slave;`

This zone is transferred from another name server. It must be used together with `masters`.

`type hint;`

The zone `.` of the `hint` type is used to set the root name servers. This zone definition can be left as is.

`file example.com.zone` or file `"slave/example.net.zone"`;

This entry specifies the file where zone data for the domain is located. This file is not required for a slave, because this data is pulled from another name server. To differentiate master and slave files, use the directory `slave` for the slave files.

`masters { server-ip-address; };`

This entry is only needed for slave zones. It specifies from which name server the zone file should be transferred.

`allow-update {! *; };`

This option controls external write access, which would allow clients to make a DNS entry—something not normally desirable for security reasons. Without this entry, zone updates are not allowed at all. The above entry achieves the same because `! *` effectively bans any such activity.

21.6 Zone Files

Two types of zone files are needed. One assigns IP addresses to hostnames and the other does the reverse: it supplies a hostname for an IP address.

TIP: Using the Dot (Period, Fullstop) in Zone Files

The "." has an important meaning in the zone files. If hostnames are given without a final ., the zone is appended. Complete hostnames specified with a full domain name must end with a . to avoid having the domain added to it again. A missing or wrongly placed "." is probably the most frequent cause of name server configuration errors.

The first case to consider is the zone file `example.com.zone`, responsible for the domain `example.com`, shown in Example 21.6, “File `/var/lib/named/example.com.zone`” (page 306).

Example 21.6 *File `/var/lib/named/example.com.zone`*

```
1. $TTL 2D
2. example.com. IN SOA      dns root.example.com. (
3.                2003072441 ; serial
4.                1D        ; refresh
5.                2H        ; retry
6.                1W        ; expiry
7.                2D )      ; minimum
8.
9.                IN NS     dns
10.               IN MX     10 mail
11.
12. gate          IN A      192.168.5.1
13.              IN A      10.0.0.1
14. dns           IN A      192.168.1.116
15. mail          IN A      192.168.3.108
16. jupiter       IN A      192.168.2.100
17. venus         IN A      192.168.2.101
18. saturn        IN A      192.168.2.102
19. mercury       IN A      192.168.2.103
20. ntp           IN CNAME  dns
21. dns6          IN A6     0 2002:c0a8:174::
```

Line 1:

\$TTL defines the default time to live that should apply to all the entries in this file.

In this example, entries are valid for a period of two days (2 D).

Line 2:

This is where the SOA (start of authority) control record begins:

- The name of the domain to administer is `example.com` in the first position. This ends with `" . "`, because otherwise the zone would be appended a second time. Alternatively, `@` can be entered here, in which case the zone would be extracted from the corresponding entry in `/etc/named.conf`.
- After `IN` SOA is the name of the name server in charge as master for this zone. The name is expanded from `dns` to `dns.example.com`, because it does not end with a `" . "`.
- An e-mail address of the person in charge of this name server follows. Because the `@` sign already has a special meaning, `" . "` is entered here instead. For `root@example.com` the entry must read `root.example.com..` The `" . "` must be included at the end to prevent the zone from being added.
- The `(` includes all lines up to `)` into the SOA record.

Line 3:

The `serial number` is an arbitrary number that is increased each time this file is changed. It is needed to inform the secondary name servers (slave servers) of changes. For this, a 10 digit number of the date and run number, written as `YYYYMMDDNN`, has become the customary format.

Line 4:

The `refresh rate` specifies the time interval at which the secondary name servers verify the zone `serial number`. In this case, one day.

Line 5:

The `retry rate` specifies the time interval at which a secondary name server, in case of error, attempts to contact the primary server again. Here, two hours.

Line 6:

The `expiration time` specifies the time frame after which a secondary name server discards the cached data if it has not regained contact to the primary server. Here, a week.

Line 7:

The last entry in the SOA record specifies the `negative caching TTL`—the time for which results of unresolved DNS queries from other servers may be cached.

Line 9:

The `IN NS` specifies the name server responsible for this domain. `dns` is extended to `dns.example.com` because it does not end with a `"."`. There can be several lines like this—one for the primary and one for each secondary name server. If `notify` is not set to `no` in `/etc/named.conf`, all the name servers listed here are informed of the changes made to the zone data.

Line 10:

The `MX` record specifies the mail server that accepts, processes, and forwards e-mails for the domain `example.com`. In this example, this is the host `mail.example.com`. The number in front of the hostname is the preference value. If there are multiple `MX` entries, the mail server with the smallest value is taken first and, if mail delivery to this server fails, an attempt is made with the next higher value.

Lines 12–19:

These are the actual address records where one or more IP addresses are assigned to hostnames. The names are listed here without a `"."` because they do not include their domain, so `example.com` is added to all of them. Two IP addresses are assigned to the host `gate`, as it has two network cards. Wherever the host address is a traditional one (IPv4), the record is marked with `A`. If the address is an IPv6 address, the entry is marked with `AAAA`.

NOTE: IPv6 Syntax

The IPv6 record has a slightly different syntax than IPv4. Because of the fragmentation possibility, it is necessary to provide information about missed bits before the address. To just fill up the IPv6 address with the needed number of “0”, add two colons at the correct place in the address.

```
pluto      AAAA 2345:00C1:CA11::1234:5678:9ABC:DEF0
pluto      AAAA 2345:00D2:DA11::1234:5678:9ABC:DEF0
```

Line 20:

The alias `ntp` can be used to address `dns` (`CNAME` means *canonical name*).

The pseudodomain `in-addr.arpa` is used for the reverse lookup of IP addresses into hostnames. It is appended to the network part of the address in reverse notation. So `192.168` is resolved into `168.192.in-addr.arpa`. See Example 21.7, “Reverse Lookup” (page 309).

Example 21.7 *Reverse Lookup*

```
1. $TTL 2D
2. 168.192.in-addr.arpa.    IN SOA dns.example.com. root.example.com. (
3.                          2003072441      ; serial
4.                          1D              ; refresh
5.                          2H              ; retry
6.                          1W              ; expiry
7.                          2D )            ; minimum
8.
9.                          IN NS          dns.example.com.
10.
11. 1.5                     IN PTR         gate.example.com.
12. 100.3                   IN PTR         www.example.com.
13. 253.2                   IN PTR         cups.example.com.
```

Line 1:

\$TTL defines the standard TTL that applies to all entries here.

Line 2:

The configuration file should activate reverse lookup for the network `192.168`. Given that the zone is called `168.192.in-addr.arpa`, it should not be added to the hostnames. Therefore, all hostnames are entered in their complete form—with their domain and with a “.” at the end. The remaining entries correspond to those described for the previous `example.com` example.

Lines 3–7:

See the previous example for `example.com`.

Line 9:

Again this line specifies the name server responsible for this zone. This time, however, the name is entered in its complete form with the domain and a “.” at the end.

Lines 11–13:

These are the pointer records hinting at the IP addresses on the respective hosts. Only the last part of the IP address is entered at the beginning of the line, without

the "." at the end. Appending the zone to this (without the `.in-addr.arpa`) results in the complete IP address in reverse order.

Normally, zone transfers between different versions of BIND should be possible without any problem.

21.7 Dynamic Update of Zone Data

The term *dynamic update* refers to operations by which entries in the zone files of a master server are added, changed, or deleted. This mechanism is described in RFC 2136. Dynamic update is configured individually for each zone entry by adding an optional `allow-update` or `update-policy` rule. Zones to update dynamically should not be edited by hand.

Transmit the entries to update to the server with the command `nsupdate`. For the exact syntax of this command, check the manual page for `nsupdate` (`man 8 nsupdate`). For security reasons, any such update should be performed using TSIG keys as described in Section 21.8, “Secure Transactions” (page 310).

21.8 Secure Transactions

Secure transactions can be made with the help of transaction signatures (TSIGs) based on shared secret keys (also called TSIG keys). This section describes how to generate and use such keys.

Secure transactions are needed for communication between different servers and for the dynamic update of zone data. Making the access control dependent on keys is much more secure than merely relying on IP addresses.

Generate a TSIG key with the following command (for details, see `man dnssec-keygen`):

```
dnssec-keygen -a hmac-md5 -b 128 -n HOST host1-host2
```

This creates two files with names similar to these:

```
Khost1-host2.+157+34265.private Khost1-host2.+157+34265.key
```

The key itself (a string like `ejIkuCyyGJwwuN3xAteKgg==`) is found in both files. To use it for transactions, the second file (`Khost1-host2.+157+34265.key`) must be transferred to the remote host, preferably in a secure way (using `scp`, for example). On the remote server, the key must be included in the `/etc/named.conf` file to enable a secure communication between `host1` and `host2`:

```
key host1-host2 {
    algorithm hmac-md5;
    secret "ejIkuCyyGJwwuN3xAteKgg=";
};
```

WARNING: File Permissions of `/etc/named.conf`

Make sure that the permissions of `/etc/named.conf` are properly restricted. The default for this file is `0640`, with the owner being `root` and the group `named`. As an alternative, move the keys to an extra file with specially limited permissions, which is then included from `/etc/named.conf`. To include an external file, use:

```
include "filename"
```

Replace `filename` with an absolute path to your file with keys.

To enable the server `host1` to use the key for `host2` (which has the address `10.1.2.3` in this example), the server's `/etc/named.conf` must include the following rule:

```
server 10.1.2.3 {
    keys { host1-host2. ; };
};
```

Analogous entries must be included in the configuration files of `host2`.

Add TSIG keys for any ACLs (access control lists, not to be confused with file system ACLs) that are defined for IP addresses and address ranges to enable transaction security. The corresponding entry could look like this:

```
allow-update { key host1-host2. ;};
```

This topic is discussed in more detail in the *BIND Administrator Reference Manual* under `update-policy`.

21.9 DNS Security

DNSSEC, or DNS security, is described in RFC 2535. The tools available for DNSSEC are discussed in the BIND Manual.

A zone considered secure must have one or several zone keys associated with it. These are generated with `dnssec-keygen`, just like the host keys. The DSA encryption algorithm is currently used to generate these keys. The public keys generated should be included in the corresponding zone file with an `$INCLUDE` rule.

With the command `dnssec-makekeyset`, all keys generated are packaged into one set, which must then be transferred to the parent zone in a secure manner. On the parent, the set is signed with `dnssec-signkey`. The files generated by this command are then used to sign the zones with `dnssec-signzone`, which in turn generates the files to include for each zone in `/etc/named.conf`.

21.10 For More Information

For additional information, refer to the *BIND Administrator Reference Manual* from package `bind-doc`, which is installed under `/usr/share/doc/packages/bind/`. Consider additionally consulting the RFCs referenced by the manual and the manual pages included with BIND. `/usr/share/doc/packages/bind/README`. SuSE contains up-to-date information about BIND in SUSE Linux Enterprise Server.

DHCP

The purpose of the *dynamic host configuration protocol* (DHCP) is to assign network settings centrally (from a server) rather than configuring them locally on each and every workstation. A host configured to use DHCP does not have control over its own static address. It is enabled to configure itself completely and automatically according to directions from the server. If you use the NetworkManager on the client side, you do not need to configure the client at all. This is useful if you have changing environments and only one interface active at a time. Never use NetworkManager on a machine that runs a DHCP server.

TIP: IBM System z: DHCP Support

On IBM System z platforms, DHCP only works on interfaces using the OSA and OSA Express network cards. These cards are the only ones with a MAC, which is required for the DHCP autoconfiguration features.

One way to configure a DHCP server is to identify each client using the hardware address of its network card (which should be fixed in most cases), then supply that client with identical settings each time it connects to the server. DHCP can also be configured to assign addresses to each relevant client dynamically from an address pool set up for this purpose. In the latter case, the DHCP server tries to assign the same address to the client each time it receives a request, even over extended periods. This works only if the network does not have more clients than addresses.

DHCP makes life easier for system administrators. Any changes, even bigger ones, related to addresses and the network configuration in general can be implemented centrally by editing the server's configuration file. This is much more convenient than reconfig-

uring numerous workstations. It is also much easier to integrate machines, particularly new machines, into the network, because they can be given an IP address from the pool. Retrieving the appropriate network settings from a DHCP server is especially useful in the case of laptops regularly used in different networks.

In this chapter, the DHCP server will run in the same subnet as the workstations, 192.168.2.0/24 with 192.168.2.1 as gateway. It has the fixed IP address 192.168.2.254 and serves two address ranges, 192.168.2.10 to 192.168.2.20 and 192.168.2.100 to 192.168.2.200.

A DHCP server supplies not only the IP address and the netmask, but also the hostname, domain name, gateway, and name server addresses for the client to use. In addition to that, DHCP allows a number of other parameters to be configured in a centralized way, for example, a time server from which clients may poll the current time or even a print server.

22.1 Configuring a DHCP Server with YaST

IMPORTANT: LDAP Support

In this version of SUSE Linux Enterprise Server, the YaST DHCP module can be set up to store the server configuration locally (on the host that runs the DHCP server) or to have its configuration data managed by an LDAP server. If you want to use LDAP, set up your LDAP environment before configuring the DHCP server.

The YaST DHCP module allows you to set up your own DHCP server for the local network. The module can run in simple mode or expert mode.

22.1.1 Initial Configuration (Wizard)

When the module is started for the first time, a wizard starts, prompting you to make a few basic decisions concerning server administration. Completing this initial setup produces a very basic server configuration that should function in its essential aspects. The expert mode can be used to deal with more advanced configuration tasks.

Card Selection

In the first step, YaST looks for the network interfaces available on your system, then displays them in a list. From the list, select the interface on which the DHCP server should listen and click *Select*. After this, select *Open Firewall for Selected Interfaces* to open the firewall for this interface, and click *Next*. See Figure 22.1, “DHCP Server: Card Selection” (page 315).

Figure 22.1 *DHCP Server: Card Selection*

DHCP Server Wizard (1 of 4): Card Selection

Network Cards for DHCP Server

Selected	Interface Name	Device Name	IP
<input checked="" type="checkbox"/>	eth0	VT6105/VT6106S [Rhine-III]	192.168.2.254

☒ Open Firewall for Selected Interfaces

Global Settings

Use the check box to determine whether your DHCP settings should be automatically stored by an LDAP server. In the entry fields, provide the network specifics for all clients the DHCP server should manage. These specifics are the domain name, address of a time server, addresses of the primary and secondary name server, addresses of a print and a WINS server (for a mixed network with both Windows and Linux clients), gateway address, and lease time. See Figure 22.2, “DHCP Server: Global Settings” (page 316).

Figure 22.2 *DHCP Server: Global Settings*

DHCP Server Wizard (2 of 4): Global Settings

☐ LDAP Support

DHCP Server Name (optional)

Domain Name

NTP Time Server

Primary Name Server IP

Print Server

Secondary Name Server IP

WINS Server

Default Gateway (Router)

Default Lease Time

Units
Hours

Dynamic DHCP

In this step, configure how dynamic IP addresses should be assigned to clients. To do so, specify an IP range from which the server can assign addresses to DHCP clients. All these addresses must be covered by the same netmask. Also specify the lease time during which a client may keep its IP address without needing to request an extension of the lease. Optionally, specify the maximum lease time—the period during which the server reserves an IP address for a particular client. See Figure 22.3, “DHCP Server: Dynamic DHCP” (page 317).

Figure 22.3 *DHCP Server: Dynamic DHCP*

The screenshot shows the 'DHCP Server Wizard (3 of 4): Dynamic DHCP' window. It is divided into several sections for configuring the DHCP server. The 'Subnet Information' section contains fields for 'Current Network' (192.168.2.0), 'Current Netmask' (255.255.255.0), 'Netmask Bits' (24), 'Minimum IP Address' (192.168.2.1), and 'Maximum IP Address' (192.168.2.254). The 'IP Address Range' section has 'First IP Address' (192.168.2.100) and 'Last IP Address' (192.168.2.128), with an unchecked checkbox for 'Allow Dynamic BOOTP'. The 'Lease Time' section shows a 'Default' of 4 'Hours' and a 'Maximum' of 2 'Days'. At the bottom, there is a 'Synchronize DNS Server...' dropdown menu and three buttons: 'Help', 'Abort', and 'Back', followed by a 'Next' button.

Subnet Information			
Current Network	Current Netmask	Netmask Bits	
192.168.2.0	255.255.255.0	24	
Minimum IP Address	Maximum IP Address		
192.168.2.1	192.168.2.254		

IP Address Range	
First IP Address	Last IP Address
192.168.2.100	192.168.2.128
<input type="checkbox"/> Allow Dynamic BOOTP	

Lease Time			
Default	Units	Maximum	Units
4	Hours	2	Days

Synchronize DNS Server... ▾

Help Abort Back Next

Finishing the Configuration and Setting the Start Mode

After the third part of the configuration wizard, a last dialog is shown in which you can define how the DHCP server should be started. Here, specify whether to start the DHCP server automatically when the system is booted or manually when needed (for example, for test purposes). Click *Finish* to complete the configuration of the server. See Figure 22.4, “DHCP Server: Start-Up” (page 318).

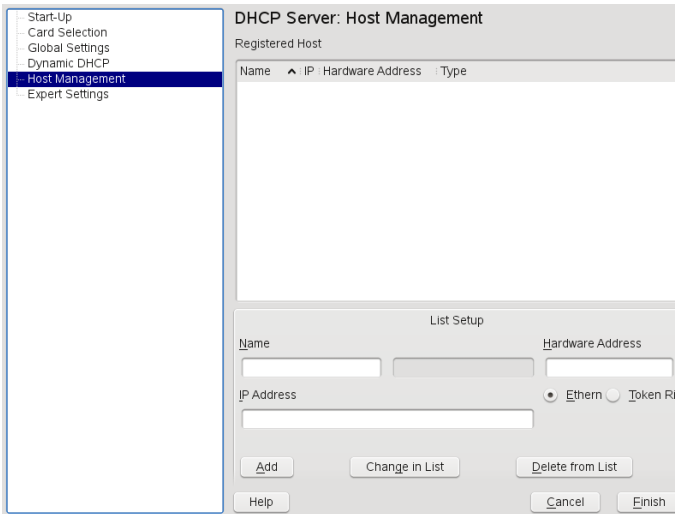
Figure 22.4 *DHCP Server: Start-Up*



Host Management

Instead of using dynamic DHCP in the way described in the preceding sections, you can also configure the server to assign addresses in quasi-static fashion. To do so, use the entry fields provided in the lower part to specify a list of the clients to manage in this way. Specifically, provide the *Name* and the *IP Address* to give to such a client, the *Hardware Address*, and the *Network Type* (token ring or ethernet). Modify the list of clients, which is shown in the upper part, with *Add*, *Edit*, and *Delete from List*. See Figure 22.5, “DHCP Server: Host Management” (page 319).

Figure 22.5 *DHCP Server: Host Management*



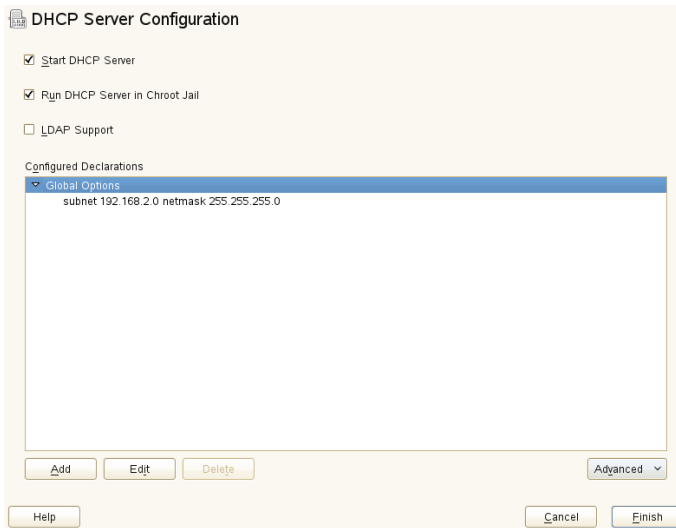
22.1.2 Expert Configuration

In addition to the configuration method discussed earlier, there is also an expert configuration mode that allows you to tweak the DHCP server setup in every detail. Start the expert configuration by selecting *Expert Settings* in the tree view in the left part of the dialog.

Chroot Environment and Declarations

In this first dialog, make the existing configuration editable by selecting *Start DHCP Server*. An important feature of the behavior of the DHCP server is its ability to run in a chroot environment, or chroot jail, to secure the server host. If the DHCP server should ever be compromised by an outside attack, the attacker will still be behind bars in the chroot jail, which prevents him from touching the rest of the system. The lower part of the dialog displays a tree view with the declarations that have already been defined. Modify these with *Add*, *Delete*, and *Edit*. Selecting *Advanced* takes you to additional expert dialogs. See Figure 22.6, “DHCP Server: Chroot Jail and Declarations” (page 320). After selecting *Add*, define the type of declaration to add. With *Advanced*, view the log file of the server, configure TSIG key management, and adjust the configuration of the firewall according to the setup of the DHCP server.

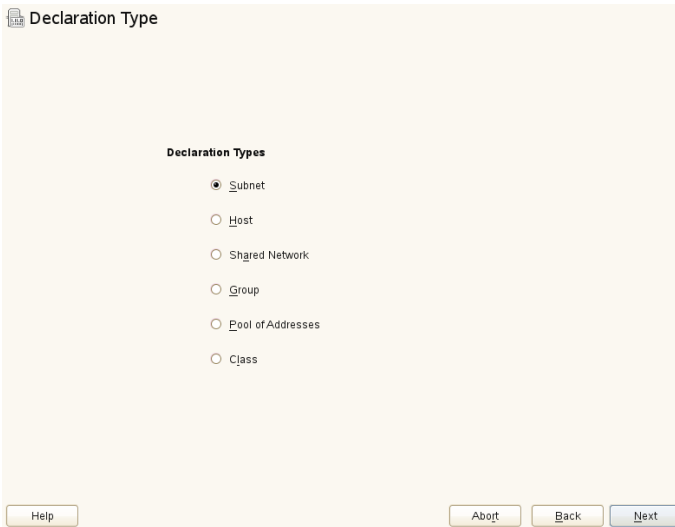
Figure 22.6 *DHCP Server: Chroot Jail and Declarations*



Selecting the Declaration Type

The *Global Options* of the DHCP server are made up of a number of declarations. This dialog lets you set the declaration types *Subnet*, *Host*, *Shared Network*, *Group*, *Pool of Addresses*, and *Class*. This example shows the selection of a new subnetwork (see Figure 22.7, “DHCP Server: Selecting a Declaration Type” (page 321)).

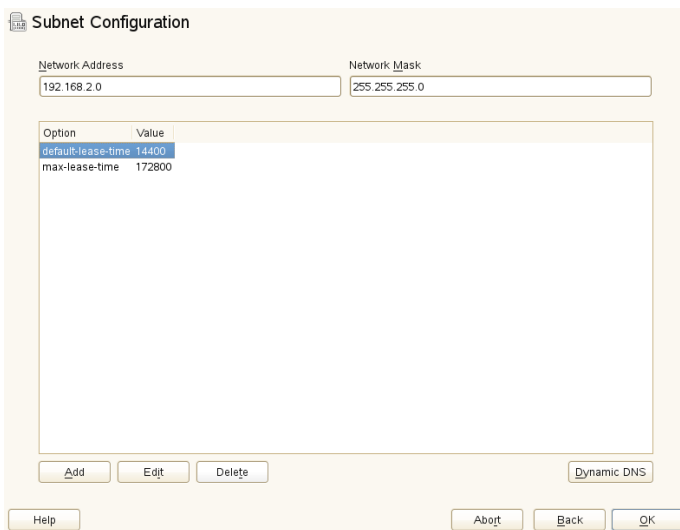
Figure 22.7 *DHCP Server: Selecting a Declaration Type*



Subnet Configuration

This dialog allows you specify a new subnet with its IP address and netmask. In the middle part of the dialog, modify the DHCP server start options for the selected subnet using *Add*, *Edit*, and *Delete*. To set up dynamic DNS for the subnet, select *Dynamic DNS*.

Figure 22.8 *DHCP Server: Configuring Subnets*



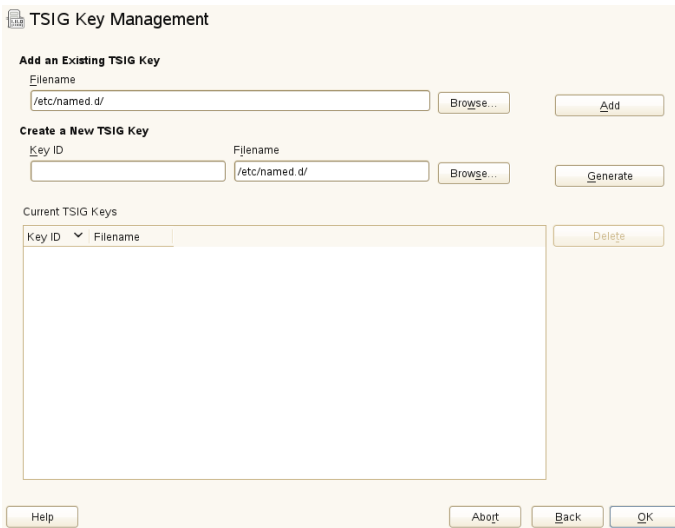
The dialog box is titled "Subnet Configuration". It contains two input fields at the top: "Network Address" with the value "192.168.2.0" and "Network Mask" with the value "255.255.255.0". Below these is a table with two columns, "Option" and "Value". The table contains two rows: "default-lease-time" with value "14400" and "max-lease-time" with value "172800". At the bottom of the table area are three buttons: "Add", "Edit", and "Delete". To the right of these buttons is a checkbox labeled "Dynamic DNS". At the very bottom of the dialog are four buttons: "Help", "Abort", "Back", and "OK".

Option	Value
default-lease-time	14400
max-lease-time	172800

TSIG Key Management

If you chose to configure dynamic DNS in the previous dialog, you can now configure the key management for a secure zone transfer. Selecting *OK* takes you to another dialog in which to configure the interface for dynamic DNS (see Figure 22.10, “DHCP Server: Interface Configuration for Dynamic DNS” (page 324)).

Figure 22.9 *DHCP Server: TSIG Configuration*



The image shows a 'TSIG Key Management' dialog box with a light beige background. At the top left is a small icon of a key. The title 'TSIG Key Management' is in bold. Below it, the section 'Add an Existing TSIG Key' contains a 'Filename' text field with the value '/etc/named.d/' and a 'Browse...' button to its right. An 'Add' button is at the bottom right of this section. The next section, 'Create a New TSIG Key', has a 'Key ID' text field, a 'Filename' text field with '/etc/named.d/' and a 'Browse...' button, and a 'Generate' button at the bottom right. Below these is a section titled 'Current TSIG Keys' which contains a table with two columns: 'Key ID' and 'Filename'. The table is currently empty. To the right of the table is a 'Delete' button. At the bottom of the dialog are four buttons: 'Help', 'Abort', 'Back', and 'OK'.

Dynamic DNS: Interface Configuration

You can now activate dynamic DNS for the subnet by selecting *Enable Dynamic DNS for This Subnet*. After doing so, use the drop-down list to choose the TSIG keys for forward and reverse zones, making sure that the keys are the same for the DNS and the DHCP server. With *Update Global Dynamic DNS Settings*, enable the automatic update and adjustment of the global DHCP server settings according to the dynamic DNS environment. Finally, define which forward and reverse zones should be updated per dynamic DNS, specifying the name of the primary name server for each of the two zones. If the name server runs on the same host as the DHCP server, you can leave these fields blank. Selecting *OK* returns to the subnet configuration dialog (see Figure 22.8, “DHCP Server: Configuring Subnets” (page 322)). Selecting *OK* again returns to the original expert configuration dialog.

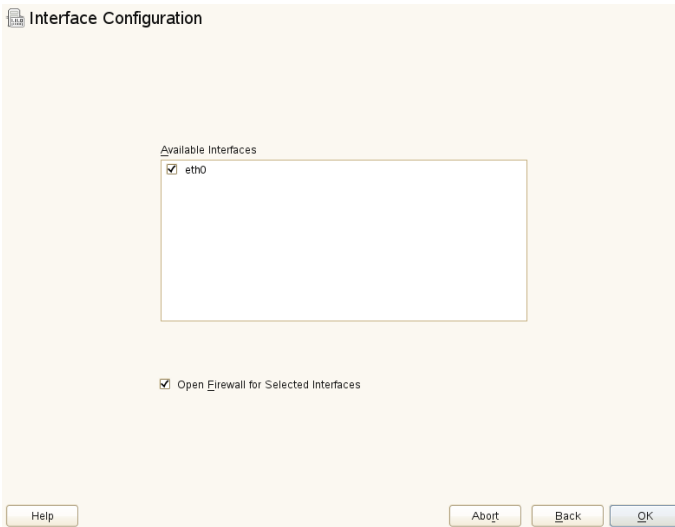
Figure 22.10 *DHCP Server: Interface Configuration for Dynamic DNS*

The screenshot shows a window titled "Interface Configuration". Inside, there is a checked checkbox labeled "Enable Dynamic DNS for This Subnet". Below it are two dropdown menus: "Forward Zone TSIG Key" and "Reverse Zone TSIG Key", both set to "1 a". There is an unchecked checkbox labeled "Update Global Dynamic DNS Settings". Below these are four text input fields arranged in two rows. The first row has "Zone" and "Primary DNS Server". The second row has "Reverse Zone" and "Primary DNS Server". At the bottom of the window are three buttons: "Help", "Abort", and "OK".

Network Interface Configuration

To define the interfaces where the DHCP server should listen and to adjust the firewall configuration, select *Advanced > Interface Configuration* from the expert configuration dialog. From the list of interfaces displayed, select one or more that should be attended by the DHCP server. If clients in all of the subnets need to be able to communicate with the server and the server host also runs a firewall, adjust the firewall accordingly. To do so, select *Adapt Firewall Settings*. YaST then adjusts the rules of SuSEfirewall2 to the new conditions (see Figure 22.11, “DHCP Server: Network Interface and Firewall” (page 325)), after which you can return to the original dialog by selecting *OK*.

Figure 22.11 *DHCP Server: Network Interface and Firewall*



After completing all configuration steps, close the dialog with *OK*. The server is now started with its new configuration.

22.2 DHCP Software Packages

Both the DHCP server and the DHCP clients are available for SUSE Linux Enterprise Server. The DHCP server available is `dhcpcd` (published by the Internet Systems Consortium). On the client side, choose between two different DHCP client programs: `dhcpc-client` (also from ISC) and the DHCP client daemon in the `dhcpcd` package.

SUSE Linux Enterprise Server installs `dhcpcd` by default. The program is very easy to handle and is launched automatically on each system boot to watch for a DHCP server. It does not need a configuration file to do its job and works out of the box in most standard setups. For more complex situations, use the ISC `dhcpc-client`, which is controlled by means of the configuration file `/etc/dhclient.conf`.

22.3 The DHCP Server `dhcpcd`

The core of any DHCP system is the dynamic host configuration protocol daemon. This server *leases* addresses and watches how they are used, according to the settings defined in the configuration file `/etc/dhcpd.conf`. By changing the parameters and values in this file, a system administrator can influence the program's behavior in numerous ways. Look at the basic sample `/etc/dhcpd.conf` file in Example 22.1, “The Configuration File `/etc/dhcpd.conf`” (page 326).

Example 22.1 *The Configuration File `/etc/dhcpd.conf`*

```
default-lease-time 600;          # 10 minutes
max-lease-time 7200;             # 2  hours

option domain-name "example.com";
option domain-name-servers 192.168.1.116;
option broadcast-address 192.168.2.255;
option routers 192.168.2.1;
option subnet-mask 255.255.255.0;

subnet 192.168.2.0 netmask 255.255.255.0
{
    range 192.168.2.10 192.168.2.20;
    range 192.168.2.100 192.168.2.200;
}
```

This simple configuration file should be sufficient to get the DHCP server to assign IP addresses in the network. Make sure that a semicolon is inserted at the end of each line, because otherwise `dhcpcd` is not started.

The sample file can be divided into three sections. The first one defines how many seconds an IP address is leased to a requesting client by default (`default-lease-time`) before it should apply for renewal. This section also includes a statement of the maximum period for which a machine may keep an IP address assigned by the DHCP server without applying for renewal (`max-lease-time`).

In the second part, some basic network parameters are defined on a global level:

- The line `option domain-name` defines the default domain of your network.
- With the entry `option domain-name-servers`, specify up to three values for the DNS servers used to resolve IP addresses into hostnames and vice versa.

Ideally, configure a name server on your machine or somewhere else in your network before setting up DHCP. That name server should also define a hostname for each dynamic address and vice versa. To learn how to configure your own name server, read Chapter 21, *The Domain Name System* (page 289).

- The line `option broadcast-address` defines the broadcast address the requesting client should use.
- With `option routers`, set where the server should send data packets that cannot be delivered to a host on the local network (according to the source and target host address and the subnet mask provided). In most cases, especially in smaller networks, this router is identical to the Internet gateway.
- With `option subnet-mask`, specify the netmask assigned to clients.

The last section of the file defines a network, including a subnet mask. To finish, specify the address range that the DHCP daemon should use to assign IP addresses to interested clients. In Example 22.1, “The Configuration File `/etc/dhcpd.conf`” (page 326), clients may be given any address between `192.168.2.10` and `192.168.2.20` as well as `192.168.2.100` and `192.168.2.200`.

After editing these few lines, you should be able to activate the DHCP daemon with the command `rcdhcpd start`. It will be ready for use immediately. Use the command `rcdhcpd check-syntax` to perform a brief syntax check. If you encounter any unexpected problems with your configuration (the server aborts with an error or does not return `done` on start), you should be able to find out what has gone wrong by looking for information either in the main system log `/var/log/messages` or on console 10 (Ctrl + Alt + F10).

On a default SUSE Linux Enterprise Server system, the DHCP daemon is started in a chroot environment for security reasons. The configuration files must be copied to the chroot environment so the daemon can find them. Normally, there is no need to worry about this because the command `rcdhcpd start` automatically copies the files.

22.3.1 Clients with Fixed IP Addresses

DHCP can also be used to assign a predefined, static address to a specific client. Addresses assigned explicitly always take priority over dynamic addresses from the pool. A static address never expires in the way a dynamic address would, for example, if

there were not enough addresses available and the server needed to redistribute them among clients.

To identify a client configured with a static address, `dhcpd` uses the hardware address (which is a globally unique, fixed numerical code consisting of six octet pairs) for the identification of all network devices (for example, `00:30:6E:08:EC:80`). If the respective lines, like the ones in Example 22.2, “Additions to the Configuration File” (page 328), are added to the configuration file of Example 22.1, “The Configuration File `/etc/dhcpd.conf`” (page 326), the DHCP daemon always assigns the same set of data to the corresponding client.

Example 22.2 *Additions to the Configuration File*

```
host jupiter {  
    hardware ethernet 00:30:6E:08:EC:80;  
    fixed-address 192.168.2.100;  
}
```

The name of the respective client (host *hostname*, here `jupiter`) is entered in the first line and the MAC address in the second line. On Linux hosts, find the MAC address with the command `ip link show` followed by the network device (for example, `eth0`). The output should contain something like

```
link/ether 00:30:6E:08:EC:80
```

In the preceding example, a client with a network card having the MAC address `00:30:6E:08:EC:80` is assigned the IP address `192.168.2.100` and the hostname `jupiter` automatically. The type of hardware to enter is `ethernet` in nearly all cases, although `token-ring`, which is often found on IBM systems, is also supported.

22.3.2 The SUSE Linux Enterprise Server Version

To improve security, the SUSE Linux Enterprise Server version of the ISC's DHCP server comes with the non-root/chroot patch by Ari Edelkind applied. This enables `dhcpd` to run with the user ID `nobody` and run in a chroot environment (`/var/lib/dhcp`). To make this possible, the configuration file `dhcpd.conf` must be located in `/var/lib/dhcp/etc`. The init script automatically copies the file to this directory when starting.

Control the server's behavior regarding this feature by means of entries in the file `/etc/sysconfig/dhcpd`. To run `dhcpd` without the `chroot` environment, set the variable `DHCPD_RUN_CHROOTED` in `/etc/sysconfig/dhcpd` to “no”.

To enable `dhcpd` to resolve hostnames even from within the `chroot` environment, some other configuration files must be copied as well:

- `/etc/localtime`
- `/etc/host.conf`
- `/etc/hosts`
- `/etc/resolv.conf`

These files are copied to `/var/lib/dhcp/etc/` when starting the init script. Take these copies into account for any changes that they require if they are dynamically modified by scripts like `/etc/ppp/ip-up`. However, there should be no need to worry about this if the configuration file only specifies IP addresses (instead of hostnames).

If your configuration includes additional files that should be copied into the `chroot` environment, set these under the variable `DHCPD_CONF_INCLUDE_FILES` in the file `/etc/sysconfig/dhcpd`. To ensure that the DHCP logging facility keeps working even after a restart of the `syslog-ng` daemon, there is an additional entry `SYSLOGD_ADDITIONAL_SOCKET_DHCP` in the file `/etc/sysconfig/syslog`.

22.4 For More Information

More information about DHCP is available at the Web site of the *Internet Systems Consortium* (<http://www.isc.org/products/DHCP/>). Information is also available in the `dhcpd`, `dhcpd.conf`, `dhcpd.leases`, and `dhcp-options` man pages.

Using NetworkManager

NetworkManager is the ideal solution for laptops and other portable computers. It supports state-of-the-art encryption types and standards for network connections, including connections to 802.1x protected networks. With NetworkManager, you do not need to worry about configuring network interfaces and switching between wired or wireless networks when you are moving. NetworkManager can automatically connect to known wireless networks or manage several network connections in parallel—the fastest connection is then used as default. Furthermore, you can manually switch between available networks and manage your network connection using an applet in the system tray.

Instead of only one connection being active, multiple connections may be active at once. This enables you to unplug your laptop from an Ethernet and remain connected via a wireless connection.

23.1 Use Cases for NetworkManager

NetworkManager provides a sophisticated and intuitive user interface which enables users to easily switch their network environment. However, NetworkManager is not a suitable solution in the following cases:

- Your computer provides network services for other computers in your network, for example, it is a DHCP or DNS server.
- Your computer is a Xen server or your system is a virtual system inside Xen.

- 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

On laptop computers, NetworkManager is enabled by default. However, it can be at any time enabled or disabled in the YaST Network Settings module.

- 1 Run YaST and go to *Network Devices > Network Settings*.
- 2 The *Network Settings* dialog opens. Go to the *Global Options* tab.
- 3 To configure and manage your network connections with NetworkManager, select *User Controlled with NetworkManager*.
- 4 Click *OK*.
- 5 After choosing the method for managing network connections, set up your network card using automatic configuration via DHCP or a static IP address or configure your modem (for dial-up connections, use *Network Devices > Modem*). To configure an internal or USB ISDN modem, select *Network Devices > ISDN*. To configure an internal or USB DSL modem, select *Network Devices > DSL*.

Find a detailed description of the network configuration with YaST in Section 18.4, “Configuring a Network Connection with YaST” (page 233) and Chapter 15, *Wireless LAN* (page 177).

After having enabled NetworkManager, configure your network connections with the NetworkManager as described in Section 23.3, “Configuring Network Connections” (page 333).

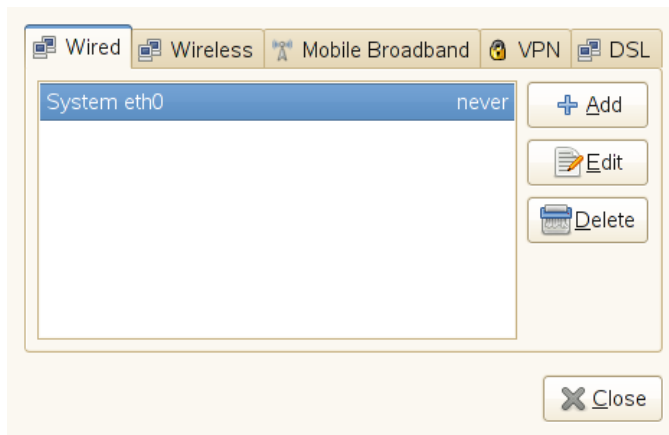
If you want to deactivate NetworkManager and control network the traditional way, choose the *Traditional Method with ifup* option in the *Network Setup Method* field.

23.3 Configuring Network Connections

After having enabled NetworkManager in YaST, configure your network connections with the NetworkManager front-ends available in KDE and GNOME. The network configuration dialogs for both front-ends are very similar. They show tabs for all types of network connections, such as wired, wireless, mobile broadband, DSL, and VPN connections. On each tab, you can add, edit or delete connections of that type. In the KDE configuration dialog, the appropriate tabs are only active if the connection type is available on your system (depending on hardware and software). By default, KNetworkManager also displays comprehensive tooltips for the input fields and options available on each tab.

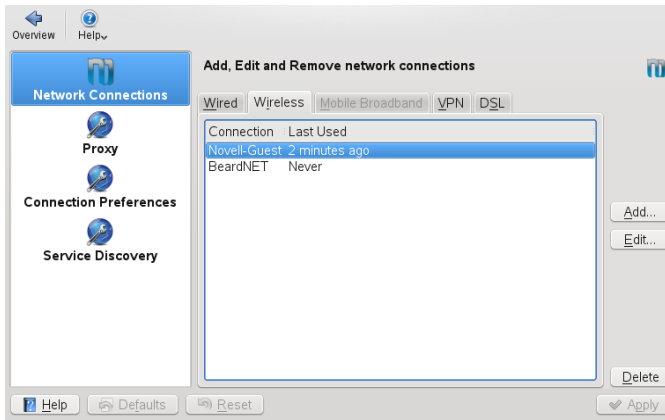
To open the network configuration dialog in GNOME, open the main menu and click the *Network* entry at the right. Conversely, press `Alt + F2` and enter `nm-connection-editor` (or in the GNOME Control Center select *System > Network Configurations*).

Figure 23.1 *GNOME Network Configuration Dialog*



If you use KDE, open the main menu and click *Configure Desktop*. In the *Personal Settings* then select *Network Settings* on the *General* tab to open the network configuration dialog.

Figure 23.2 KDE Network Configuration Dialog



Conversely, you can also start the configuration dialogs from the NetworkManager applet in the system tray. In KDE, left-click the icon and select *Manage Connections*. In GNOME, right-click the icon and select *Edit Connections*.

NOTE: Availability of Options

Depending on your system set-up, you may not be allowed to configure connections. In a secured environment, some options might be locked or require `root` permission. Ask your system administrator for details.

Procedure 23.1 *Adding or Editing Connections*

When configuring network connections with NetworkManager, you can also define `system` connections that can be shared by all users. In contrast to `user` connections, system connections are made available right after NetworkManager is started—before any users log in. For more details about both types of connections, refer to Section 23.7.1, “User and System Connections” (page 344).

Currently, the `system` connection option is not available in KNetworkManager. To set up system connections, you need to use YaST in this case.

NOTE: Hidden Networks

To connect to a “hidden” network (a network that does not broadcast its service) you have to know the Extended Service Set Identifier (ESSID) of the network because it cannot be detected automatically.

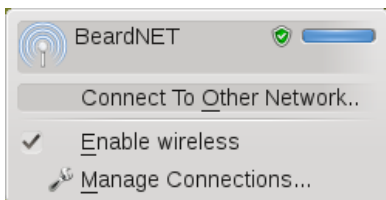
- 1 To add a new connection, click the tab for the connection type you want to use and click *Add*.
- 2 Enter a *Connection Name* and your connection details.
- 3 For a hidden network, enter the ESSID and the encryption parameters.
- 4 You can tie the connection to a certain device, if more than one physical device per connection type is available (for example, your machine is equipped with two ethernet cards or two wireless cards).

If you use KDE, do so by using the *Restrict to Interface* option. If you use GNOME, enter the *MAC address* of the device you want to tie the connection to and confirm your settings.

- 5 If you want NetworkManager to automatically use a certain connection, activate *Connect Automatically* for this connection.
- 6 To turn a connection into a `system connection` activate *Available to all users* (GNOME). To create and edit system connections, `root` permission is required.

After having confirmed your changes, the newly configured network connection appears in the list of available networks you get by left-clicking the NetworkManager applet.

Figure 23.3 *KNetworkManager—Configured and Available Connections*



23.4 Using KNetworkManager

The KDE front-end for NetworkManager is the KNetworkManager applet. If the network has been set up for NetworkManager control, the applet usually starts automatically with the desktop environment and is shown as an icon in the system tray.

If your system tray does not show any network connection icon, the applet is probably not started. Press **Alt + F2** and enter `knetworkmanager` to start it manually.

KNetworkManager only shows wireless networks that you have configured a connection for. It hides connections when you are out of range of a wireless network, or when the network cable is disconnected, thus always giving you a clear view of which connections may be used.

23.4.1 Connecting to Wired Networks

If your computer is connected to an existing network with a network cable, use KNetworkManager to choose the network connection.

- 1 Left-click the applet icon to show a menu with available networks. The connection currently being used is selected in the menu and marked as *Active*.
- 2 If you want to use a different configuration with the wired network, click *Manage Connections* and add another wired connection as described in Procedure 23.1, “Adding or Editing Connections” (page 334).
- 3 Click the KNetworkManager icon and select the newly configured connection to activate it.

23.4.2 Connecting to Wireless Networks

By default, KNetworkManager only shows wireless networks that you have configured a connection for—provided they are both available and visible. To connect to a wireless network for the first time, proceed as follows:

- 1 Left-click the applet icon and select *Create Network Connection*. KNetworkManager shows a list of available visible wireless networks, including details about signal strength and security.

- 2 To connect to a visible network, select the network from the list and click *Connect*. If the network is encrypted, a dialog opens. Choose the type of *Security* the network uses and enter the appropriate credentials.
- 3 To connect to a network that does not broadcast its service set identifier (ESSID) and therefore cannot be detected automatically, select *Connect to Other Network*.
- 4 In the dialog that opens, enter the ESSID and set encryption parameters, if necessary.
- 5 Confirm your changes and click *OK*. NetworkManager now activates the new connection.
- 6 To terminate a connection and to disable wireless networking, 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.

A wireless network that has been chosen explicitly will remain connected as long as possible. If a network cable is plugged in during that time, any connections that have been set to *Connect Automatically* will be connected, while the wireless connection remains up.

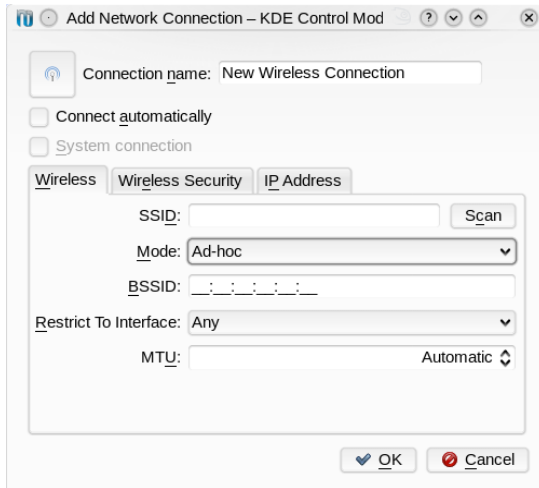
23.4.3 Configuring Your Wireless Card as an Access Point

If your wireless card supports access point mode, you can use NetworkManager for configuration.

NOTE: Availability of Options

Depending on your system set-up, you may not be allowed to configure connections. In a secured environment, some options might be locked or require `root` permission. Ask your system administrator for details.

- 1 Click the KNetworkManager applet and select *Create Network Connection > New Ad-Hoc Network*.
- 2 In the following configuration dialog, enter a name for your network in the *SSID* field.



- 3 Set the encryption on the *Wireless Security* tab.

IMPORTANT: Unprotected Wireless Networks Are a Security Risk

If you set *Security* to *None*, everybody can connect to your network, reuse your connectivity and intercept your network connection. To restrict access to your access point and to secure your connection, use encryption. You can choose between various WEP and WPA-based encryptions. If you are not sure which technology is best for you, read Section 15.3, “Authentication” (page 179).

- 4 On the *IP Address* tab, make sure the *Configure* option is set to *Shared* (which is the default option for ad-hoc networks).
- 5 Confirm your configuration with *OK*.

23.4.4 Customizing KNetworkManager

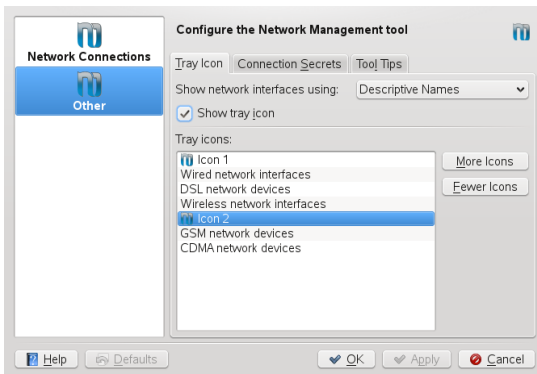
You can customize some aspects of KNetworkManager: the number of icons displayed in the system tray, which tool tips to show and how to store your password and credentials for network connections. For more information about the last aspect, refer to Section 23.7.2, “Storing Passwords and Credentials” (page 344).

To explore the options available, right-click the NetworkManager system tray icon and click *Other* on the left-hand side of the configuration dialog.

Procedure 23.2 *Configuring Multiple Tray Icons for KNetworkManager*

As KNetworkManager can keep multiple connections active at once, you might wish to be informed about the connection status for several connections at one glance. You can do so by using multiple NetworkManager icons in your system tray, each representing a different group of connection types (for example, one icon for wired connections, another icon for wireless connections).

- 1 In the configuration dialog, switch to the *Tray Icon* tab.
- 2 Click *More Icons*. A new icon entry appears in the list.
- 3 Select the network connection types you want to be represented by this icon and group them under the respective icon.



- 4 Confirm your changes.

Now the system tray shows multiple NetworkManager icons from which you then can access the connection types tied to that icon.

When configuring a network connection as described in Procedure 23.1, “Adding or Editing Connections” (page 334), KNetworkManager also allows you to customize the icon displayed for this connection. To change the icon, click the icon button next to *Connection Name* and in the following dialog, select the icon of your choice. After

confirming your changes, the new icon is displayed in the list of available connections you get by clicking the KNetworkManager icon in the system tray.

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 any network connection icon, 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 23.3 *Connecting to a Wireless Network*

- 1 To connect to a wireless network, left-click the applet icon and choose an entry from the list of available wireless networks.

- 2 If the network is encrypted, a dialog opens. It shows the type of encryption the network uses (*Wireless Security*) and holds a number of input fields according to the respective encryption and authentication settings. Enter the appropriate credentials.
- 3 To connect to a network that does not broadcast its service set identifier (ESSID) and therefore cannot be detected automatically, left-click the NetworkManager icon and choose *Connect to Hidden Wireless Network*.
- 4 In the dialog that opens, enter the ESSID in *Network Name* 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.

A wireless network that has been chosen explicitly will remain connected as long as possible. If a network cable is plugged in during that time, any connections that have been set to *Connect Automatically* will be connected, while the wireless connection remains up.

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.

NOTE: Availability of Options

Depending on your system set-up, you may not be allowed to configure connections. In a secured environment, some options might be locked or require `root` permission. Ask your system administrator for details.

- 1 Click the NetworkManager applet and select *Create New Wireless Network*.



New wireless network

Enter a name for the wireless network you wish to create.

Network Name:

Wireless Security:

Password:

- 2 Enter a *Network Name* and set the encryption to use with the *Wireless Security* drop-down list.

IMPORTANT: Unprotected Wireless Networks Are a Security Risk

If you set *Wireless Security* to `None`, everybody can connect to your network, reuse your connectivity and intercept your network connection. To restrict access to your access point and to secure your connection, use encryption. You can choose between various WEP and WPA-based encryptions. If you are not sure which technology is best for you, read Section 15.3, “Authentication” (page 179).

23.6 NetworkManager and VPN

NetworkManager supports several Virtual Private Network (VPN) technologies. For each technology, SUSE Linux Enterprise Server comes with a base package providing the generic support for NetworkManager. In addition to that, you also need to install the respective desktop-specific package for your applet.

NovellVPN

To use this VPN technology, install `NetworkManager-novellvpn` and `NetworkManager-novellvpn-kde4` or `NetworkManager-novellvpn-gnome`.

NovellVPN support for KDE is not available yet, but is currently being worked on.

OpenVPN

To use this VPN technology, install `NetworkManager-openvpn` and `NetworkManager-openvpn-kde4` or `NetworkManager-openvpn-gnome`.

vpnc (Cisco)

To use this VPN technology, install `NetworkManager-vpnc` and `NetworkManager-vpnc-kde4` or `NetworkManager-vpnc-gnome`.

PPTP (Point-to-Point Tunneling Protocol)

To use this VPN technology, install `NetworkManager-pptp` and `NetworkManager-pptp-kde4` or `NetworkManager-pptp-gnome`.

After you have installed the packages, configure your VPN connection as described in Section 23.3, “Configuring Network Connections” (page 333).

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.

If you switch from using a wireless connection to offline mode, NetworkManager blanks the ESSID. This ensures that the card is disconnected.

23.7.1 User and System Connections

NetworkManager knows two types of connections: `user` and `system` connections. User connections are connections that become available to NetworkManager when the first user logs in. Any required credentials are asked from the user and when the user logs out, the connections are disconnected and removed from NetworkManager. Connections that are defined as system connection can be shared by all users and are made available right after NetworkManager is started—before any users log in. In case of system connections, all credentials must be provided at the time the connection is created. Such system connections can be used to automatically connect to networks that require authorization. For information how to configure user or system connections with NetworkManager, refer to Section 23.3, “Configuring Network Connections” (page 333).

For KDE, system connections with NetworkManager are currently not supported (use YaST instead).

23.7.2 Storing Passwords and Credentials

If you do not want to re-enter your credentials each time you want to connect to an encrypted network, you can use the desktop-specific tools GNOME Keyring Manager or KWalletManager to store your credentials encrypted on the disk, secured by a master password.

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*).

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 ethernet cards), you can tie a connection to a certain device.

To do so in GNOME, first look up the MAC address of your device (use the *Connection Information* available from the applet, or use the output of command line tools like `nm-tool` or `ifconfig`). Then start the dialog for configuring network connections and choose the connection you want to modify. On the *Wired* or *Wireless* tab, enter the *MAC Address* of the device and confirm your changes.

If you use KDE, start the dialog for configuring network connections and choose the connection you want to modify. On the *Ethernet* or *Wireless* tab, use the *Restrict to Interface* option to select the network interface to which to tie the connection.

How to specify a certain access point in case multiple access points with the same ESSID are detected?

When multiple access points with different wireless bands (a/b/g/n) are available, the access point with the strongest signal is automatically chosen by default. To override this, use the *BSSID* field when configuring wireless connections.

The Basic Service Set Identifier (BSSID) uniquely identifies each Basic Service Set. In an infrastructure Basic Service Set, the BSSID is the MAC address of the wireless access point. In an independent (ad-hoc) Basic Service Set, the BSSID is a locally administered MAC address generated from a 46-bit random number.

Start the dialog for configuring network connections from the GNOME Control Center with *System > Network Configurations* or in KDE 4 from the *Personal Settings* with *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*. If you are using GNOME, switch to the *IPv4 Settings* tab and from the *Method* drop-down list, choose *Shared to other computers*. If you are using KDE, switch to the *IP Address* tab and from the *Configure* drop-down

list, choose *Shared*. That will enable IP traffic forwarding and run a DHCP server on the device. Confirm your changes in NetworkManager.

2. As the DHCP server uses port 67, make sure that it is not blocked by the firewall: On the machine sharing the connections, start YaST and select *Security and Users > Firewall*. Switch to the *Allowed Services* category. If *DCHP Server* is not already shown as *Allowed Service*, select *DCHP Server* from *Services to Allow* and click *Add*. Confirm your changes in YaST.

How to provide static DNS information with automatic (DHCP, PPP, VPN) addresses? In case a DHCP server provides invalid DNS information (and/or routes), you can override it. Start the dialog for configuring network connections 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*. If you are using GNOME, switch to the *IPv4 Settings* tab, and from the *Method* drop-down list, choose *Automatic (DHCP) addresses only*. If you are using KDE, switch to the *IP Address* tab, and from the *Configure* drop-down list, choose *Automatic (DHCP) addresses only*. Enter the DNS information in the *DNS Servers* and *Search Domains* fields. From the drop-down list at the bottom of the tab, select *Routes* if you want to *Ignore automatically obtained routes*. Confirm your changes.

How to make NetworkManager connect to password protected networks before a user logs in?

Define a `system` connection that can be used for such purposes. For more information, refer to Section 23.7, “NetworkManager and Security” (page 343).

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 Does Not Start

The GNOME and KDE NetworkManager applets start automatically if the network is set up for NetworkManager control. If the applet does not start, check if NetworkManager is enabled in YaST as described in Section 23.2, “Enabling NetworkManager” (page 332). 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 desktop applet is installed but is not running for some reason, start it manually with the command `nm-applet` (GNOME) or `knetworkmanager` (KDE).

NetworkManager Applet Does Not Include the VPN Option

Support for NetworkManager, applets, and VPN for NetworkManager is distributed in separate packages. If your NetworkManager applet does not include the VPN option, check if the packages with NetworkManager support for your VPN technology are installed. For more information, see Section 23.6, “NetworkManager and VPN” (page 342).

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:

NetworkManager Project Page

<http://projects.gnome.org/NetworkManager/>

KDE NetworkManager Front-End

<http://userbase.kde.org/NetworkManagement>

Package Documentation

Also check out the information in the following directories for the latest information about NetworkManager and the GNOME and KDE NetworkManager applets:

- `/usr/share/doc/packages/NetworkManager/`,
- `/usr/share/doc/packages/NetworkManager-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.

TIP: IBM System z: NetBIOS Support

IBM System z merely supports SMB over TCP/IP. NetBIOS support is not available on these systems.

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

The Samba client is a system that uses Samba services from a Samba server over the SMB protocol. All common operating systems, such as Mac OS X, Windows, and OS/2, support the SMB protocol. The TCP/IP protocol must be installed on all computers. Samba provides a client for the different UNIX flavors. For Linux, there is a kernel module for SMB that allows the integration of SMB resources on the Linux system level. You do not need to run any daemon for the Samba client.

Shares

SMB servers provide resources to the clients by means of shares. Shares are printers and directories with their subdirectories on the server. It is exported by means of a name and can be accessed by its name. The share name can be set to any name—it does not have to be the name of the export directory. A printer is also assigned a name. Clients can access the printer by its name.

DC

A domain controller (DC) is a server that handles accounts in domain. For data replication, additional domain controllers are available in one domain.

24.2 Starting and Stopping Samba

You can start or stop the Samba server automatically (during boot) or manually. Starting and stopping policy is a part of the YaST Samba server configuration described in Section 24.3.1, “Configuring a Samba Server with YaST” (page 351).

To stop or start running Samba services with YaST, use *System > System Services (Runlevel)* and check winbind, smb, and nmb. From a command line, stop services required for Samba with `rcsmb stop && rcnmb stop` and start them with `rcnmb start && rcsmb start`; rcsmb cares about winbind if needed.

24.3 Configuring a Samba Server

A Samba server in SUSE® Linux Enterprise Server can be configured in two different ways: with YaST or manually. Manual configuration offers a higher level of detail, but lacks the convenience of the YaST GUI.

24.3.1 Configuring a Samba Server with YaST

To configure a Samba server, start YaST and select *Network Services > Samba Server*.

Initial Samba Configuration

When starting the module for the first time, the *Samba Installation* dialog starts, prompting you to make just a few basic decisions concerning administration of the server. At the end of the configuration it prompts for the Samba administrator password (*Samba Root Password*). For later starts, the *Samba Server Configuration* dialog appears.

The *Samba Installation* dialog consists of two steps and optional detailed settings:

Workgroup or Domain Name

Select an existing name from *Workgroup or Domain Name* or enter a new one and click *Next*.

Samba Server Type

In the next step, specify whether your server should act as CD (PDC) and click *Next*.

Start-Up

Select whether you want to start Samba *During Boot* or *Manually* and click *OK*. Then in the final popup box, set the *Samba root Password*.

You can change all settings later in the *Samba Configuration* dialog with the *Start-Up*, *Shares*, *Identity*, *Trusted Domains*, and *LDAP Settings* tabs.

Advanced Samba Configuration

During the first start of the Samba server module the *Samba Configuration* dialog appears directly after the two initial steps described in Section “Initial Samba Configuration” (page 351). Use it to adjust your Samba server configuration.

After editing your configuration, click *OK* to save your settings.

Starting the Server

In the *Start Up* tab, configure the start of the Samba server. To start the service every time your system boots, select *During Boot*. To activate manual start, choose *Manually*. More information about starting a Samba server is provided in Section 24.2, “Starting and Stopping Samba” (page 351).

In this tab, you can also open ports in your firewall. To do so, select *Open Port in Firewall*. If you have multiple network interfaces, select the network interface for Samba services by clicking *Firewall Details*, selecting the interfaces, and clicking *OK*.

Shares

In the *Shares* tab, determine the Samba shares to activate. There are some predefined shares, like homes and printers. Use *Toggle Status* to switch between *Active* and *Inactive*. Click *Add* to add new shares and *Delete* to delete the selected share.

Allow Users to Share Their Directories enables members of the group in *Permitted Group* to share directories they own with other users. For example, `users` for a local scope or `DOMAIN\Users` for a domain scope. The user also must make sure that the file system permissions allow access. With *Maximum Number of Shares*, limit the total amount of shares that may be created. To permit access to user shares without authentication, enable *Allow Guest Access*.

Identity

In the *Identity* tab, you can determine the domain with which the host is associated (*Base Settings*) and whether to use an alternative hostname in the network (*NetBIOS Hostname*). It is also possible to use Microsoft Windows Internet Name Service (WINS) for name resolution. In this case, activate *Use WINS for Hostname Resolution* and decide whether to *Retrieve WINS server via DHCP*. To set expert global settings or set a user authentication source, for example LDAP instead of TDB database, click *Advanced Settings*.

Trusted Domains

To enable users from other domains to access your domain, make the appropriate settings in the *Trusted Domains* tab. To add a new domain, click *Add*. To remove the selected domain, click *Delete*.

LDAP Settings

In the tab *LDAP Settings*, you can determine the LDAP server to use for authentication. To test the connection to your LDAP server, click *Test Connection*. To set expert LDAP settings or use default values, click *Advanced Settings*.

For more information about LDAP configuration, see Chapter 4, *LDAP—A Directory Service* (↑*Security Guide*).

24.3.2 Web Administration with SWAT

An alternative tool for Samba server administration is SWAT (Samba Web Administration Tool). It provides a simple Web interface with which to configure the Samba server. To use SWAT, open <http://localhost:901> in a Web browser and log

in as user `root`. If you do not have a special Samba root account, use the system `root` account.

NOTE: Activating SWAT

After Samba server installation, SWAT is not activated. To activate it, open *Network Services > Network Services (xinetd)* in YaST, enable the network services configuration, select *swat* from the table, and click *Toggle Status (On or Off)*.

24.3.3 Configuring the Server Manually

If you intend to use Samba as a server, install `samba`. The main configuration file of Samba is `/etc/samba/smb.conf`. This file can be divided into two logical parts. The `[global]` section contains the central and global settings. The `[share]` sections contain the individual file and printer shares. By means of this approach, details regarding the shares can be set differently or globally in the `[global]` section, which enhances the structural transparency of the configuration file.

The global Section

The following parameters of the `[global]` section need some adjustment to match the requirements of your network setup so other machines can access your Samba server via SMB in a Windows environment.

```
workgroup = TUX-NET
```

This line assigns the Samba server to a workgroup. Replace `TUX-NET` with an appropriate workgroup of your networking environment. Your Samba server appears under its DNS name unless this name has been assigned to some other machine in the network. If the DNS name is not available, set the server name using `netbiosname=MYNAME`. For more details about this parameter, see the `smb.conf` man page.

```
os level = 20
```

This parameter triggers whether your Samba server tries to become LMB (local master browser) for its workgroup. With the Samba 3 release series, it is seldom necessary to override the default setting (20). Choose a very low value such as 2 to spare the existing Windows network from any disturbances caused by a miscon-

figured Samba server. More information about this important topic can be found in the Network Browsing chapter of the Samba 3 Howto; for more information on the Samba 3 Howto, see Section 24.7, “For More Information” (page 361).

If no other SMB server is present in your network (such as a Windows 2000 server) and you want the Samba server to keep a list of all systems present in the local environment, set the `os_level` to a higher value (for example, 65). Your Samba server is then chosen as LMB for your local network.

When changing this setting, consider carefully how this could affect an existing Windows network environment. First test the changes in an isolated network or at a noncritical time of day.

`wins support` and `wins server`

To integrate your Samba server into an existing Windows network with an active WINS server, enable the `wins server` option and set its value to the IP address of that WINS server.

If your Windows machines are connected to separate subnets and need to still be aware of each other, you need to set up a WINS server. To turn a Samba server into such a WINS server, set the option `wins support = Yes`. Make sure that only one Samba server of the network has this setting enabled. The options `wins server` and `wins support` must never be enabled at the same time in your `smb.conf` file.

Shares

The following examples illustrate how a CD-ROM drive and the user directories (homes) are made available to the SMB clients.

`[cdrom]`

To avoid having the CD-ROM drive accidentally made available, these lines are deactivated with comment marks (semicolons in this case). Remove the semicolons in the first column to share the CD-ROM drive with Samba.

Example 24.1 *A CD-ROM Share (deactivated)*

```
;[cdrom]
;comment = Linux CD-ROM
;path = /media/cdrom
;locking = No
```

[cdrom] and comment

The [cdrom] section entry is the name of the share that can be seen by all SMB clients on the network. An additional comment can be added to further describe the share.

```
path = /media/cdrom
```

path exports the directory /media/cdrom.

By means of a very restrictive default configuration, this kind of share is only made available to the users present on this system. If this share should be made available to everybody, add a line `guest ok = yes` to the configuration. This setting gives read permissions to anyone on the network. It is recommended to handle this parameter with great care. This applies even more to the use of this parameter in the [global] section.

[homes]

The [homes] share is of special importance here. If the user has a valid account and password for the Linux file server and his own home directory, he can be connected to it.

Example 24.2 [homes] Share

```
[homes]
comment = Home Directories
valid users = %S
browseable = No
read only = No
create mask = 0640
directory mask = 0750
```

[homes]

As long as there is no other share using the share name of the user connecting to the SMB server, a share is dynamically generated using the [homes] share directives. The resulting name of the share is the username.

```
valid users = %S
```

%S is replaced with the concrete name of the share as soon as a connection has been successfully established. For a [homes] share, this is always the username. As a consequence, access rights to a user's share are restricted exclusively to that user.

```
browseable = No
```

This setting makes the share invisible in the network environment.

```
read only = No
```

By default, Samba prohibits write access to any exported share by means of the `read only = Yes` parameter. To make a share writable, set the value `read only = No`, which is synonymous with `writable = Yes`.

```
create mask = 0640
```

Systems that are not based on MS Windows NT do not understand the concept of UNIX permissions, so they cannot assign permissions when creating a file. The parameter `create mask` defines the access permissions assigned to newly created files. This only applies to writable shares. In effect, this setting means the owner has read and write permissions and the members of the owner's primary group have read permissions. `valid users = %S` prevents read access even if the group has read permissions. For the group to have read or write access, deactivate the line `valid users = %S`.

Security Levels

To improve security, each share access can be protected with a password. SMB offers the following ways of checking permissions:

Share Level Security (`security = share`)

A password is firmly assigned to a share. Everyone who knows this password has access to that share.

User Level Security (`security = user`)

This variant introduces the concept of the user to SMB. Each user must register with the server with his or her own password. After registration, the server can grant access to individual exported shares dependent on usernames.

Server Level Security (`security = server`)

To its clients, Samba pretends to be working in user level mode. However, it passes all password queries to another user level mode server, which takes care of authentication. This setting requires the additional `password server` parameter.

ADS Level Security (`security = ADS`)

In this mode, Samba will act as a domain member in an Active Directory environment. To operate in this mode, the machine running Samba needs Kerberos installed and configured. You must join the machine using Samba to the ADS realm. This can be done using the YaST *Windows Domain Membership* module.

Domain Level Security (`security = domain`)

This mode will only work correctly if the machine has been joined into a Windows NT Domain. Samba will try to validate username and password by passing it to a Windows NT Primary or Backup Domain Controller. The same way as a Windows NT Server would do. It expects the encrypted passwords parameter to be set to `yes`.

The selection of share, user, server, or domain level security applies to the entire server. It is not possible to offer individual shares of a server configuration with share level security and others with user level security. However, you can run a separate Samba server for each configured IP address on a system.

More information about this subject can be found in the Samba 3 HOWTO. For multiple servers on one system, pay attention to the options `interfaces` and `bind interfaces only`.

24.4 Configuring Clients

Clients can only access the Samba server via TCP/IP. NetBEUI and NetBIOS via IPX cannot be used with Samba.

24.4.1 Configuring a Samba Client with YaST

Configure a Samba client to access resources (files or printers) on the Samba server. Enter the domain or workgroup in the dialog *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.

24.5 Samba as Login Server

In networks where predominantly Windows clients are found, it is often preferable that users may only register with a valid account and password. In a Windows-based network, this task is handled by a primary domain controller (PDC). You can use a Windows NT server configured as PDC, but this task can also be done with a Samba server. The entries that must be made in the `[global]` section of `smb.conf` are shown in Example 24.3, “Global Section in `smb.conf`” (page 359).

Example 24.3 *Global Section in `smb.conf`*

```
[global]
    workgroup = TUX-NET
    domain logons = Yes
    domain master = Yes
```

If encrypted passwords are used for verification purposes the Samba server must be able to handle these. The entry `encrypt passwords = yes` in the `[global]` section enables this (with Samba version 3, this is now the default). In addition, it is necessary to prepare user accounts and passwords in an encryption format that conforms with Windows. Do this with the command `smbpasswd -a name`. Create the domain account for the computers, required by the Windows domain concept, with the following commands:

```
useradd hostname\$$
smbpasswd -a -m hostname
```

With the `useradd` command, a dollar sign is added. The command `smbpasswd` inserts this automatically when the parameter `-m` is used. The commented configuration example (`/usr/share/doc/packages/samba/examples/smb.conf.SUSE`) contains settings that automate this task.

```
add machine script = /usr/sbin/useradd -g nogroup -c "NT Machine Account" \
-s /bin/false %m$
```

To make sure that Samba can execute this script correctly, choose a Samba user with the required administrator permissions and add it to the `ntadmin` group. Then all users belonging to this Linux group can be assigned Domain Admin status with the command:

```
net groupmap add ntgroup="Domain Admins" unixgroup=ntadmin
```

For more information about this topic, see Chapter 12 of the Samba 3 HOWTO, found in `/usr/share/doc/packages/samba/Samba3-HOWTO.pdf`.

24.6 Samba Server in the Network with Active Directory

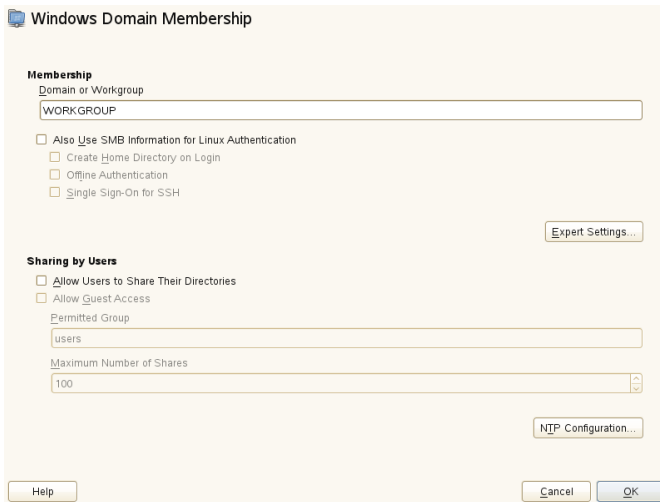
If you run Linux servers and Windows servers together, you can build two independent authentication systems and networks or connect servers to one network with one central authentication system. Because Samba can cooperate with an active directory domain, you can join your SUSE Linux Enterprise Server to Active Directory (AD).

Join an existing AD domain during installation or by later activating SMB user authentication with YaST in the installed system. Domain join during installation is covered in Section “User Authentication Method” (Chapter 6, *Installation with YaST, ↑Deployment Guide*).

To join an AD domain in a running system, proceed as follows:

- 1 Log in as `root` and start YaST.
- 2 Start *Network Services > Windows Domain Membership*.
- 3 Enter the domain to join at *Domain or Workgroup* in the *Windows Domain Membership* screen.

Figure 24.1 *Determining Windows Domain Membership*



- 4 Check *Also Use SMB Information for Linux Authentication* to use the SMB source for Linux authentication on your SUSE Linux Enterprise Server.
- 5 Click *OK* and confirm the domain join when prompted for it.
- 6 Provide the password for the Windows Administrator on the AD server and click *OK*.

Your server is now set up to pull in all authentication data from the Active Directory domain controller.

24.7 For More Information

Detailed Samba information is available in the digital documentation. Enter `apropos samba` at the command line to display some manual pages or just browse the `/usr/share/doc/packages/samba` directory if Samba documentation is installed for more online documentation and examples. Find a commented example configuration (`smb.conf.SUSE`) in the `examples` subdirectory.

The Samba 3 HOWTO provided by the Samba team includes a section about troubleshooting. In addition to that, Part V of the document provides a step-by-step guide to checking your configuration. You can find Samba 3 HOWTO in `/usr/share/doc/packages/samba/Samba3-HOWTO.pdf` after installing the package `samba-doc`.

Sharing File Systems with NFS

Distributing and sharing file systems over a network is a common task in corporate environments. The proven NFS system works together with NIS, the yellow pages protocol. For a more secure protocol that works together with LDAP and may also be kerberized, check NFSv4.

NFS with NIS makes a network transparent to the user. With NFS, it is possible to distribute arbitrary file systems over the network. With an appropriate setup, users always find themselves in the same environment regardless of the terminal they currently use.

Like NIS, NFS is a client/server system. However, a machine can be both—it can supply file systems over the network (export) and mount file systems from other hosts (import).

IMPORTANT: Need for DNS

In principle, all exports can be made using IP addresses only. To avoid timeouts, you should have a working DNS system. This is necessary at least for logging purposes, because the mountd daemon does reverse lookups.

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.

NFS server software is not part of the default installation. To install the NFS server software, start YaST and select *Software > Software Management*. Now choose *Filter*

> *Patterns* and select *Misc. Server* or use the *Search* option and search for *NFS Server*. Confirm the installation of the packages to finish the installation process.

25.2 Importing File Systems with YaST

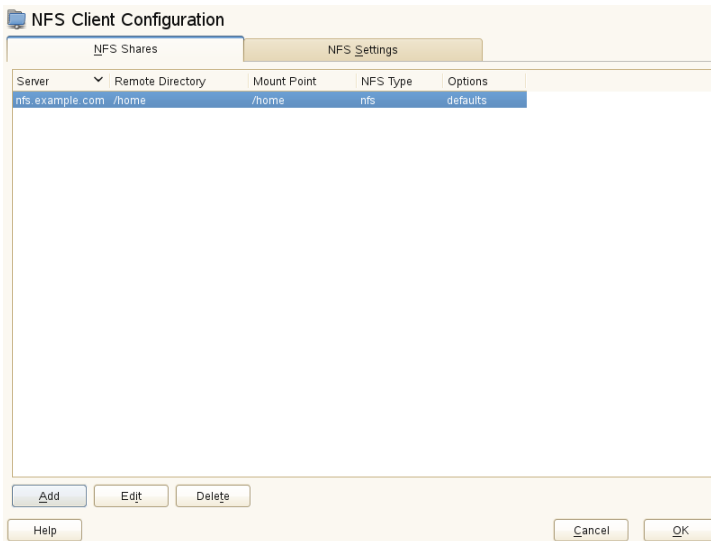
Authorized users can mount NFS directories from an NFS server into the local file tree using the YaST NFS client module. Click on *Add* and enter the hostname of the NFS server, the directory to import, and the mount point at which to mount this directory locally. The changes will take effect after *Finish* is clicked in the first dialog.

In the *NFS Settings* tab, enable *Open Port in Firewall* to allow access to the service from remote computers. The firewall status is displayed next to the check box. When using NFSv4, make sure that the checkbox *Enable NFSv4* is enabled, and that the *NFSv4 Domain Name* contains the same value as used by the NFSv4 server. The default domain is `localdomain`.

Click *OK* to save your changes. See Figure 25.1, “NFS Client Configuration with YaST” (page 365).

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

The prerequisite for importing file systems manually from an NFS server is a running RPC port mapper. Start it by entering `rcrpcbind start` as root. Then remote file systems can be mounted in the file system like local partitions using `mount`:

```
mount host:remote-path local-path
```

To import user directories from the `nfs.example.com` machine, for example, use:

```
mount nfs.example.com:/home /home
```

25.3.1 Using the Automount Service

The `autofs` daemon can be used to mount remote file systems automatically. Add the following entry in the your `/etc/auto.master` file:

```
/nfsmounts /etc/auto.nfs
```

Now the `/nfsmounts` directory acts as the root for all the NFS mounts on the client if the `auto.nfs` file is filled appropriately. The name `auto.nfs` is chosen for the sake of convenience—you can choose any name. In `auto.nfs` add entries for all the NFS mounts as follows:

```
localdata -fstype=nfs server1:/data
nfs4mount -fstype=nfs4 server2:/
```

Activate the settings with `rcautofs start` as root. In this example, `/nfsmounts/localdata`, the `/data` directory of `server1`, is mounted with NFS and `/nfsmounts/nfs4mount` from `server2` is mounted with NFSv4.

If the `/etc/auto.master` file is edited while the service `autofs` is running, the automounter must be restarted for the changes to take effect with `rcautofs restart`.

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. For these mounts, use `nfs4` instead of `nfs` in the third column and make sure that the remote file system is given as `/` after the `nfs.example.com:` in the first column. A sample line for an NFSv4 mount in `/etc/fstab` looks like this:

```
nfs.example.com:/ /local/pathv4 nfs4 rw,noauto 0 0
```

The `noauto` option prevents the file system from being mounted automatically at start up. If you want to mount the respective file system manually, it is possible to shorten the mount command specifying the mount point only:

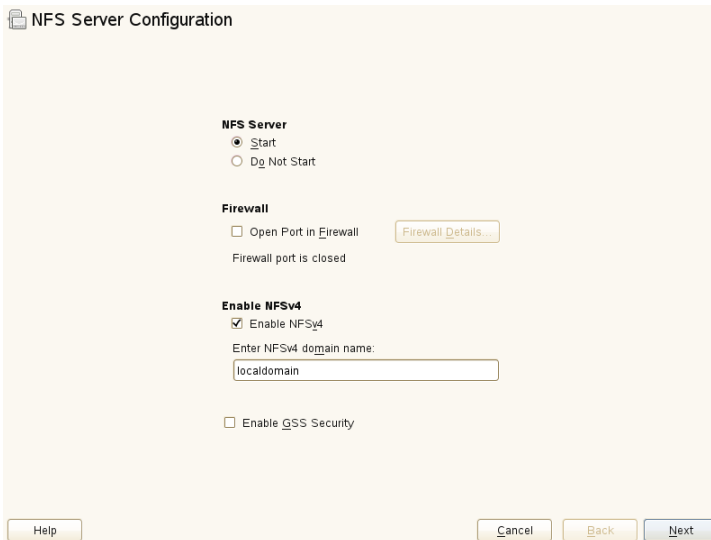
```
mount /local/path
```

Note, that if you do not enter the `noauto` option, the initialization scripts of the system will handle the mount of those file systems at start up.

25.4 Exporting File Systems with YaST

With YaST, turn a host in your network into an NFS server—a server that exports directories and files to all hosts granted access to it. This could be done to provide applications to all members of a group without installing them locally on each and every host. To install such a server, start YaST and select *Network Services > NFS Server*; see Figure 25.2, “NFS Server Configuration Tool” (page 367).

Figure 25.2 *NFS Server Configuration Tool*



The screenshot shows the "NFS Server Configuration" dialog box. It has a title bar with a document icon and the text "NFS Server Configuration". The main area contains three sections: "NFS Server" with radio buttons for "Start" (selected) and "Do Not Start"; "Firewall" with a checkbox for "Open Port in Firewall" (unchecked) and a "Firewall Details..." button, with the text "Firewall port is closed" below; and "Enable NFSv4" with a checked checkbox for "Enable NFSv4", a text field for "Enter NFSv4 domain name:" containing "localdomain", and an unchecked checkbox for "Enable GSS Security". At the bottom are four buttons: "Help", "Cancel", "Back", and "Next".

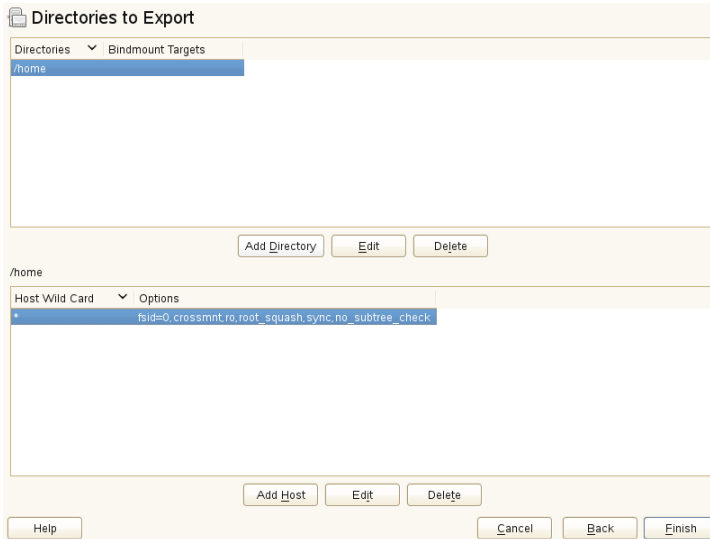
Then activate *Start* and enter the *NFSv4 Domain Name*.

Click *Enable GSS Security* if you need secure access to the server. A prerequisite for this is to have Kerberos installed on your domain and to have both the server and the clients kerberized. Click *Next*.

In the upper text field, enter the directories to export. Below, enter the hosts that should have access to them. This dialog is shown in Figure 25.3, “Configuring an NFS Server with YaST” (page 368). The figure shows the scenario where NFSv4 is enabled in the previous dialog. *Bindmount Targets* is shown in the right pane. For more details, click *Help*. In the lower half of the dialog, there are four options that can be set for each host: *single host*, *netgroups*, *wildcards*, and *IP networks*. For a more

thorough explanation of these options, refer to the `exports` man page. Click *Finish* to complete the configuration.

Figure 25.3 *Configuring an NFS Server with YaST*



IMPORTANT: Automatic Firewall Configuration

If a firewall is active on your system (SuSEfirewall2), YaST adapts its configuration for the NFS server by enabling the `nfs` service when *Open Ports in Firewall* is selected.

25.4.1 Exporting for NFSv4 Clients

Activate *Enable NFSv4* to support NFSv4 clients. Clients with NFSv3 can still access the server's exported directories if they are exported appropriately. This is explained in detail in Section 25.4.3, “Coexisting v3 and v4 Exports” (page 371).

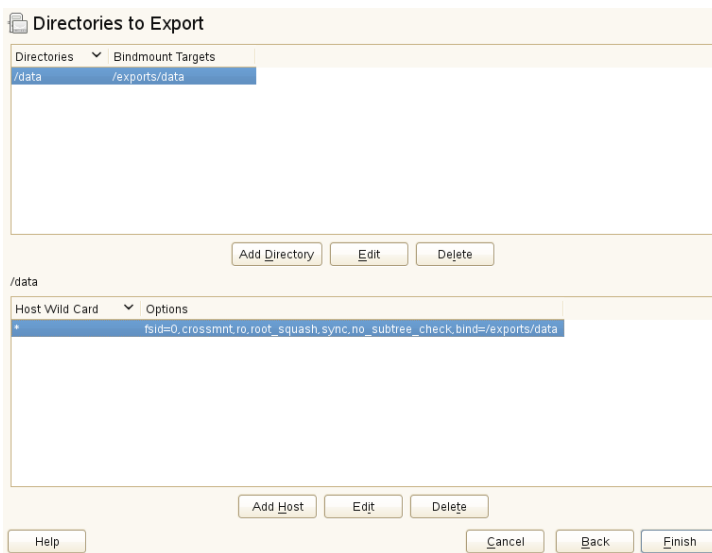
After activating NFSv4, enter an appropriate domain name. Make sure the name is the same as the one in the `/etc/ldapd.conf` file of any NFSv4 client that accesses this particular server. This parameter is for the `ldapd` service that is required for NFSv4 support (on both server and client). Leave it as `localdomain` (the default) if you do

not have special requirements. For more information, see the links in Section 25.7, “For More Information” (page 375).

Click *Next*. The dialog that follows has two sections. The upper half consists of two columns named *Directories* and *Bindmount Targets*. *Directories* is a directly editable column that lists the directories to export.

For a fixed set of clients, there are two types of directories that can be exported—directories that act as pseudo root file systems and those that are bound to some subdirectory of the pseudo file system. This pseudo file system acts as a base point under which all file systems exported for the same client set take their place. For a client or set of clients, only one directory on the server can be configured as pseudo root for export. For this client, export multiple directories by binding them to some existing subdirectory in the pseudo root.

Figure 25.4 *Exporting Directories with NFSv4*



In the lower half of the dialog, enter the client (wild card) and export options for a particular directory. After adding a directory in the upper half, another dialog for entering the client and option information pops up automatically. After that, to add a new client (client set), click *Add Host*.

In the small dialog that opens, enter the host wild card. There are four possible types of host wild cards that can be set for each host: a single host (name or IP address), netgroups, wild cards (such as `*` indicating all machines can access the server), and IP networks. Then, in *Options*, include `fsid=0` in the comma-separated list of options to configure the directory as pseudo root. If this directory needs to be bound to another directory under an already configured pseudo root, make sure that a target bind path is given in the option list with `bind=/target/path`.

For example, suppose that the directory `/exports` is chosen as the pseudo root directory for all the clients that can access the server. Then add this in the upper half and make sure that the options entered for this directory include `fsid=0`. If there is another directory, `/data`, that also needs to be NFSv4 exported, add this directory to the upper half. While entering options for this, make sure that `bind=/exports/data` is in the list and that `/exports/data` is an already existing subdirectory of `/exports`. Any change in the option `bind=/target/path`, whether addition, deletion, or change in value, is reflected in *Bindmount targets*. This column is not a directly editable column, but instead summarizes directories and their nature. After the information is complete, click *Finish* to complete the configuration .

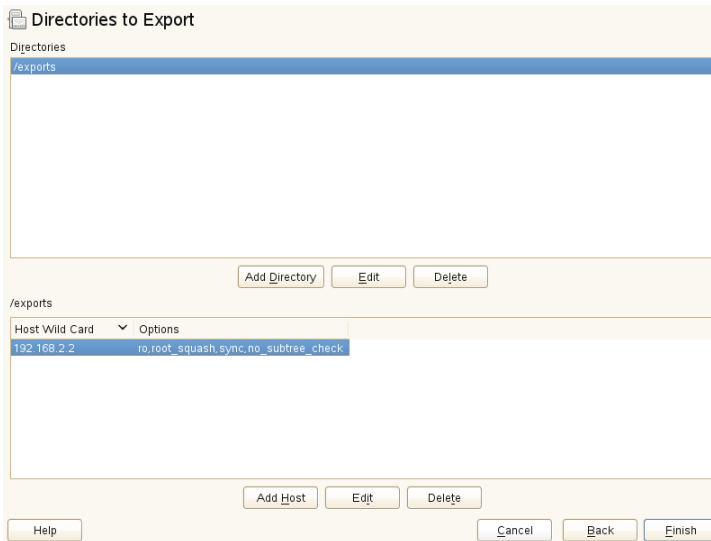
25.4.2 NFSv3 and NFSv2 Exports

Make sure that *Enable NFSv4* is not checked in the initial dialog before clicking *Next*.

The next dialog has two parts. In the upper text field, enter the directories to export. Below, enter the hosts that should have access to them. There are four types of host wild cards that can be set for each host: a single host (name or IP address), netgroups, wild cards (such as `*` indicating all machines can access the server), and IP networks.

This dialog is shown in Figure 25.5, “Exporting Directories with NFSv2 and v3” (page 371). Find a more thorough explanation of these options in `man exports`. Click *Finish* to complete the configuration.

Figure 25.5 *Exporting Directories with NFSv2 and v3*



25.4.3 Coexisting v3 and v4 Exports

Both, NFSv3 and NFSv4 exports can coexist on a server. After enabling the support for NFSv4 in the initial configuration dialog, those exports for which `fsid=0` and `bind=/target/path` are not included in the option list are considered v3 exports. Consider the example in Figure 25.3, “Configuring an NFS Server with YaST” (page 368). If you add another directory, such as `/data2`, using *Add Directory* then in the corresponding options list do not mention either `fsid=0` or `bind=/target/path`, this export acts as a v3 export.

IMPORTANT

Automatic Firewall Configuration

If `SuSEfirewall2` is active on your system, YaST adapts its configuration for the NFS server by enabling the `nfs` service when *Open Ports in Firewall* is selected.

25.5 Exporting File Systems Manually

The configuration files for the NFS export service are `/etc/exports` and `/etc/sysconfig/nfs`. In addition to these files, `/etc/idmapd.conf` is needed for the NFSv4 server configuration. To start or restart the services, run the command `rcnfsserver restart`. This also starts the `rpc.idmapd` if NFSv4 is configured in `/etc/sysconfig/nfs`. The NFS server depends on a running RPC portmapper. Therefore, also start or restart the portmapper service with `rcrpcbind restart`.

25.5.1 Exporting File Systems with NFSv4

NFSv4 is the latest version of NFS protocol available on SUSE Linux Enterprise Server. Configuring the directories for export with NFSv4 differs slightly from the previous NFS versions.

The `/etc/exports` File

This file contains a list of entries. Each entry indicates a directory that is shared and how it is shared. A typical entry in `/etc/exports` consists of:

```
/shared/directory host(option_list)
```

For example:

```
/export 192.168.1.2(rw,fsid=0,sync,crossmnt)
/export/data 192.168.1.2(rw,bind=/data,sync)
```

Here the IP address `192.168.1.2` is used to identify the allowed client. You can also use the name of the host, a wild card indicating a set of hosts (`*.abc.com`, `*`, etc.), or netgroups (`@my-hosts`).

The directory which specifies `fsid=0` is special in that it is the root of the filesystem that is exported, sometime referred to as the pseudo root filesystem. This directory must also have the `crossmnt` for correct operation with NFSv4. All other directories exported via NFSv4 must be mounted below this point. If you want to export a directory that is not within the exported root, it needs to be bound into the exported tree. This can be done using the `bind=` syntax.

In the example above, `/data` is not within `/export`, so we export `/export/data`, and specify that the `/data` directory should be bound to that name. The directory `/export/data` must exist and should normally be empty.

When clients mount from this server, they just mount `servername:/` rather than `servername:/export`. It is not necessary to mount `servername:/data`, because it will automatically appear beneath wherever `servername:/` was mounted.

/etc/sysconfig/nfs

This file contains a few parameters that determine NFSv4 server daemon behavior. Importantly, the parameter `NFSv4_SUPPORT` must be set to yes. This parameter determines whether the NFS server supports NFSv4 exports and clients.

/etc/idmapd.conf

Every user on a Linux machine has a name and ID. `idmapd` does the name-to-ID mapping for NFSv4 requests to the server and replies to the client. It must be running on both server and client for NFSv4, because NFSv4 uses only names for its communication.

Make sure that there is a uniform way in which usernames and IDs (uid) are assigned to users across machines that might probably be sharing file systems using NFS. This can be achieved by using NIS, LDAP, or any uniform domain authentication mechanism in your domain.

The parameter `Domain` must be set the same for both, client and server in this file. If you are not sure, leave the domain as `localdomain` in the server and client files. A sample configuration file looks like the following:

```
[General]

Verbosity = 0
Pipefs-Directory = /var/lib/nfs/rpc_pipefs
Domain = localdomain

[Mapping]
```

```
Nobody-User = nobody
Nobody-Group = nobody
```

For further reference, read the man page of `idmapd` and `idmapd.conf`; `man idmapd`, `man idmapd.conf`.

Starting and Stopping Services

After changing `/etc/exports` or `/etc/sysconfig/nfs`, start or restart the `nfs` server service with `rcnfsserver restart`. After changing `/etc/idmapd.conf`, reload the configuration file with the command `killall -HUP rpc.idmapd`.

If this service needs to start at boot time, run the command `chkconfig nfsserver on`.

25.5.2 Exporting File Systems with NFSv2 and NFSv3

This section is specific to NFSv3 and NFSv2 exports. Refer to Section 25.4.1, “Exporting for NFSv4 Clients” (page 368) for exporting with NFSv4.

Exporting file systems with NFS involves two configuration files: `/etc/exports` and `/etc/sysconfig/nfs`. A typical `/etc/exports` file entry is in the format:

```
/shared/directory host(list_of_options)
```

For example:

```
/export 192.168.1.2(rw,sync)
```

Here, the directory `/export` is shared with the host `192.168.1.2` with the option list `rw, sync`. This IP address can be replaced with a client name or set of clients using a wild card (such as `*.abc.com`) or even `netgroups`.

For a detailed explanation of all options and their meanings, refer to the man page of `exports` (`man exports`).

After changing `/etc/exports` or `/etc/sysconfig/nfs`, start or restart the NFS server using the command `rcnfsserver restart`.

25.6 NFS with Kerberos

To use Kerberos authentication for NFS, GSS security must be enabled. To do so, select *Enable GSS Security* in the initial YaST 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. They must access the same KDC (Key Distribution Center) server and share their `krb5.keytab` file (the default location on any machine is `/etc/krb5.keytab`).
- 2 Start the `gssd` service on the client with `rcgssd start`.
- 3 Start the `svcgssd` service on the server with `rcsvcgssd start`.

For further information about configuring kerberized NFS, refer to the links in Section 25.7, “For More Information” (page 375).

25.7 For More Information

As well as the man pages of `exports`, `nfs`, and `mount`, information about configuring an NFS server and client is available in `/usr/share/doc/packages/nfsidmap/README`. Online documentation can be found at the following Web documents:

- Find the detailed technical documentation online at SourceForge [<http://nfs.sourceforge.net/>].
- For instructions for setting up kerberized NFS, refer to NFS Version 4 Open Source Reference Implementation [<http://www.citi.umich.edu/projects/nfsv4/linux/krb5-setup.html>].
- If you have questions on NFSv4, refer to the Linux NFSv4 FAQ [<http://www.citi.umich.edu/projects/nfsv4/linux/faq/>].

File Synchronization

These days, 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 so that you 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 of keeping 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 applies to 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 the safe side, a versioning system can additionally be employed.

26.2.5 Selecting and Adding Files

In CVS, new directories and files must be added explicitly using the command `cv$ 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 `cv$ 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 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 changes 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 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/>.

The Apache HTTP Server

With a share of more than 50%, the Apache HTTP Server (Apache) is the world's most widely-used Web server according to the Survey from <http://www.netcraft.com/>. Apache, developed by the Apache Software Foundation (<http://www.apache.org/>), is available for most operating systems. SUSE® Linux Enterprise Server includes Apache version 2.2. In this chapter, learn how to install, configure and set up a Web server; how to use SSL, CGI, and additional modules; and how to troubleshoot Apache.

27.1 Quick Start

With the help of this section, quickly set up and start Apache. You must be `root` to install and configure Apache.

27.1.1 Requirements

Make sure that the following requirements are met before trying to set up the Apache Web server:

1. The machine's network is configured properly. For more information about this topic, refer to Chapter 18, *Basic Networking* (page 215).
2. The machine's exact system time is maintained by synchronizing with a time server. This is necessary because parts of the HTTP protocol depend on the correct

time. See Chapter 20, *Time Synchronization with NTP* (page 281) to learn more about this topic.

3. The latest security updates are installed. If in doubt, run a YaST Online Update.
4. The default Web server port (port 80) is opened in the firewall. For this, configure the SUSEFirewall2 to allow the service *HTTP Server* in the external zone. This can be done using YaST. Section “Configuring the Firewall with YaST” (Chapter 15, *Masquerading and Firewalls*, ↑*Security Guide*) gives details.

27.1.2 Installation

Apache on SUSE Linux Enterprise Server is not installed by default. To install it, start YaST and select *Software > Software Management*. Now choose *Filter > Patterns* and select *Web and LAMP Server* under *Server Functions*. Confirm the installation of the dependent packages to finish the installation process.

Apache is installed with a standard, predefined configuration that runs “out of the box”. The installation includes the multiprocessing module `apache2-prefork` as well the PHP5 module. Refer to Section 27.4, “Installing, Activating, and Configuring Modules” (page 407) for more information about modules.

27.1.3 Start

To start Apache and make sure that it is automatically started during boot, start YaST and select *System > System Services (Runlevel)*. Search for *apache2* and *Enable* the service. The Web server starts immediately. By saving your changes with *Finish*, the system is configured to automatically start Apache in runlevels 3 and 5 during boot. For more information about the runlevels in SUSE Linux Enterprise Server and a description of the YaST runlevel editor, refer to Section 7.2.3, “Configuring System Services (Runlevel) with YaST” (page 76).

To start Apache using the shell, run `rcapache2 start`. To make sure that Apache is automatically started during boot in runlevels 3 and 5, use `chkconfig -a apache2`.

If you do not receive error messages when starting Apache, the Web server should be running. Start a browser and open <http://localhost/>. You should see an Apache

test page stating “It works!”. If you do not see this page, refer to Section 27.8, “Troubleshooting” (page 425).

Now that the Web server is running, you can add your own documents, adjust the configuration according to your needs, or add functionality by installing modules.

27.2 Configuring Apache

Apache in SUSE Linux Enterprise Server can be configured in two different ways: with YaST or manually. Manual configuration offers a higher level of detail, but lacks the convenience of the YaST GUI.

IMPORTANT: Configuration Changes

Changes to most configuration values for Apache only take effect after Apache is restarted or reloaded. This happens automatically when using YaST and finishing the configuration with *Enabled* checked for the *HTTP Service*. Manual restart is described in Section 27.3, “Starting and Stopping Apache” (page 404). Most configuration changes only require a reload with `rcapache2 reload`.

27.2.1 Configuring Apache Manually

Configuring Apache manually involves editing the plain text configuration files as the user `root`.

Configuration Files

Apache configuration files can be found in two different locations:

- `/etc/sysconfig/apache2`
- `/etc/apache2/`

/etc/sysconfig/apache2

`/etc/sysconfig/apache2` controls some global settings of Apache, like modules to load, additional configuration files to include, flags with which the server should be started, and flags that should be added to the command line. Every configuration option in this file is extensively documented and therefore not mentioned here. For a general-purpose Web server, the settings in `/etc/sysconfig/apache2` should be sufficient for any configuration needs.

/etc/apache2/

`/etc/apache2/` hosts all configuration files for Apache. In the following, the purpose of each file is explained. Each file includes several configuration options (also referred to as *directives*). Every configuration option in these files is extensively documented and therefore not mentioned here.

The Apache configuration files are organized as follows:

```
/etc/apache2/
|
|  - charset.conv
|  - conf.d/
|    |
|    |  - *.conf
|
|  - default-server.conf
|  - errors.conf
|  - httpd.conf
|  - listen.conf
|  - magic
|  - mime.types
|  - mod_*.conf
|  - server-tuning.conf
|  - ssl.*
|  - ssl-global.conf
|  - sysconfig.d
|    |
|    |  - global.conf
|    |  - include.conf
|    |  - loadmodule.conf . .
|
|  - uid.conf
|  - vhosts.d
|    |  - *.conf
```

Apache Configuration Files in /etc/apache2/

`charset.conv`

Specifies which character sets to use for different languages. Do not edit.

`conf.d/*.conf`

Configuration files added by other modules. These configuration files can be included into your virtual host configuration where needed. See `vhhosts.d/vhost.template` for examples. By doing so, you can provide different module sets for different virtual hosts.

`default-server.conf`

Global configuration for all virtual hosts with reasonable defaults. Instead of changing the values, overwrite them with a virtual host configuration.

`errors.conf`

Defines how Apache responds to errors. To customize these messages for all virtual hosts, edit this file. Otherwise overwrite these directives in your virtual host configurations.

`httpd.conf`

The main Apache server configuration file. Avoid changing this file. It primarily contains include statements and global settings. Overwrite global settings in the pertinent configuration files listed here. Change host-specific settings (such as document root) in your virtual host configuration.

`listen.conf`

Binds Apache to specific IP addresses and ports. Name-based virtual hosting (see Section “Name-Based Virtual Hosts” (page 394) is also configured here.

`magic`

Data for the `mime_magic` module that helps Apache automatically determine the MIME type of an unknown file. Do not change.

`mime.types`

MIME types known by the system (this actually is a link to `/etc/mime.types`). Do not edit. If you need to add MIME types not listed here, add them to `mod_mime-defaults.conf`.

`mod_*.conf`

Configuration files for the modules that are installed by default. Refer to Section 27.4, “Installing, Activating, and Configuring Modules” (page 407) for details. Note that configuration files for optional modules reside in the directory `conf.d`.

`server-tuning.conf`

Contains configuration directives for the different MPMs (see Section 27.4.4, “Multiprocessing Modules” (page 411)) as well as general configuration options that control Apache's performance. Properly test your Web server when making changes here.

`ssl-global.conf` and `ssl.*`

Global SSL configuration and SSL certificate data. Refer to Section 27.6, “Setting Up a Secure Web Server with SSL” (page 417) for details.

`sysconfig.d/*.conf`

Configuration files automatically generated from `/etc/sysconfig/apache2`. Do not change any of these files—edit `/etc/sysconfig/apache2` instead. Put no other configuration files in this directory.

`uid.conf`

Specifies under which user and group ID Apache runs. Do not change.

`vhosts.d/*.conf`

Your virtual host configuration should go here. The directory contains template files for virtual hosts with and without SSL. Every file in this directory ending in `.conf` is automatically included in the Apache configuration. Refer to Section “Virtual Host Configuration” (page 392) for details.

Virtual Host Configuration

The term *virtual host* refers to Apache's ability to serve multiple URIs (universal resource identifiers) from the same physical machine. This means that several domains, such as `www.example.com` and `www.example.net`, are run by a single Web server on one physical machine.

It is common practice to use virtual hosts to save administrative effort (only a single Web server needs to be maintained) and hardware expenses (each domain does not require a dedicated server). Virtual hosts can be name based, IP based, or port based.

To list all existing virtual hosts, use the command `httpd2 -S`. This outputs a list showing the default server and all virtual hosts together with their IP addresses and listening ports. Furthermore, the list also contains an entry for each virtual host showing its location in the configuration files.

Virtual hosts can be configured via YaST (see Section “Virtual Hosts” (page 400)) or by manually editing a configuration file. By default, Apache in SUSE Linux Enterprise Server is prepared for one configuration file per virtual host in `/etc/apache2/vhosts.d/`. All files in this directory with the extension `.conf` are automatically included to the configuration. A basic template for a virtual host is provided in this directory (`vhost.template` or `vhost-ssl.template` for a virtual host with SSL support).

TIP: Always Create a Virtual Host Configuration

It is recommended to always create a virtual host configuration file, even if your Web server only hosts one domain. In doing so, you not only have the domain-specific configuration in one file, but you can always fall back to a working basic configuration by simply moving, deleting, or renaming the configuration file for the virtual host. For the same reason, you should also create separate configuration files for each virtual host.

When using name-based virtual hosts it is recommended to set up a default configuration that will be used when a domain name does not match a virtual host configuration. The default virtual host is the one whose configuration is loaded first. Since the order of the configuration files is determined by filename, start the filename of the default virtual host configuration with an “_”, e.g. `_default_vhost.conf`, to make sure it is loaded first.

The `<VirtualHost></VirtualHost>` block holds the information that applies to a particular domain. When Apache receives a client request for a defined virtual host, it uses the directives enclosed in this section. Almost all directives can be used in a virtual host context. See <http://httpd.apache.org/docs/2.2/mod/quickreference.html> for further information about Apache's configuration directives.

Name-Based Virtual Hosts

With name-based virtual hosts, more than one Web site is served per IP address. Apache uses the host field in the HTTP header sent by the client to connect the request to a matching `ServerName` entry of one of the virtual host declarations. If no matching `ServerName` is found, the first specified virtual host is used as a default.

The directive `NameVirtualHost` tells Apache on which IP address and, optionally, which port to listen for requests by clients containing the domain name in the HTTP header. This option is configured in the configuration file `/etc/apache2/listen.conf`.

The first argument can be a fully qualified domain name, but it is recommended to use the IP address. The second argument is the port and is optional. By default, port 80 is used and is configured via the `Listen` directive.

The wild card `*` can be used for both the IP address and the port number to receive requests on all interfaces. IPv6 addresses must be enclosed in square brackets.

Example 27.1 *Variations of Name-Based VirtualHost Entries*

```
# NameVirtualHost IP-address[:Port]
NameVirtualHost 192.168.3.100:80
NameVirtualHost 192.168.3.100
NameVirtualHost *:80
NameVirtualHost *
NameVirtualHost [2002:c0a8:364::]:80
```

The opening `VirtualHost` tag takes the IP address (or fully qualified domain name) previously declared with the `NameVirtualHost` as an argument in a name-based virtual host configuration. A port number previously declared with the `NameVirtualHost` directive is optional.

The wild card `*` is also allowed as a substitute for the IP address. This syntax is only valid in combination with the wild card usage in `NameVirtualHost *`. When using IPv6 addresses, the address must be included in square brackets.

Example 27.2 *Name-Based VirtualHost Directives*

```
<VirtualHost 192.168.3.100:80>
...
</VirtualHost>

<VirtualHost 192.168.3.100>
...
</VirtualHost>

<VirtualHost *:80>
...
</VirtualHost>

<VirtualHost *>
...
</VirtualHost>

<VirtualHost [2002:c0a8:364::]>
...
</VirtualHost>
```

IP-Based Virtual Hosts

This alternative virtual host configuration requires the setup of multiple IPs for a machine. One instance of Apache hosts several domains, each of which is assigned a different IP.

The physical server must have one IP address for each IP-based virtual host. If the machine does not have multiple network cards, virtual network interfaces (IP aliasing) can also be used.

The following example shows Apache running on a machine with the IP 192.168.3.100, hosting two domains on the additional IPs 192.168.3.101 and 192.168.3.102. A separate `VirtualHost` block is needed for every virtual server.

Example 27.3 *IP-Based VirtualHost Directives*

```
<VirtualHost 192.168.3.101>
...
</VirtualHost>

<VirtualHost 192.168.3.102>
...
</VirtualHost>
```

Here, `VirtualHost` directives are only specified for interfaces other than `192.168.3.100`. When a `Listen` directive is also configured for `192.168.3.100`, a separate IP-based virtual host must be created to answer HTTP requests to that interface—otherwise the directives found in the default server configuration (`/etc/apache2/default-server.conf`) are applied.

Basic Virtual Host Configuration

At least the following directives should be present in each virtual host configuration in order to set up a virtual host. See `/etc/apache2/vhosts.d/vhost.template` for more options.

`ServerName`

The fully qualified domain name under which the host should be addressed.

`DocumentRoot`

Path to the directory from which Apache should serve files for this host. For security reasons, access to the entire file system is forbidden by default, so you must explicitly unlock this directory within a `Directory` container.

`ServerAdmin`

E-mail address of the server administrator. This address is, for example, shown on error pages Apache creates.

`ErrorLog`

The error log file for this virtual host. Although it is not necessary to create separate error log files for each virtual host, it is common practice to do so, because it makes the debugging of errors much easier. `/var/log/apache2/` is the default directory for Apache's log files.

`CustomLog`

The access log file for this virtual host. Although it is not necessary to create separate access log files for each virtual host, it is common practice to do so, because it allows the separate analysis of access statistics for each host. `/var/log/apache2/` is the default directory for Apache's log files.

As mentioned above, access to the whole file system is forbidden by default for security reasons. Therefore, explicitly unlock the directories in which you have placed the files Apache should serve—for example the `DocumentRoot`:

```
<Directory "/srv/www/www.example.com/htdocs">
    Order allow,deny
    Allow from all
</Directory>
```

The complete configuration file looks like this:

Example 27.4 *Basic VirtualHost Configuration*

```
<VirtualHost 192.168.3.100>
    ServerName www.example.com
    DocumentRoot /srv/www/www.example.com/htdocs
    ServerAdmin webmaster@example.com
    ErrorLog /var/log/apache2/www.example.com_log
    CustomLog /var/log/apache2/www.example.com-access_log common
    <Directory "/srv/www/www.example.com/htdocs">
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

27.2.2 Configuring Apache with YaST

To configure your Web server with YaST, start YaST and select *Network Services > HTTP Server*. When starting the module for the first time, the HTTP Server Wizard starts, prompting you to make a few basic decisions concerning administration of the server. After having finished the wizard, the dialog in Section “HTTP Server Configuration” (page 402) starts each time you call the *HTTP Server* module.

HTTP Server Wizard

The HTTP Server Wizard consists of five steps. In the last step of the dialog, you are given the opportunity to enter the expert configuration mode to make even more specific settings.

Network Device Selection

Here, specify the network interfaces and ports Apache uses to listen for incoming requests. You can select any combination of existing network interfaces and their respective IP addresses. Ports from all three ranges (well-known ports, registered ports, and dynamic or private ports) that are not reserved by other services can be used. The default setting is to listen on all network interfaces (IP addresses) on port 80.

Check *Open Port in Firewall* to open the ports in the firewall that the Web server listens on. This is necessary to make the Web server available on the network, which can be a LAN, WAN, or the public Internet. Keeping the port closed is only useful in test situations where no external access to the Web server is necessary. If you have multiple network interfaces, click on *Firewall Details...* to specify on which interface(s) the port(s) should be opened.

Click *Next* to continue with configuration.

Modules

The *Modules* configuration option allows for the activation or deactivation of the script languages, the Web server should support. For the activation or deactivation of other modules, refer to Section “Server Modules” (page 403). Click *Next* to advance to the next dialog.

Default Host

This option pertains to the default Web server. As explained in Section “Virtual Host Configuration” (page 392), Apache can serve multiple virtual hosts from a single physical machine. The first declared virtual host in the configuration file is commonly referred to as the *default host*. Each virtual host inherits the default host's configuration.

To edit the host settings (also called *directives*), choose the appropriate entry in the table then click *Edit*. To add new directives, click *Add*. To delete a directive, select it and click *Delete*.

Figure 27.1 *HTTP Server Wizard: Default Host*

HTTP Server Wizard (3/5)--Default Host

Option	Value
Document Root	/srv/www/htdocs
Directory	/srv/www/htdocs/...
Alias	/icons/ /usr/share/apache2/icons/
Directory	/usr/share/apache2/icons/...
ScriptAlias	/cgi-bin/ /srv/www/cgi-bin/
Directory	/srv/www/cgi-bin/...
mod_userdir.c	
Include	/etc/apache2/conf.d/*.conf
Include	/etc/apache2/conf.d/apache2-manual.conf
Server Name	bourbaki
Server Administrator E-Mail	root@bourbaki

Add Edit Delete

Help Cancel Back Next

Here is list of the default settings of the server:

Document Root

Path to the directory from which Apache serves files for this host. `/srv/www/htdocs` is the default location.

Alias

With the help of `Alias` directives, URLs can be mapped to physical file system locations. This means that a certain path even outside the `Document Root` in the file system can be accessed via a URL aliasing that path.

The default SUSE Linux Enterprise Server `Alias /icons` points to `/usr/share/apache2/icons` for the Apache icons displayed in the directory index view.

ScriptAlias

Similar to the `Alias` directive, the `ScriptAlias` directive maps a URL to a file system location. The difference is that `ScriptAlias` designates the target directory as a CGI location, meaning that CGI scripts should be executed in that location.

Directory

With `Directory` settings, you can enclose a group of configuration options that will only apply to the specified directory.

Access and display options for the directories `/srv/www/htdocs`, `/usr/share/apache2/icons` and `/srv/www/cgi-bin` are configured here. It should not be necessary to change the defaults.

Include

With `include`, additional configuration files can be specified. Two `Include` directives are already preconfigured: `/etc/apache2/conf.d/` is the directory containing the configuration files that come with external modules. With this directive, all files in this directory ending in `.conf` are included. With the second directive, `/etc/apache2/conf.d/apache2-manual.conf`, the `apache2-manual` configuration file is included.

Server Name

This specifies the default URL used by clients to contact the Web server. Use a fully qualified domain name (FQDN) to reach the Web server at `http://FQDN/` or its IP address. You cannot choose an arbitrary name here—the server must be “known” under this name.

Server Administrator E-Mail

E-mail address of the server administrator. This address is, for example, shown on error pages Apache creates.

After finishing with the *Default Host* step, click *Next* to continue with the configuration.

Virtual Hosts

In this step, the wizard displays a list of already configured virtual hosts (see Section “Virtual Host Configuration” (page 392)). If you have not made manual changes prior to starting the YaST HTTP wizard, no virtual host is present.

To add a host, click *Add* to open a dialog in which to enter basic information about the host, such as *Server Name*, *Server Contents Root* (`DocumentRoot`), and the *Administrator E-Mail*. *Server Resolution* is used to determine how a host is identified (name based or IP based). Specify the name or IP address with *Change Virtual Host ID*

Clicking *Next* advances to the second part of the virtual host configuration dialog.

In part two of the virtual host configuration you can specify whether or not to enable CGI scripts and which directory to use for these scripts. It is also possible to enable SSL. If you do so, you must specify the path to the certificate as well. See Section 27.6.2, “Configuring Apache with SSL” (page 422) for details on SSL and certificates. With the *Directory Index* option, you can specify which file to display when the client requests a directory (by default, `index.html`). Add one or more filenames (space-separated) if you want to change this. With *Enable Public HTML*, the content of the users public directories (`~user/public_html/`) is made available on the server under `http://www.example.com/~user`.

IMPORTANT: Creating Virtual Hosts

It is not possible to add virtual hosts at will. If using name-based virtual hosts, each hostname must be resolved on the network. If using IP-based virtual hosts, you can assign only one host to each IP address available.

Summary

This is the final step of the wizard. Here, determine how and when the Apache server is started: when booting or manually. Also see a short summary of the configuration made so far. If you are satisfied with your settings, click *Finish* to complete configuration. If you want to change something, click *Back* until you have reached the desired dialog. Clicking *HTTP Server Expert Configuration* opens the dialog described in Section “HTTP Server Configuration” (page 402).

Figure 27.2 *HTTP Server Wizard: Summary*



HTTP Server Configuration

The *HTTP Server Configuration* dialog also lets you make even more adjustments to the configuration than the wizard (which only runs if you configure your Web server for the first time). It consists of four tabs described in the following. No configuration option you change here is effective immediately—you always must confirm your changes with *Finish* to make them effective. Clicking *Abort* leaves the configuration module and discards your changes.

Listen Ports and Addresses

In *HTTP Service*, select whether Apache should be running (*Enabled*) or stopped (*Disabled*). In *Listen on Ports*, *Add*, *Edit*, or *Delete* addresses and ports on which the server should be available. The default is to listen on all interfaces on port 80. You should always check *Open Port in Firewall*, because otherwise the Web server is not reachable from the outside. Keeping the port closed is only useful in test situations where no external access to the Web server is necessary. If you have multiple network interfaces, click on *Firewall Details...* to specify on which interface(s) the port(s) should be opened.

With *Log Files*, watch either the access log or the error log. This is useful if you want to test your configuration. The log file opens in a separate window from which you can also restart or reload the Web server (see Section 27.3, “Starting and Stopping Apache” (page 404) for details). These commands are effective immediately and their log messages are also displayed immediately.

Figure 27.3 *HTTP Server Configuration: Listen Ports and Addresses*

HTTP Server Configuration

Listen Ports and Addresses Server Modules Main Host Hosts

HTTP Service

☐ Disabled

☒ Enabled

Listen on Ports:

Network Address	Port
All Addresses	80

Add Edit Delete

☒ Open Port in Firewall Firewall Details...

Firewall port is open on all interfaces

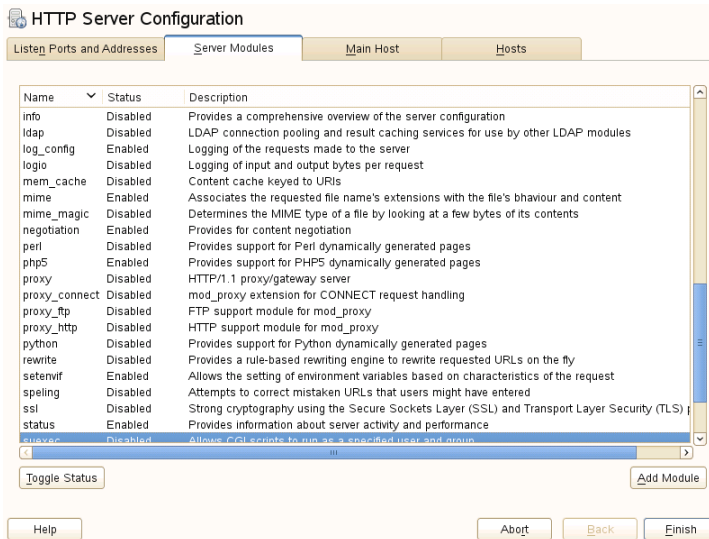
Log Files

Help Abort Back Finish

Server Modules

You can change the status (enabled or disabled) of Apache2 modules by clicking *Toggle Status*. Click *Add Module* to add a new module that is already installed but not yet listed. Learn more about modules in Section 27.4, “Installing, Activating, and Configuring Modules” (page 407).

Figure 27.4 *HTTP Server Configuration: Server Modules*



Main Host or Hosts

These dialogs are identical to the ones already described. Refer to Section “Default Host” (page 398) and Section “Virtual Hosts” (page 400).

27.3 Starting and Stopping Apache

If configured with YaST (see Section 27.2.2, “Configuring Apache with YaST” (page 397)), Apache is started at boot time in runlevels 3 and 5 and stopped in runlevels 0, 1, 2, and 6. You can change this behavior using YaST's runlevel editor or the command line tool `chkconfig`.

To start, stop, or manipulate Apache on a running system, use the init script `/usr/sbin/rcapache2` (refer to Section 7.2.2, “Init Scripts” (page 72) for a general information about init scripts.). The `rcapache2` command takes the following parameters:

`status`
Checks if Apache is started.

`start`

Starts Apache if it is not already running.

`startssl`

Starts Apache with SSL support if it is not already running. For more information about SSL support, refer to Section 27.6, “Setting Up a Secure Web Server with SSL” (page 417).

`stop`

Stops Apache by terminating the parent process.

`restart`

Stops and then restarts Apache. Starts the Web server if it was not running before.

`try-restart`

Stops then restarts Apache only if it is already running.

`reload or graceful`

Stops the Web server by advising all forked Apache processes to first finish their requests before shutting down. As each process dies, it is replaced by a newly started one, resulting in complete “restart” of Apache.

TIP

`rcapach2 reload` is the preferred method of restarting Apache in production environments, for example, to activate a change in the configuration, because it allows all clients to be served without causing connection break-offs.

`restart-graceful`

Starts a second Web server that immediately serves all incoming requests. The previous instance of the Web server continues to handle all existing requests for a defined period of time configured with `GracefulShutdownTimeout`.

`rcapach2 restart-graceful` is either useful when upgrading to a new version or when having changed configuration options that require a restart. Using this option ensures a minimum server downtime.

`GracefulShutdownTimeout` needs to be set, otherwise `restart-graceful` will result in a regular restart. If set to zero, the server will wait indefinitely until all remaining requests have been fully served.

A graceful restart can fail if the original Apache instance is not able to clear all necessary resources. In this case, the command will result in a graceful stop.

`stop-graceful`

Stops the Web server after a defined period of time configured with `GracefulShutdownTimeout` in order to ensure that existing requests can be finished.

`GracefulShutdownTimeout` needs to be set, otherwise `stop-graceful` will result in a regular restart. If set to zero, the server will wait indefinitely until all remaining requests have been fully served.

`configtest` or `extreme-configtest`

Checks the syntax of the configuration files without affecting a running Web server. Because this check is forced every time the server is started, reloaded, or restarted, it is usually not necessary to run the test explicitly (if a configuration error is found, the Web server is not started, reloaded, or restarted). The `extreme-configtest` options starts the Web server as user `nobody` and actually loads the configuration, so more errors can be detected. Note that although the configuration is loaded, it is not possible to test the SSL setup because the SSL certificates cannot be read by `nobody`.

`probe`

Probes for the necessity of a reload (checks whether the configuration has changed) and suggests the required arguments for the `rcapache2` command.

`server-status` and `full-server-status`

Dumps a short or full status screen, respectively. Requires either `lynx` or `w3m` installed as well as the module `mod_status` enabled. In addition to that, `status` must be added to `APACHE_SERVER_FLAGS` in the file `/etc/sysconfig/apache2`.

TIP: Additional Flags

If you specify additional flags to the `rcapache2`, these are passed through to the Web server.

27.4 Installing, Activating, and Configuring Modules

The Apache software is built in a modular fashion: all functionality except some core tasks is handled by modules. This has progressed so far that even HTTP is processed by a module (`http_core`).

Apache modules can be compiled into the Apache binary at build time or dynamically loaded at runtime. Refer to Section 27.4.2, “Activation and Deactivation” (page 408) for details of how to load modules dynamically.

Apache modules can be divided into four different categories:

Base Modules

Base modules are compiled into Apache by default. Apache in SUSE Linux Enterprise Server has only `mod_so` (needed to load other modules) and `http_core` compiled in. All others are available as shared objects: rather than being included in the server binary itself, they can be included at runtime.

Extension Modules

In general, modules labeled as extensions are included in the Apache software package, but are usually not compiled into the server statically. In SUSE Linux Enterprise Server, they are available as shared objects that can be loaded into Apache at runtime.

External Modules

Modules labeled external are not included in the official Apache distribution. SUSE Linux Enterprise Server provides several of them readily available for use.

Multiprocessing Modules

MPMs are responsible for accepting and handling requests to the Web server, representing the core of the Web server software.

27.4.1 Module Installation

If you have followed the default way of installing Apache (described in Section 27.1.2, “Installation” (page 388)), it is installed with all base and extension modules, the multiprocessing module Prefork MPM, and the external modules `mod_php5` and `mod_python`.

You can install additional external modules by starting YaST and choosing *Software > Software Management*. Now choose *Filter > Search* and search for *apache*. Among other packages, the results list contains all available external Apache modules.

27.4.2 Activation and Deactivation

Using YaST, you can activate or deactivate the script language modules (PHP5, Perl, and Python) with the module configuration described in Section “HTTP Server Wizard” (page 397). All other modules can be enabled or disabled as described in Section “Server Modules” (page 403).

If you prefer to activate or deactivate the modules manually, use the commands `a2enmod mod_foo` or `a2dismod mod_foo`, respectively. `a2enmod -l` outputs a list of all currently active modules.

IMPORTANT: Including Configuration Files for External Modules

If you have activated external modules manually, make sure to load their configuration files in all virtual host configurations. Configuration files for external modules are located under `/etc/apache2/conf.d/` and are not loaded by default. If you need the same modules on each virtual host, you can include `*.conf` from this directory. Otherwise include individual files. See `/etc/apache2/vhost.d/vhost.template` for examples.

27.4.3 Base and Extension Modules

All base and extension modules are described in detail in the Apache documentation. Only a brief description of the most important modules is available here. Refer to <http://httpd.apache.org/docs/2.2/mod/> to learn details about each module.

`mod_actions`

Provides methods to execute a script whenever a certain MIME type (such as `application/pdf`), a file with a specific extension (like `.rpm`), or a certain request method (such as `GET`) is requested. This module is enabled by default.

`mod_alias`

Provides `Alias` and `Redirect` directives with which you can map a URI to a specific directory (`Alias`) or redirect a requested URL to another location. This module is enabled by default.

`mod_auth*`

The authentication modules provide different authentication methods: basic authentication with `mod_auth_basic` or digest authentication with `mod_auth_digest`. Digest authentication in Apache 2.2 is considered experimental.

`mod_auth_basic` and `mod_auth_digest` must be combined with an authentication provider module, `mod_authn_*` (for example, `mod_authn_file` for text file–based authentication) and with an authorization module `mod_authz_*` (for example, `mod_authz_user` for user authorization).

More information about this topic is available in the “Authentication HOWTO” at <http://httpd.apache.org/docs/2.2/howto/auth.html>

`mod_autoindex`

Autoindex generates directory listings when no index file (for example, `index.html`) is present. The look and feel of these indexes is configurable. This module is enabled by default. However, directory listings are disabled by default via the `Options` directive—overwrite this setting in your virtual host configuration. The default configuration file for this module is located at `/etc/apache2/mod_autoindex-defaults.conf`.

`mod_cgi`

`mod_cgi` is needed to execute CGI scripts. This module is enabled by default.

`mod_deflate`

Using this module, Apache can be configured to compress given file types on the fly before delivering them.

`mod_dir`

`mod_dir` provides the `DirectoryIndex` directive with which you can configure which files are automatically delivered when a directory is requested (`index.html` by default). It also provides an automatic redirect to the correct URI when a directory request does not contain a trailing slash. This module is enabled by default.

`mod_env`

Controls the environment that is passed to CGI scripts or SSI pages. Environment variables can be set or unset or passed from the shell that invoked the `httpd` process. This module is enabled by default.

`mod_expires`

With `mod_expires`, you can control how often proxy and browser caches refresh your documents by sending an `Expires` header. This module is enabled by default.

`mod_include`

`mod_include` lets you use Server Side Includes (SSI), which provide a basic functionality to generate HTML pages dynamically. This module is enabled by default.

`mod_info`

Provides a comprehensive overview of the server configuration under `http://localhost/server-info/`. For security reasons, you should always limit access to this URL. By default only `localhost` is allowed to access this URL. `mod_info` is configured at `/etc/apache2/mod_info.conf`

`mod_log_config`

With this module, you can configure the look of the Apache log files. This module is enabled by default.

`mod_mime`

The `mime` module makes certain that a file is delivered with the correct MIME header based on the filename's extension (for example `text/html` for HTML documents). This module is enabled by default.

`mod_negotiation`

Necessary for content negotiation. See <http://httpd.apache.org/docs/2.2/content-negotiation.html> for more information. This module is enabled by default.

`mod_rewrite`

Provides the functionality of `mod_alias`, but offers more features and flexibility. With `mod_rewrite`, you can redirect URLs based on multiple rules, request headers, and more.

`mod_setenvif`

Sets environment variables based on details of the client's request, such as the browser string the client sends, or the client's IP address. This module is enabled by default.

`mod_speling`

`mod_speling` attempts to automatically correct typographical errors in URLs, such as capitalization errors.

`mod_ssl`

Enables encrypted connections between Web server and clients. See Section 27.6, “Setting Up a Secure Web Server with SSL” (page 417) for details. This module is enabled by default.

`mod_status`

Provides information on server activity and performance under `http://localhost/server-status/`. For security reasons, you should always limit access to this URL. By default, only `localhost` is allowed to access this URL. `mod_status` is configured at `/etc/apache2/mod_status.conf`

`mod_suexec`

`mod_suexec` lets you run CGI scripts under a different user and group. This module is enabled by default.

`mod_userdir`

Enables user-specific directories available under `~user/`. The `UserDir` directive must be specified in the configuration. This module is enabled by default.

27.4.4 Multiprocessing Modules

SUSE Linux Enterprise Server provides two different multiprocessing modules (MPMs) for use with Apache.

Prefork MPM

The prefork MPM implements a nonthreaded, preforking Web server. It makes the Web server behave similarly to Apache version 1.x in that it isolates each request and handles it by forking a separate child process. Thus problematic requests cannot affect others, avoiding a lockup of the Web server.

While providing stability with this process-based approach, the prefork MPM consumes more system resources than its counterpart, the worker MPM. The prefork MPM is considered the default MPM for Unix-based operating systems.

IMPORTANT: MPMs in This Document

This document assumes Apache is used with the prefork MPM.

Worker MPM

The worker MPM provides a multithreaded Web server. A thread is a “lighter” form of a process. The advantage of a thread over a process is its lower resource consumption. Instead of only forking child processes, the worker MPM serves requests by using threads with server processes. The preforked child processes are multithreaded. This approach makes Apache perform better by consuming fewer system resources than the prefork MPM.

One major disadvantage is the stability of the worker MPM: if a thread becomes corrupt, all threads of a process can be affected. In the worst case, this may result in a server crash. Especially when using the Common Gateway Interface (CGI) with Apache under heavy load, internal server errors might occur due to threads being unable to communicate with system resources. Another argument against using the worker MPM with Apache is that not all available Apache modules are thread-safe and thus cannot be used in conjunction with the worker MPM.

WARNING: Using PHP Modules with MPMs

Not all available PHP modules are thread-safe. Using the worker MPM with `mod_php` is strongly discouraged.

27.4.5 External Modules

Find a list of all external modules shipped with SUSE Linux Enterprise Server [here](#). Find the module's documentation in the listed directory.

`mod-apparmor`

Adds support to Apache to provide Novell AppArmor confinement to individual CGI scripts handled by modules like `mod_php5` and `mod_perl`.

Package Name: `apache2-mod_apparmor`
More Information: Part “Confining Privileges with Novell AppArmor” (*↑Security Guide*)

`mod_mono`

Using `mod_mono` allows you to run ASP.NET pages in your server.

Package Name: `apache2-mod_mono`
Configuration File: `/etc/apache2/conf.d/mod_mono.conf`

`mod_perl`

`mod_perl` enables you to run Perl scripts in an embedded interpreter. The persistent interpreter embedded in the server avoids the overhead of starting an external interpreter and the penalty of Perl start-up time.

Package Name: `apache2-mod_perl`
Configuration File: `/etc/apache2/conf.d/mod_perl.conf`
More Information: `/usr/share/doc/packages/apache2-mod_perl`

`mod_php5`

PHP is a server-side, cross-platform HTML embedded scripting language.

Package Name: `apache2-mod_php5`
Configuration File: `/etc/apache2/conf.d/php5.conf`
More Information: `/usr/share/doc/packages/apache2-mod_php5`

`mod_python`

`mod_python` allows embedding Python within the Apache HTTP server for a considerable boost in performance and added flexibility in designing Web-based applications.

Package Name: `apache2-mod_python`
More Information: `/usr/share/doc/packages/apache2-mod_python`

27.4.6 Compilation

Apache can be extended by advanced users by writing custom modules. To develop modules for Apache or compile third-party modules, the package `apache2-devel` is required along with the corresponding development tools. `apache2-devel` also

contains the `apxs2` tools, which are necessary for compiling additional modules for Apache.

`apxs2` enables the compilation and installation of modules from source code (including the required changes to the configuration files), which creates *dynamic shared objects* (DSOs) that can be loaded into Apache at runtime.

The `apxs2` binaries are located under `/usr/sbin`:

- `/usr/sbin/apxs2`—suitable for building an extension module that works with any MPM. The installation location is `/usr/lib/apache2`.
- `/usr/sbin/apxs2-prefork`—suitable for prefork MPM modules. The installation location is `/usr/lib/apache2-prefork`.
- `/usr/sbin/apxs2-worker`—suitable for worker MPM modules. The installation location is `/usr/lib/apache2-worker`.

Install and activate a module from source code with the commands `cd /path/to/module/source; apxs2 -cia mod_foo.c` (`-c` compiles the module, `-i` installs it, and `-a` activates it). Other options of `apxs2` are described in the `apxs2(1)` man page.

27.5 Getting CGI Scripts to Work

Apache's Common Gateway Interface (CGI) lets you create dynamic content with programs or scripts usually referred to as CGI scripts. CGI scripts can be written in any programming language. Usually, script languages such as Perl or PHP are used.

To enable Apache to deliver content created by CGI scripts, `mod_cgi` needs to be activated. `mod_alias` is also needed. Both modules are enabled by default. Refer to Section 27.4.2, “Activation and Deactivation” (page 408) for details on activating modules.

WARNING: CGI Security

Allowing the server to execute CGI scripts is a potential security hole. Refer to Section 27.7, “Avoiding Security Problems” (page 423) for additional information.

27.5.1 Apache Configuration

In SUSE Linux Enterprise Server, the execution of CGI scripts is only allowed in the directory `/srv/www/cgi-bin/`. This location is already configured to execute CGI scripts. If you have created a virtual host configuration (see Section “Virtual Host Configuration” (page 392)) and want to place your scripts in a host-specific directory, you must unlock and configure this directory.

Example 27.5 VirtualHost CGI Configuration

```
ScriptAlias /cgi-bin/ "/srv/www/www.example.com/cgi-bin/"❶

<Directory "/srv/www/www.example.com/cgi-bin/">
    Options +ExecCGI❷
    AddHandler cgi-script .cgi .pl❸
    Order allow,deny❹
    Allow from all
</Directory>
```

- ❶ Tells Apache to handle all files within this directory as CGI scripts.
- ❷ Enables CGI script execution
- ❸ Tells the server to treat files with the extensions `.pl` and `.cgi` as CGI scripts. Adjust according to your needs.
- ❹ The `Order` and `Allow` directives control the default access state and the order in which `Allow` and `Deny` directives are evaluated. In this case “allow” statements are evaluated before “deny” statements and universal access is enabled.

27.5.2 Running an Example Script

CGI programming differs from “regular” programming in that the CGI programs and scripts must be preceded by a MIME-Type header such as `Content-type: text/html`. This header is sent to the client, so it understands what kind of content it receives. Secondly, the script's output must be something the client, usually a Web browser, understands—HTML in most cases or plain text or images, for example.

A simple test script available under `/usr/share/doc/packages/apache2/test-cgi` is part of the Apache package. It outputs the content of some environment variables as plain text. Copy this script to either `/srv/www/cgi-bin/` or the script

directory of your virtual host (`/srv/www/www.example.com/cgi-bin/`) and name it `test.cgi`.

Files accessible by the Web server should be owned by to the user `root` (see Section 27.7, “Avoiding Security Problems” (page 423) for additional information). Because the Web server runs with a different user, the CGI scripts must be world-executable and world-readable. Change into the CGI directory and use the command `chmod 755 test.cgi` to apply the proper permissions.

Now call `http://localhost/cgi-bin/test.cgi` or `http://www.example.com/cgi-bin/test.cgi`. You should see the “CGI/1.0 test script report”.

27.5.3 Troubleshooting

If you do not see the output of the test program but an error message instead, check the following:

CGI Troubleshooting

- Have you reloaded the server after having changed the configuration? Check with `rcapache2 probe`.
- If you have configured your custom CGI directory, is it configured properly? If in doubt, try the script within the default CGI directory `/srv/www/cgi-bin/` and call it with `http://localhost/cgi-bin/test.cgi`.
- Are the file permissions correct? Change into the CGI directory and execute the `ls -l test.cgi`. Its output should start with

```
-rwxr-xr-x  1 root root
```
- Make sure that the script does not contain programming errors. If you have not changed `test.cgi`, this should not be the case, but if you are using your own programs, always make sure that they do not contain programming errors.

27.6 Setting Up a Secure Web Server with SSL

Whenever sensitive data, such as credit card information, is transferred between Web server and client, it is desirable to have a secure, encrypted connection with authentication. `mod_ssl` provides strong encryption using the secure sockets layer (SSL) and transport layer security (TLS) protocols for HTTP communication between a client and the Web server. Using SSL/TLS, a private connection between Web server and client is established. Data integrity is ensured and client and server are able to authenticate each other.

For this purpose, the server sends an SSL certificate that holds information proving the server's valid identity before any request to a URL is answered. In turn, this guarantees that the server is the uniquely correct end point for the communication. Additionally, the certificate generates an encrypted connection between client and server that can transport information without the risk of exposing sensitive, plain-text content.

`mod_ssl` does not implement the SSL/TLS protocols itself, but acts as an interface between Apache and an SSL library. In SUSE Linux Enterprise Server, the OpenSSL library is used. OpenSSL is automatically installed with Apache.

The most visible effect of using `mod_ssl` with Apache is that URLs are prefixed with `https://` instead of `http://`.

27.6.1 Creating an SSL Certificate

In order to use SSL/TLS with the Web server, you need to create an SSL certificate. This certificate is needed for the authorization between Web server and client, so that each party can clearly identify the other party. To ensure the integrity of the certificate, it must be signed by a party every user trusts.

There are three types of certificates you can create: a “dummy” certificate for testing purposes only, a self-signed certificate for a defined circle of users that trust you, and a certificate signed by an independent, publicly-known certificate authority (CA).

Creating a certificate is basically a two step process. First, a private key for the certificate authority is generated then the server certificate is signed with this key.

TIP: For More Information

To learn more about concepts and definitions of SSL/TSL, refer to http://httpd.apache.org/docs/2.2/ssl/ssl_intro.html.

Creating a “Dummy” Certificate

Generating a dummy certificate is simple. Just call the script `/usr/bin/gensslcert`. It creates or overwrites the files listed below. Make use of `gensslcert`'s optional switches to fine-tune the certificate. Call `/usr/bin/gensslcert -h` for more information.

- `/etc/apache2/ssl.crt/ca.crt`
- `/etc/apache2/ssl.crt/server.crt`
- `/etc/apache2/ssl.key/server.key`
- `/etc/apache2/ssl.csr/server.csr`
- `/root/.mkcert.cfg`

A copy of `ca.crt` is also placed at `/srv/www/htdocs/CA.crt` for download.

IMPORTANT

A dummy certificate should never be used on a production system. Only use it for testing purposes.

Creating a Self-Signed Certificate

If you are setting up a secure Web server for an Intranet or for a defined circle of users, it might be sufficient if you sign a certificate with your own certificate authority (CA).

Creating a self-signed certificate is an interactive nine-step process. Change into the directory `/usr/share/doc/packages/apache2` and run the following command:
`./mkcert.sh make --no-print-directory /usr/bin/openssl`

`/usr/sbin/` custom. Do not attempt to run this command from outside this directory. The program provides a series of prompts, some of which require user input.

Procedure 27.1 *Creating a Self-Signed Certificate with `mkcert.sh`*

- 1 Decide the signature algorithm used for certificates

Choose RSA (R, the default), because some older browsers have problems with DSA.

- 2 Generating RSA private key for CA (1024 bit)

No interaction needed.

- 3 Generating X.509 certificate signing request for CA

Create the CA's distinguished name here. This requires you to answer a few questions, such as country name or organization name. Enter valid data, because everything you enter here later shows up in the certificate. You do not need to answer every question. If one does not apply to you or you want to leave it blank, use “.”. Common name is the name of the CA itself—choose a significant name, such as *My company* CA.

IMPORTANT: Common Name of the CA

The common name of the CA must be different from the server's common name, so do not choose the fully qualified hostname in this step.

- 4 Generating X.509 certificate for CA signed by itself

Choose certificate version 3 (the default).

- 5 Generating RSA private key for SERVER (1024 bit)

No interaction needed.

- 6 Generating X.509 certificate signing request for SERVER

Create the distinguished name for the server key here. Questions are almost identical to the ones already answered for the CA's distinguished name. The data

entered here applies to the Web server and does not necessarily need to be identical to the CA's data (for example, if the server is located elsewhere).

IMPORTANT: Selecting a Common Name

The common name you enter here must be the fully qualified hostname of your secure server (for example, `www.example.com`). Otherwise the browser issues a warning that the certificate does not match the server when accessing the Web server.

7 `Generating X.509 certificate signed by own CA`

Choose certificate version 3 (the default).

8 `Encrypting RSA private key of CA with a pass phrase for security`

It is strongly recommended to encrypt the private key of the CA with a password, so choose Y and enter a password.

9 `Encrypting RSA private key of SERVER with a pass phrase for security`

Encrypting the server key with a password requires you to enter this password every time you start the Web server. This makes it difficult to automatically start the server on boot or to restart the Web server. Therefore, it is common sense to say N to this question. Keep in mind that your key is unprotected when not encrypted with a password and make sure that only authorized persons have access to the key.

IMPORTANT: Encrypting the Server Key

If you choose to encrypt the server key with a password, increase the value for `APACHE_TIMEOUT` in `/etc/sysconfig/apache2`. Otherwise you do not have enough time to enter the passphrase before the attempt to start the server is stopped unsuccessfully.

The script's result page presents a list of certificates and keys it has generated. Contrary to what the script outputs, the files have not been generated in the local directory `conf`, but to the correct locations under `/etc/apache2/`.

The last step is to copy the CA certificate file from `/etc/apache2/ssl.crt/ca.crt` to a location where your users can access it in order to incorporate it into the list of known and trusted CAs in their Web browsers. Otherwise a browser complains that the certificate was issued by an unknown authority. The certificate is valid for one year.

IMPORTANT: Self-Signed Certificates

Only use a self-signed certificate on a Web server that is accessed by people who know and trust you as a certificate authority. It is not recommended to use such a certificate on a public shop, for example.

Getting an Officially Signed Certificate

There are a number of official certificate authorities that sign your certificates. The certificate is signed by a trustworthy third party, so can be fully trusted. Publicly operating secure Web servers usually have got an officially signed certificate.

The best-known official CAs are Thawte (<http://www.thawte.com/>) or Verisign (<http://www.verisign.com>). These and other CAs are already compiled into all browsers, so certificates signed by these certificate authorities are automatically accepted by the browser.

When requesting an officially signed certificate, you do not send a certificate to the CA. Instead, issue a Certificate Signing Request (CSR). To create a CSR, call the script `/usr/share/ssl/misc/CA.sh -newreq`.

First the script asks for a password with which the CSR should be encrypted. Then you are asked to enter a distinguished name. This requires you to answer a few questions, such as country name or organization name. Enter valid data—everything you enter here later shows up in the certificate and is checked. You do not need to answer every question. If one does not apply to you or you want to leave it blank, use “.”. Common name is the name of the CA itself—choose a significant name, such as *My company* CA. Last, a challenge password and an alternative company name must be entered.

Find the CSR in the directory from which you called the script. The file is named `newreq.pem`.

27.6.2 Configuring Apache with SSL

The default port for SSL and TLS requests on the Web server side is 443. There is no conflict between a “regular” Apache listening on port 80 and an SSL/TLS-enabled Apache listening on port 443. In fact, HTTP and HTTPS can be run with the same Apache instance. Usually separate virtual hosts are used to dispatch requests to port 80 and port 443 to separate virtual servers.

IMPORTANT: Firewall Configuration

Do not forget to open the firewall for SSL-enabled Apache on port 443. This can be done with YaST as described in Section “Configuring the Firewall with YaST” (Chapter 15, *Masquerading and Firewalls*, ↑*Security Guide*).

The SSL module is enabled by default in the global server configuration. In case it has been disabled on your host, activate it with the following command: `a2enmod ssl`. To finally enable SSL, the server needs to be started with the flag “SSL”. To do so, call `a2enflag SSL`. If you have chosen to encrypt your server certificate with a password, you should also increase the value for `APACHE_TIMEOUT` in `/etc/sysconfig/apache2`, so you have enough time to enter the passphrase when Apache starts. Restart the server to make these changes active. A reload is not sufficient.

The virtual host configuration directory contains a template `/etc/apache2/vhosts.d/vhost-ssl.template` with SSL-specific directives that are extensively documented. Refer to Section “Virtual Host Configuration” (page 392) for the general virtual host configuration.

To get started, copy the template to `/etc/apache2/vhosts.d/mySSL-host.conf` and edit it. Adjusting the values for the following directives should be sufficient:

- `DocumentRoot`
- `ServerName`
- `ServerAdmin`
- `ErrorLog`
- `TransferLog`

IMPORTANT: Name-Based Virtual Hosts and SSL

It is not possible to run multiple SSL-enabled virtual hosts on a server with only one IP address. Users connecting to such a setup receive a warning message stating that the certificate does not match the server name every time they visit the URL. A separate IP address or port is necessary for every SSL-enabled domain to achieve communication based on a valid SSL certificate.

27.7 Avoiding Security Problems

A Web server exposed to the public Internet requires an ongoing administrative effort. It is inevitable that security issues appear, both related to the software and to accidental misconfiguration. Here are some tips for how to deal with them.

27.7.1 Up-to-Date Software

If there are vulnerabilities found in the Apache software, a security advisory will be issued by SUSE. It contains instructions for fixing the vulnerabilities, which in turn should be applied as soon as possible. The SUSE security announcements are available from the following locations:

- **Web Page** <http://www.novell.com/linux/security/securitysupport.html>
- **Mailing List** <http://en.opensuse.org/Communicate#Mailinglists>
- **RSS Feed** http://www.novell.com/linux/security/suse_security.xml

27.7.2 DocumentRoot Permissions

By default in SUSE Linux Enterprise Server, the `DocumentRoot` directory `/srv/www/htdocs` and the CGI directory `/srv/www/cgi-bin` belong to the user and group `root`. You should not change these permissions. If the directories were writable

for all, any user could place files into them. These files might then be executed by Apache with the permissions of `wwwrun`, which may give the user unintended access to file system resources. Use subdirectories of `/srv/www` to place the `DocumentRoot` and CGI directories for your virtual hosts and make sure that directories and files belong to user and group `root`.

27.7.3 File System Access

By default, access to the whole file system is denied in `/etc/apache2/httpd.conf`. You should never overwrite these directives, but specifically enable access to all directories Apache should be able to read (see Section “Basic Virtual Host Configuration” (page 396) for details). In doing so, ensure that no critical files, such as password or system configuration files, can be read from the outside.

27.7.4 CGI Scripts

Interactive scripts in Perl, PHP, SSI, or any other programming language can essentially run arbitrary commands and therefore present a general security issue. Scripts that will be executed from the server should only be installed from sources the server administrator trusts—allowing users to run their own scripts is generally not a good idea. It is also recommended to do security audits for all scripts.

To make the administration of scripts as easy as possible, it is common practice to limit the execution of CGI scripts to specific directories instead of globally allowing them. The directives `ScriptAlias` and `Option ExecCGI` are used for configuration. The SUSE Linux Enterprise Server default configuration does not allow execution of CGI scripts from everywhere.

All CGI scripts run as the same user, so different scripts can potentially conflict with each other. The module `suEXEC` lets you run CGI scripts under a different user and group.

27.7.5 User Directories

When enabling user directories (with `mod_userdir` or `mod_rewrite`) you should strongly consider not allowing `.htaccess` files, which would allow users to overwrite

security settings. At least you should limit the user's engagement by using the directive `AllowOverride`. In SUSE Linux Enterprise Server, `.htaccess` files are enabled by default, but the user is not allowed to overwrite any `Option` directives when using `mod_userdir` (see the `/etc/apache2/mod_userdir.conf` configuration file).

27.8 Troubleshooting

If Apache does not start, the Web page is not accessible, or users cannot connect to the Web server, it is important to find the cause of the problem. Here are some typical places to look for error explanations and important things to check.

First, `rcapache2` (described in Section 27.3, “Starting and Stopping Apache” (page 404)) is verbose about errors, so can be quite helpful if it is actually used for operating Apache. Sometimes it is tempting to use the binary `/usr/sbin/httpd2` for starting or stopping the Web server. Avoid doing this and use the `rcapache2` script instead. `rcapache2` even provides tips and hints for solving configuration errors.

Second, the importance of log files cannot be overemphasized. In case of both fatal and nonfatal errors, the Apache log files, mainly the error log file, are the places to look for causes. Additionally, you can control the verbosity of the logged messages with the `LogLevel` directive if more detail is needed in the log files. By default, the error log file is located at `/var/log/apache2/error_log`.

TIP: A Simple Test

Watch the Apache log messages with the command `tail -F /var/log/apache2/my_error_log`. Then run `rcapache2 restart`. Now, try to connect with a browser and check the output.

A common mistake is to not open the ports for Apache in the firewall configuration of the server. If you configure Apache with YaST, there is a separate option available to take care of this specific issue (see Section 27.2.2, “Configuring Apache with YaST” (page 397)). If you are configuring Apache manually, open firewall ports for HTTP and HTTPS via YaST's firewall module.

If the error cannot be tracked down with the help of any these, check the online Apache bug database at http://httpd.apache.org/bug_report.html. Additionally,

the Apache user community can be reached via a mailing list available at <http://httpd.apache.org/userslist.html>. A recommended newsgroup is comp.infosystems.www.servers.unix.

27.9 For More Information

The package `apache2-doc` contains the complete Apache manual in various localizations for local installation and reference. It is not installed by default—the quickest way to install it is to use the command `zypper in apache2-doc`. Once installed, the Apache manual is available at <http://localhost/manual/>. You may also access it on the Web at <http://httpd.apache.org/docs-2.2/>. SUSE-specific configuration hints are available in the directory `/usr/share/doc/packages/apache2/README.*`.

27.9.1 Apache 2.2

For a list of new features in Apache 2.2, refer to http://httpd.apache.org/docs/2.2/new_features_2_2.html. Information about upgrading from version 2.0 to 2.2 is available at <http://httpd.apache.org/docs-2.2/upgrading.html>.

27.9.2 Apache Modules

More information about external Apache modules from Section 27.4.5, “External Modules” (page 412) is available at the following locations:

`mod-apparmor`

<http://en.opensuse.org/AppArmor>

`mod_mono`

http://www.mono-project.com/Mod_mono

`mod_perl`

<http://perl.apache.org/>

mod_php5

<http://www.php.net/manual/en/install.unix.apache2.php>

mod_python

<http://www.modpython.org/>

27.9.3 Development

More information about developing Apache modules or about getting involved in the Apache Web server project are available at the following locations:

Apache Developer Information

<http://httpd.apache.org/dev/>

Apache Developer Documentation

<http://httpd.apache.org/docs/2.2/developer/>

Writing Apache Modules with Perl and C

<http://www.modperl.com/>

27.9.4 Miscellaneous Sources

If you experience difficulties specific to Apache in SUSE Linux Enterprise Server, take a look at the Technical Information Search at <http://www.novell.com/support>. The history of Apache is provided at http://httpd.apache.org/ABOUT__APACHE.html. This page also explains why the server is called Apache.

Setting up an FTP server with YaST

28

Using the YaST *FTP Server* module, you can configure your machine to function as an FTP server. Anonymous and/or authenticated users can connect to your machine and download and, depending on the configuration, upload files using the FTP protocol. YaST provides a unified configuration interface for various FTP server daemons installed on your system.

The YaST *FTP Server* configuration module can be used to configure two different FTP server daemons: vsftpd (Very Secure FTP Daemon) and pure-ftpd. Only installed servers can be configured.

The vsftpd and pure-ftpd servers have slightly different configuration options, especially in the *Experts Settings* dialog. This chapter describes the settings of the vsftpd server .

If the YaST FTP Server module is not available in your system, install the `yast2-ftp-server` package.

To configure the FTP server using YaST, follow these steps:

- 1 Open YaST Control Center and choose *Network Services > FTP Server* or run the `yast2 ftp-server` command as `root`.
- 2 If there is not any FTP server installed in your system, you will be asked which server to install when the YaST FTP Server module starts. Choose a server and confirm the dialog.
- 3 In the *Start-Up* dialog, configure the starting of the FTP server. For more information, see Section 28.1, “Starting the FTP server” (page 430).

In the *General* dialog, configure FTP directories, welcome message, file creation masks and various other parameters. For more information, see Section 28.2, “FTP General Settings” (page 431).

In the *Performance* dialog, set the parameters that affect the load on the FTP server. For more information, see Section 28.3, “FTP Performance Settings” (page 432).

In the *Authentication* dialog, set whether the FTP server should be available for anonymous and/or authenticated users. For more information, see Section 28.4, “Authentication” (page 432).

In the *Expert Settings* dialog, configure the operation mode of the FTP server, SSL connections and firewall settings. For more information, see Section 28.5, “Expert Settings” (page 433).

4 Press *Finish* to save the configurations.

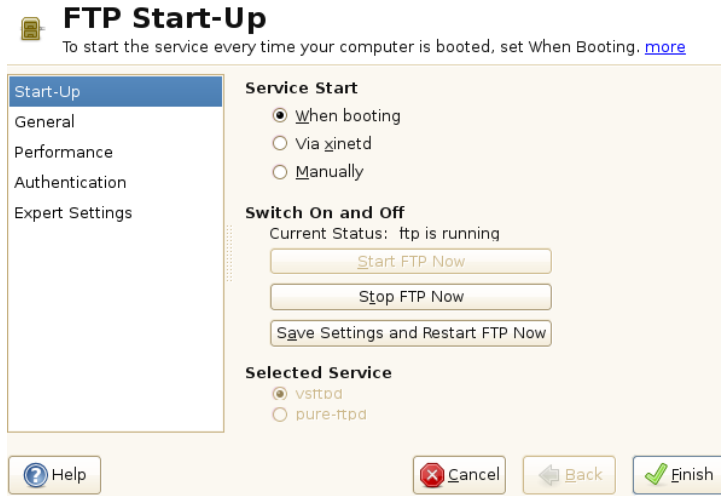
28.1 Starting the FTP server

In the *Service Start* frame of the *FTP Start-Up* dialog set the way the FTP server is started up. You can choose between starting the server automatically during the system boot and starting it manually. If the FTP server should be started only after FTP connection request, choose *Via xinetd*.

The current status of the FTP server is shown in the *Switch On and Off* frame of the *FTP Start-Up* dialog. Start the FTP server by pressing *Start FTP Now*. To stop the server, press *Stop FTP Now*. After having changed the settings of the server press *Save Settings and Restart FTP Now*. Your configurations will be saved by leaving the configuration module with *Finish*, as well.

The *Selected Service* frame of the *FTP Start-Up* dialog shows which FTP server is used. Either vsftpd (Very Secure FTP Daemon) or pure-ftpd can be used. If both servers are installed, you can switch between them—the current configuration will automatically be converted.

Figure 28.1 *FTP Server Configuration — Start-Up*



28.2 FTP General Settings

In the *General Settings* frame of the *FTP General Settings* dialog you can set the *Welcome message* which is shown after connecting to the FTP server.

If you check the *Chroot Everyone* option, all local users will be placed in a chroot jail in their home directory after login. This option has security implications, especially if the users have upload permission or shell access, so be careful enabling this option.

If you check the *Verbose Logging* option, all FTP requests and responses are logged.

You can limit permissions of files created by anonymous and/or authenticated users with umask. The bits that are set in the umask identify permissions that are always to be disabled for newly created files. Set the file creation mask for anonymous users in *Umask for Anonymous* and the file creation mask for authenticated users in *Umask for Authenticated Users*. The masks should be entered as octal numbers with a leading zero. For more information about umask, see the umask man page (`man 1p umask`).

In the *FTP Directories* frame set the directories used for anonymous and authorized users. By pressing *Browse*, you can select a directory to be used from the local filesystem.

tem. The default FTP directory for anonymous users is `/srv/ftp`. Note that `vsftpd` does not allow this directory to be writable for all users. The subdirectory `upload` with write permissions for anonymous users is created instead.

NOTE

The pure-`ftpd` server allows the FTP directory for anonymous users to be writable. When switching between servers, make sure you remove the write permissions in the directory that was used with pure-`ftpd` before switching back to the `vsftpd` server.

28.3 FTP Performance Settings

In the *Performance* dialog set the parameters which affect the load on the FTP server. *Max Idle Time* is the maximum time (in minutes) the remote client may spend between FTP commands. In case of longer inactivity, the remote client is disconnected. *Max Clients for One IP* determines the maximum number of clients which can be connected from a single IP address. *Max Clients* determines the maximum number of clients which may be connected. Any additional clients will get an error message.

The maximum data transfer rate (in KB/s) is set in *Local Max Rate* for local authenticated users, and in *Anonymous Max Rate* for anonymous clients respectively. The default value for the rate settings is 0, which means unlimited data transfer rate.

28.4 Authentication

In the *Enable/Disable Anonymous and Local Users* frame of the *Authentication* dialog, you are able to set which users are allowed to access your FTP server. You can grant access only for anonymous users, only for authenticated users with accounts on the system or for both types of users.

If you want to allow users to upload files to the FTP server, check *Enable Upload* in the *Uploading* frame of the *Authentication* dialog. Here you are able to allow uploading or creating directories even for anonymous users by checking the respective box.

NOTE

If a vsftpd server is used and you want anonymous users to be able to upload files or create directories, a subdirectory with writing permissions for all users has to be created in the anonymous FTP directory.

28.5 Expert Settings

An FTP server can run in active or in passive mode. By default the server runs in passive mode. To switch into active mode, just uncheck *Enable Passive Mode* option in *Expert Settings* dialog. You can also change the range of ports on the server used for the data stream by tweaking the *Min Port for Pas. Mode* and *Max Port for Pas. Mode* options.

If you want encrypted communication between clients and the server, you can *Enable SSL*. Check the versions of the protocol to be supported and specify the DSA certificate to be used for SSL encrypted connections.

If your system is protected by a firewall, check *Open Port in Firewall* to enable a connection to the FTP server.

28.6 For more information

For more information about ftp servers read the manual pages of `pure-ftpd`, `vsftpd` and `vsftpd.conf`.

The Squid Proxy Server

Squid is a widely-used proxy cache for Linux and UNIX platforms. This means that it stores requested Internet objects, such as data on a Web or FTP server, on a machine that is closer to the requesting workstation than the server. It may be set up in multiple hierarchies to assure optimal response times and low bandwidth usage, even in modes that are transparent for the end user. Additional software like squidGuard may be used to filter Web contents.

Squid acts as a proxy cache. It redirects object requests from clients (in this case, from Web browsers) to the server. When the requested objects arrive from the server, it delivers the objects to the client and keeps a copy of them in the hard disk cache. One of the advantages of caching is that several clients requesting the same object can be served from the hard disk cache. This enables clients to receive the data much faster than from the Internet. This procedure also reduces the network traffic.

Along with the actual caching, Squid offers a wide range of features such as distributing the load over intercommunicating hierarchies of proxy servers, defining strict access control lists for all clients accessing the proxy, allowing or denying access to specific Web pages with the help of other applications, and generating statistics about frequently-visited Web pages for the assessment of the users' surfing habits. Squid is not a generic proxy. It normally proxies only HTTP connections. It supports the protocols FTP, Gopher, SSL, and WAIS, but it does not support other Internet protocols, such as Real Audio, news, or video conferencing. Because Squid only supports the UDP protocol to provide communication between different caches, many other multimedia programs are not supported.

29.1 Some Facts about Proxy Caches

As a proxy cache, Squid can be used in several ways. When combined with a firewall, it can help with security. Multiple proxies can be used together. It can also determine what types of objects should be cached and for how long.

29.1.1 Squid and Security

It is possible to use Squid together with a firewall to secure internal networks from the outside using a proxy cache. The firewall denies all clients access to external services except Squid. All Web connections must be established by the proxy. With this configuration, Squid completely controls Web access.

If the firewall configuration includes a DMZ, the proxy should operate within this zone. Section 29.5, “Configuring a Transparent Proxy” (page 447) describes how to implement a *transparent* proxy. This simplifies the configuration of the clients, because in this case they do not need any information about the proxy.

29.1.2 Multiple Caches

Several instances of Squid can be configured to exchange objects between them. This reduces the total system load and increases the chances of finding an object already existing in the local network. It is also possible to configure cache hierarchies, so a cache is able to forward object requests to sibling caches or to a parent cache—causing it to get objects from another cache in the local network or directly from the source.

Choosing the appropriate topology for the cache hierarchy is very important, because it is not desirable to increase the overall traffic on the network. For a very large network, it would make sense to configure a proxy server for every subnetwork and connect them to a parent proxy, which in turn is connected to the proxy cache of the ISP.

All this communication is handled by ICP (Internet cache protocol) running on top of the UDP protocol. Data transfers between caches are handled using HTTP (hypertext transmission protocol) based on TCP.

To find the most appropriate server from which to get the objects, one cache sends an ICP request to all sibling proxies. These answer the requests via ICP responses with a

HIT code if the object was detected or a MISS if it was not. If multiple HIT responses were found, the proxy server decides from which server to download, depending on factors such as which cache sent the fastest answer or which one is closer. If no satisfactory responses are received, the request is sent to the parent cache.

TIP

To avoid duplication of objects in different caches in the network, other ICP protocols are used, such as CARP (cache array routing protocol) or HTCP (hypertext cache protocol). The more objects maintained in the network, the greater the possibility of finding the desired one.

29.1.3 Caching Internet Objects

Not all objects available in the network are static. There are a lot of dynamically generated CGI pages, visitor counters, and encrypted SSL content documents. Objects like this are not cached because they change each time they are accessed.

The question remains as to how long all the other objects stored in the cache should stay there. To determine this, all objects in the cache are assigned one of various possible states. Web and proxy servers find out the status of an object by adding headers to these objects, such as “Last modified” or “Expires” and the corresponding date. Other headers specifying that objects must not be cached are used as well.

Objects in the cache are normally replaced, due to a lack of free hard disk space, using algorithms such as LRU (last recently used). Basically this means that the proxy expunges the objects that have not been requested for the longest time.

29.2 System Requirements

The most important thing is to determine the maximum network load the system must bear. Therefore, it is important to pay more attention to the load peaks, because these might be more than four times the day's average. When in doubt, it would be better to overestimate the system's requirements, because having Squid working close to the limit of its capabilities could lead to a severe loss in the quality of the service. The following sections point to the system factors in order of significance.

29.2.1 Hard Disks

Speed plays an important role in the caching process, so this factor deserves special attention. For hard disks, this parameter is described as *random seek time*, measured in milliseconds. Because the data blocks that Squid reads from or writes to the hard disk tend to be rather small, the seek time of the hard disk is more important than its data throughput. For the purposes of a proxy, hard disks with high rotation speeds are probably the better choice, because they allow the read-write head to be positioned in the required spot more quickly. One possibility to speed up the system is to use a number of disks concurrently or to employ striping RAID arrays.

29.2.2 Size of the Disk Cache

In a small cache, the probability of a HIT (finding the requested object already located there) is small, because the cache is easily filled and the less requested objects are replaced by newer ones. If, for example, one GB is available for the cache and the users only surf ten MB per day, it would take more than one hundred days to fill the cache.

The easiest way to determine the needed cache size is to consider the maximum transfer rate of the connection. With a 1 Mbit/s connection, the maximum transfer rate is 125 KB/s. If all this traffic ends up in the cache, in one hour it would add up to 450 MB and, assuming that all this traffic is generated in only eight working hours, it would reach 3.6 GB in one day. Because the connection is normally not used to its upper volume limit, it can be assumed that the total data volume handled by the cache is approximately 2 GB. This is why 2 GB of disk space is required in the example for Squid to keep one day's worth of browsed data cached.

29.2.3 RAM

The amount of memory (RAM) required by Squid directly correlates to the number of objects in the cache. Squid also stores cache object references and frequently requested objects in the main memory to speed up retrieval of this data. Random access memory is much faster than a hard disk.

In addition to that, there is other data that Squid needs to keep in memory, such as a table with all the IP addresses handled, an exact domain name cache, the most frequently requested objects, access control lists, buffers, and more.

It is very important to have sufficient memory for the Squid process, because system performance is dramatically reduced if it must be swapped to disk. The `cachemgr.cgi` tool can be used for the cache memory management. This tool is introduced in Section 29.6, “`cachemgr.cgi`” (page 450).

29.2.4 CPU

Squid is not a program that requires intensive CPU usage. The load of the processor is only increased while the contents of the cache are loaded or checked. Using a multiprocessor machine does not increase the performance of the system. To increase efficiency, it is better to buy faster disks or add more memory.

29.3 Starting Squid

If not already installed, install the `squid` package. `squid` does not belong to the default SUSE Linux Enterprise Server installation scope.

Squid is already preconfigured in SUSE® Linux Enterprise Server, you can start it right after the installation. To ensure a smooth start-up, the network should be configured in a way that at least one name server and the Internet can be reached. Problems can arise if a dial-up connection is used with a dynamic DNS configuration. In this case, at least the name server should be entered, because Squid does not start if it does not detect a DNS server in `/etc/resolv.conf`.

29.3.1 Commands for Starting and Stopping Squid

To start Squid, enter `rcsquid start` at the command line as `root`. In the initial start-up, the directory structure of the cache must first be defined in `/var/cache/squid`. This is done automatically by the start script `/etc/init.d/squid` and can take a few seconds or even minutes. If `done` appears to the right in green, Squid has been successfully loaded. To test the functionality of Squid on the local system, enter `localhost` as the proxy and `3128` as the port in the browser.

Then in the `/etc/squid/squid.conf` configuration file define ACLs that control access to the proxy. More information about ACLs, see Section 29.4.2, “Options for Access Controls” (page 444) and the comment in the `squid.conf` file near `http_access`.

After modifying the configuration file `/etc/squid/squid.conf`, Squid must reload the configuration file. Do this with `rcsquid reload`. Alternatively, completely restart Squid with `rcsquid restart`.

The command `rcsquid status` can be used to check whether the proxy is running. The command `rcsquid stop` causes Squid to shut down. This can take a while, because Squid waits up to half a minute (`shutdown_lifetime` option in `/etc/squid/squid.conf`) before dropping the connections to the clients and writing its data to the disk.

WARNING: Terminating Squid

Terminating Squid with `kill` or `killall` can damage the cache. To be able to restart Squid, a damaged cache must be deleted.

If Squid dies after a short period of time even though it was started successfully, check whether there is a faulty name server entry or whether the `/etc/resolv.conf` file is missing. Squid logs the cause of a start-up failure in the `/var/log/squid/cache.log` file. If Squid should be loaded automatically when the system boots, use the YaST runlevel editor to activate Squid for the desired runlevels. See Section 7.2.3, “Configuring System Services (Runlevel) with YaST” (page 76).

An uninstall of Squid does not remove the cache hierarchy or the log files. To remove these, delete the `/var/cache/squid` directory manually.

29.3.2 Local DNS Server

Setting up a local DNS server makes sense even if it does not manage its own domain. It then simply acts as a caching-only name server and is also able to resolve DNS requests via the root name servers without requiring any special configuration (see Section 21.4, “Starting the Name Server BIND” (page 300)). How this can be done depends on whether or not you chose dynamic DNS during the configuration of the Internet connection.

Dynamic DNS

Normally, with dynamic DNS, the DNS server is set by the provider during the establishment of the Internet connection and the local `/etc/resolv.conf` file is adjusted automatically. This behavior is controlled in the `/etc/sysconfig/network/config` file with the `NETCONFIG_DNS_POLICY` sysconfig variable. Set `NETCONFIG_DNS_POLICY` to `" "` with the YaST sysconfig editor (see Section 7.3.1, “Changing the System Configuration Using the YaST sysconfig Editor” (page 78)). Then enter the local DNS server in the `/etc/resolv.conf` file with the IP address `127.0.0.1` for `localhost`. This way Squid can always find the local name server when it starts.

To make the provider's name server accessible, enter it in the configuration file `/etc/named.conf` under `forwarders` along with its IP address. With dynamic DNS, this can be achieved automatically during connection establishment by setting the sysconfig variable `NETCONFIG_DNS_POLICY` to `auto`.

Static DNS

With static DNS, no automatic DNS adjustments take place while establishing a connection, so there is no need to change any sysconfig variables. You must, however, enter the local DNS server in the file `/etc/resolv.conf` as described above. Additionally, the providers static name server must be entered manually in the `/etc/named.conf` file under `forwarders` along with its IP address.

TIP: DNS and Firewall

If you have a firewall running, make sure DNS requests can pass it.

29.4 The `/etc/squid/squid.conf` Configuration File

All Squid proxy server settings are made in the `/etc/squid/squid.conf` file. To start Squid for the first time, no changes are necessary, but external clients are initially denied access. The proxy is available for `localhost`. The default port is `3128`. The preinstalled `/etc/squid/squid.conf` configuration file provides detailed information about the options and many examples. Nearly all entries begin with `#` (the lines are commented) and the relevant specifications can be found at the end of the line.

The given values almost always correlate with the default values, so removing the comment signs without changing any of the parameters actually has little effect in most cases. If possible, leave the sample as it is and insert the options along with the modified parameters in the line below. This way, the default values may easily be recovered and compared with the changes.

TIP: Adapting the Configuration File after an Update

If you have updated from an earlier Squid version, it is recommended to edit the new `/etc/squid/squid.conf` and only apply the changes made in your previous file. If you try to use the old `squid.conf`, you risk that the configuration no longer works, because options are sometimes modified and new changes added.

29.4.1 General Configuration Options (Selection)

`http_port 3128`

This is the port on which Squid listens for client requests. The default port is 3128, but 8080 is also common. If desired, specify several port numbers separated by blank spaces.

`cache_peer hostname type proxy-port icp-port`

Here, enter a parent proxy, for example, if you want to use the proxy of your ISP. As *hostname*, enter the name or IP address of the proxy to use and, as *type*, enter *parent*. For *proxy-port*, enter the port number that is also given by the operator of the parent for use in the browser (usually 8080). Set the *icp-port* to 7 or 0 if the ICP port of the parent is not known and its use is irrelevant to the provider. In addition, *default* and *no-query* may be specified after the port numbers to prohibit the use of the ICP protocol. Squid then behaves like a normal browser as far as the provider's proxy is concerned.

`cache_mem 8 MB`

This entry defines the amount of memory Squid can use for very popular replies. The default is 8 MB. This does not specify the memory usage of Squid and may be exceeded.

`cache_dir ufs /var/cache/squid/ 100 16 256`

The entry *cache_dir* defines the directory where all the objects are stored on disk. The numbers at the end indicate the maximum disk space in MB to use and the number of directories in the first and second level. The *ufs* parameter should be left alone. The default is 100 MB occupied disk space in the `/var/cache/squid` directory and creation of 16 subdirectories inside it, each containing 256 more subdirectories. When specifying the disk space to use, leave sufficient reserve disk space. Values from a minimum of 50% to a maximum of 80% of the available disk space make the most sense here. The last two numbers for the directories should only be increased with caution, because too many directories can also lead to performance problems. If you have several disks that share the cache, enter several *cache_dir* lines.

`cache_access_log /var/log/squid/access.log , cache_log /var/log/squid/cache.log ,
cache_store_log /var/log/squid/store.log`

These three entries specify the paths where Squid logs all its actions. Normally, nothing is changed here. If Squid is experiencing a heavy usage burden, it might make sense to distribute the cache and the log files over several disks.

`emulate_httpd_log off`

If the entry is set to *on*, obtain readable log files. Some evaluation programs cannot interpret this, however.

`client_netmask 255.255.255.255`

With this entry, mask IP addresses of clients in the log files. The last digit of the IP address is set to zero if you enter `255.255.255.0` here. You may protect the privacy of your clients this way.

`ftp_user Squid@`

With this, set the password Squid should use for the anonymous FTP login. It can make sense to specify a valid e-mail address here, because some FTP servers check these for validity.

`cache_mgr webmaster`

An e-mail address to which Squid sends a message if it unexpectedly crashes. The default is *webmaster*.

`logfile_rotate 0`

If you run `squid -k rotate`, Squid can rotate secured log files. The files are numbered in this process and, after reaching the specified value, the oldest file is

overwritten. The default value is 0 because archiving and deleting log files in SUSE Linux Enterprise Server is carried out by a cron job set in the configuration file `/etc/logrotate/squid`.

`append_domain domain`

With `append_domain`, specify which domain to append automatically when none is given. Usually, your own domain is entered here, so entering `www` in the browser accesses your own Web server.

`forwarded_for on`

If you set the entry to *off*, Squid removes the IP address and the system name of the client from HTTP requests. Otherwise it adds a line to the header like

```
X-Forwarded-For: 192.168.0.1
```

`negative_ttl 5 minutes; negative_dns_ttl 5 minutes`

Normally, you do not need to change these values. If you have a dial-up connection, however, the Internet may, at times, not be accessible. Squid makes a note of the failed requests then refuses to issue new ones, although the Internet connection has been reestablished. In a case such as this, change the *minutes* to *seconds*. Then, after clicking *Reload* in the browser, the dial-up process should be reengaged after a few seconds.

`never_direct allow acl_name`

To prevent Squid from taking requests directly from the Internet, use the above command to force connection to another proxy. This must have previously been entered in `cache_peer`. If `all` is specified as the `acl_name`, force all requests to be forwarded directly to the *parent*. This might be necessary, for example, if you are using a provider that strictly stipulates the use of its proxies or denies its firewall direct Internet access.

29.4.2 Options for Access Controls

Squid provides a detailed system for controlling the access to the proxy. By implementing ACLs, it can be configured easily and comprehensively. This involves lists with rules that are processed sequentially. ACLs must be defined before they can be used. Some default ACLs, such as *all* and *localhost*, already exist. However, the mere definition of an ACL does not mean that it is actually applied. This only happens in conjunction with `http_access` rules.

`acl <acl_name> <type> <data>`

An ACL requires at least three specifications to define it. The name *<acl_name>* can be chosen arbitrarily. For *<type>*, select from a variety of different options, which can be found in the *ACCESS CONTROLS* section in the `/etc/squid/squid.conf` file. The specification for *<data>* depends on the individual ACL type and can also be read from a file, for example, via hostnames, IP addresses, or URLs. The following are some simple examples:

```
acl mysurfers srcdomain .my-domain.com
acl teachers src 192.168.1.0/255.255.255.0
acl students src 192.168.7.0-192.168.9.0/255.255.255.0
acl lunch time MTWHF 12:00-15:00
```

`http_access allow <acl_name>`

http_access defines who is allowed to use the proxy and who can access what on the Internet. For this, ACLs must be given. *localhost* and *all* have already been defined above, which can deny or allow access via *deny* or *allow*. A list containing any number of *http_access* entries can be created, processed from top to bottom, and, depending on which occurs first, access is allowed or denied to the respective URL. The last entry should always be *http_access deny all*. In the following example, the *localhost* has free access to everything while all other hosts are denied access completely.

```
http_access allow localhost
http_access deny all
```

In another example using these rules, the group *teachers* always has access to the Internet. The group *students* only gets access Monday to Friday during lunch time.

```
http_access deny localhost
http_access allow teachers
http_access allow students lunch time
http_access deny all
```

The list with the *http_access* entries should only be entered, for the sake of readability, at the designated position in the `/etc/squid/squid.conf` file. That is, between the text

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR  
# CLIENTS
```

and the last

```
http_access deny all
```

`redirect_program /usr/bin/squidGuard`

With this option, specify a redirector such as squidGuard, which allows the blocking of unwanted URLs. Internet access can be individually controlled for various user groups with the help of proxy authentication and the appropriate ACLs. squidGuard is a separate package that can be installed and configured.

`auth_param basic program /usr/sbin/pam_auth`

If users must be authenticated on the proxy, set a corresponding program, such as pam_auth. When accessing pam_auth for the first time, the user sees a login window in which to enter the username and password. In addition, an ACL is still required, so only clients with a valid login can use the Internet:

```
acl password proxy_auth REQUIRED
```

```
http_access allow password  
http_access deny all
```

The *REQUIRED* after *proxy_auth* can be replaced with a list of permitted usernames or with the path to such a list.

`ident_lookup_access allow <acl_name>`

With this, have an ident request run for all ACL-defined clients to find each user's identity. If you apply *all* to the *<acl_name>*, this is valid for all clients. Also, an ident daemon must be running on all clients. For Linux, install the pidentd package for this purpose. For Microsoft Windows, free software is available for download from the Internet. To ensure that only clients with a successful ident lookup are permitted, define a corresponding ACL here:

```
acl identhosts ident REQUIRED
```

```
http_access allow identhosts  
http_access deny all
```

Here, too, replace *REQUIRED* with a list of permitted usernames. Using *ident* can slow down the access time quite a bit, because ident lookups are repeated for each request.

29.5 Configuring a Transparent Proxy

The usual way of working with proxy servers is the following: the Web browser sends requests to a certain port in the proxy server and the proxy provides these required objects, whether they are in its cache or not. When working in a network, several situations may arise:

- For security reasons, it is recommended that all clients use a proxy to surf the Internet.
- All clients must use a proxy, regardless of whether they are aware of it.
- The proxy in a network is moved, but the existing clients need to retain their old configuration.

In all these cases, a transparent proxy may be used. The principle is very easy: the proxy intercepts and answers the requests of the Web browser, so the Web browser receives the requested pages without knowing from where they are coming. As the name indicates, the entire process is done transparently.

29.5.1 Configuration Options in `/etc/squid/squid.conf`

To inform squid that it should act as a transparent proxy, use the option `transparent` at the tag `http_port` in the main `/etc/squid/squid.conf` configuration file. After restarting squid, you must reconfigure the firewall to redirect the `http` port to the port given in `http_port`. In the following squid config line, this would be the port 3128.

```
http_port 3128 transparent
```

29.5.2 Firewall Configuration with SuSEfirewall2

Now redirect all incoming requests via the firewall with a port forwarding rule to the Squid port. Use the SuSEfirewall2 tool, described in Section “Configuring the Firewall

with YaST” (Chapter 15, *Masquerading and Firewalls*, ↑*Security Guide*). Its `/etc/sysconfig/SuSEfirewall2` configuration file consists of well-documented entries. To configure a transparent proxy, you must set several firewall options:

- Device pointing to the Internet: `FW_DEV_EXT="eth1"`
- Device pointing to the network: `FW_DEV_INT="eth0"`

Define ports and services (see `/etc/services`) on the firewall that are accessed from untrusted (external) networks such as the Internet. In this example, only Web services are offered to the outside:

```
FW_SERVICES_EXT_TCP="www"
```

Define ports or services (see `/etc/services`) on the firewall that are accessed from the secure (internal) network, both via TCP and UDP:

```
FW_SERVICES_INT_TCP="domain www 3128"  
FW_SERVICES_INT_UDP="domain"
```

This allows accessing Web services and Squid (whose default port is 3128). The service “domain” stands for DNS (domain name service). This service is commonly used. Otherwise, simply take it out of the above entries and set the following option to `no`:

```
FW_SERVICE_DNS="yes"
```

The most important option is option number 15:

Example 29.1 Firewall Configuration: Option 15

```
# 15.)
# Which accesses to services should be redirected to a local port on
# the firewall machine?
#
# This option can be used to force all internal users to surf via
# your squid proxy, or transparently redirect incoming webtraffic to
# a secure webserver.
#
# Format:
# list of <source network>[,<destination network>,<protocol>[,dport[:lport]]
# Where protocol is either tcp or udp. dport is the original
# destination port and lport the port on the local machine to
# redirect the traffic to
#
# An exclamation mark in front of source or destination network
# means everything EXCEPT the specified network
#
# Example: "10.0.0.0/8,0/0,tcp,80,3128 0/0,172.20.1.1,tcp,80,8080"
```

The comments above show the syntax to follow. First, enter the IP address and the netmask of the internal networks accessing the proxy firewall. Second, enter the IP address and the netmask to which these clients send their requests. In the case of Web browsers, specify the networks 0/0, a wild card that means “to everywhere.” After that, enter the original port to which these requests are sent and, finally, the port to which all these requests are redirected. Because Squid supports protocols other than HTTP, redirect requests from other ports to the proxy, such as FTP (port 21), HTTPS, or SSL (port 443). In this example, Web services (port 80) are redirected to the proxy port (port 3128). If there are more networks or services to add, they must be separated by a blank space in the respective entry.

```
FW_REDIRECT="192.168.0.0/16,0/0,tcp,80,3128"
```

To start the firewall and the new configuration with it, change an entry in the `/etc/sysconfig/SuSEfirewall12` file. The entry `START_FW` must be set to "yes".

Start Squid as shown in Section 29.3, “Starting Squid” (page 439). To verify that everything is working properly, check the Squid logs in `/var/log/squid/access.log`. To verify that all ports are correctly configured, perform a port scan on the machine from any computer outside your network. Only the Web services (port 80) should be open. To scan the ports with `nmap`, the command syntax is `nmap -O IP_address`.

29.6 cachemgr.cgi

The cache manager (`cachemgr.cgi`) is a CGI utility for displaying statistics about the memory usage of a running Squid process. It is also a more convenient way to manage the cache and view statistics without logging the server.

29.6.1 Setup

First, a running Web server on your system is required. Configure Apache as described in Chapter 27, *The Apache HTTP Server* (page 387). To check if Apache is already running, as `root` enter the command `rcapache status`. If a message like this appears:

```
Checking for service httpd: OK
Server uptime: 1 day 18 hours 29 minutes 39 seconds
```

Apache is running on the machine. Otherwise, enter `rcapache start` to start Apache with the SUSE Linux Enterprise Server default settings. The last step to set it up is to copy the file `cachemgr.cgi` to the Apache directory `cgi-bin`. For 32-bit, this works as follows:

```
cp /usr/lib/squid/cachemgr.cgi /srv/www/cgi-bin/
```

In a 64-bit environment, the file `cachemgr.cgi` is located below `/usr/lib64/squid/` and the command to copy it to the Apache directory is the following:

```
cp /usr/lib64/squid/cachemgr.cgi /srv/www/cgi-bin/
```

29.6.2 Cache Manager ACLs in `/etc/squid/squid.conf`

There are some default settings in the original file required for the cache manager. First, two ACLs are defined, then `http_access` options use these ACLs to grant access from the CGI script to Squid. The first ACL is the most important, because the cache manager tries to communicate with Squid over the `cache_object` protocol.

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
```

The following rules give Apache the access rights to Squid:

```
http_access allow manager localhost
http_access deny manager
```

These rules assume that the Web server and Squid are running on the same machine. If the communication between the cache manager and Squid originates at the Web server on another computer, include an extra ACL as in Example 29.2, “Access Rules” (page 451).

Example 29.2 Access Rules

```
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl webserver src 192.168.1.7/255.255.255.255 # webserver IP
```

Then add the rules in Example 29.3, “Access Rules” (page 451) to permit access from the Web server.

Example 29.3 Access Rules

```
http_access allow manager localhost
http_access allow manager webserver
http_access deny manager
```

Configure a password for the manager for access to more options, like closing the cache remotely or viewing more information about the cache. For this, configure the entry `cachemgr_passwd` with a password for the manager and the list of options to view. This list appears as a part of the entry comments in `/etc/squid/squid.conf`.

Restart Squid every time the configuration file is changed. Do this easily with `rcsquid reload`.

29.6.3 Viewing the Statistics

Go to the corresponding Web site—<http://webserver.example.org/cgi-bin/cachemgr.cgi>. Press *continue* and browse through the different statistics.

More details for each entry shown by the cache manager is in the Squid FAQ at <http://www.squid-cache.org/Doc/FAQ/FAQ-9.html>.

29.7 Cache Report Generation with Calamaris

Calamaris is a Perl script used to generate reports of cache activity in ASCII or HTML format. It works with native Squid access log files. The Calamaris home page is located at <http://Calamaris.Cord.de/>. This tool does not belong to the SUSE Linux Enterprise Server default installation scope—to use it, install the `calamaris` package.

Log in as `root` then enter `cat access.log | calamaris options > reportfile`. It is important when piping more than one log file that the log files are chronologically ordered with older files first. These are some options of the program:

TIP: Shell and File Sequences

If you have multiple similar files like `access.log.1`, `access.log.2` and so on, the default shell `bash` would sort those files not in the number sequence when listing `access.log.*`. To solve this issue, you may use the syntax `access.log.{1..42}` which generates a list of files extended with numbers from 1 to 42.

- `-a`
output all available reports
- `-w`
output as HTML report
- `-l`
include a message or logo in report header

More information about the various options can be found in the program's manual page with `man calamaris`.

A typical example is:

```
cat access.log.{10..1} access.log | calamaris -a -w \  
> /usr/local/httpd/htdocs/Squid/squidreport.html
```

This puts the report in the directory of the Web server. Apache is required to view the reports.

29.8 For More Information

Visit the home page of Squid at <http://www.squid-cache.org/>. Here, find the “Squid User Guide” and a very extensive collection of FAQs on Squid. The “CompleteFAQ” is shipped with the squid package in `/usr/share/doc/packages/squid/FAQ`.

A HOWTO about transparent proxies is available in the `howto` package as `/usr/share/doc/howto/en/txt/TransparentProxy.gz`. In addition, mailing lists are available for Squid at squid-users@squid-cache.org. The mailing list archive is located at <http://www.squid-cache.org/mail-archive/squid-users/>.

Web Based Enterprise Management using SFCB

30

30.1 Introduction and Basic Concept

SUSE® Linux Enterprise Server (SLES) provides a collection of open standards based tools for the unified management of disparate computing systems and environments. Our enterprise solutions implement the standards proposed by the Distributed Management Task Force. The following paragraphs describe their basic components.

Distributed Management Task Force, Inc (DMTF) is the industry organization which leads the development of management standards for enterprise and Internet environments. Their goal is to unify management standards and initiatives, and to enable more integrated, cost effective and interoperable management solutions. DMTF standards provide common system management components for control and communication. Their solutions are independent of platforms and technologies. *Web Based Enterprise Management* and *Common Information Model* are one of their key technologies.

Web Based Enterprise Management (WBEM) is a set of management and Internet standard technologies. WBEM is developed to unify the management of enterprise computing environments. It provides the ability for the industry to deliver a well-integrated collection of management tools using Web technologies. WBEM consists of the following standards:

- A data model: the Common Information Model (CIM) standard
- An encoding specification: CIM-XML Encoding Specification

- A transport mechanism: CIM operations over HTTP

Common Information Model is a conceptual information model that describes system management. It is not bound to a particular implementation and enables the interchange of management information between management systems, networks, services and applications. There are two parts to CIM: the CIM Specification and the CIM Schema:

- The *CIM Specification* describes the language, naming and meta schema. The meta schema is a formal definition of the model. It defines the terms used to express the model and their usage and semantics. The elements of the meta schema are *classes*, *properties*, and *methods*. The meta schema also supports *indications* and *associations* as types of *classes*, and *references* as types of *properties*.
- The *CIM Schema* provides the actual model descriptions. It supplies a set of classes with properties and associations that provide a well understood conceptual framework within which it is possible to organize the available information about the managed environment.

The Common Information Model Object Manager (CIMOM) is a CIM object manager or, more specifically, an application that manages objects according to the CIM standard. CIMOM manages communication between CIMOM providers and a CIM client, where the administrator manages the system.

CIMOM providers are software performing specific tasks within the CIMOM that are requested by client applications. Each provider instruments one or more aspects of the CIMOM's schema. These providers interact directly with the hardware.

Standards Based Linux Instrumentation for Manageability (SBLIM) is a collection of tools designed to support Web-Based Enterprise Management (WBEM). SUSE® Linux Enterprise Server uses the open source CIMOM (or CIM server) from the SBLIM project called *Small Footprint CIM Broker*.

Small Footprint CIM Broker is a CIM server intended for use in resource-limited or embedded environments. It is designed to be modular and lightweight at the same time. Its based on open standards and it supports CMPI providers, CIM-XML encoding, and *Managed Object Format (MOF)*. It is highly configurable and performs stability even if the provider crashes. It is also easily accessible as it supports various transport protocols, such as HTTP, HTTPS, Unix domain sockets, Service Location Protocol (SLP), and Java Database Connectivity (JDBC).

30.2 Setting Up SFCB

To set up the Small Footprint CIM Broker (SFCB) environment, make sure the *Web-Based Enterprise Management* pattern in YaST is selected during SUSE Linux Enterprise Server installation. Alternatively, select it as a component to install on a server that is already running. Make sure the following packages are installed on your system:

`cim-schema`, Common Information Model (CIM) Schema

Contains the Common Information Model (CIM). CIM is a model for describing overall management information in a network or enterprise environments. CIM consists of a specification and a schema. The specification defines the details for integration with other management models. The schema provides the actual model descriptions.

`cmapi-bindings-pywbem`

Contains an adapter to write and run CMPI-type CIM providers in Python.

`cmapi-pywbem-base`

Contains base system CIM providers.

`cmapi-pywbem-power-management`

Contains power management providers based on DSP1027.

`python-pywbem`

Contains a Python module for making CIM operation calls through the WBEM protocol to query and update managed objects.

`cmapi-provider-register`, CIMOM neutral provider registration utility

Contains a utility allowing CMPI provider packages to register with whatever CIMOM happens to be present on the system.

`sblim-sfcb`, Small Footprint CIM Broker

Contains Small Footprint CIM Broker. It is a CIM server conforming to the CIM Operations over HTTP protocol. It is robust, with low resource consumption and, therefore, specifically suited for embedded and resource constrained environments. SFCB supports providers written against the Common Manageability Programming Interface (CMPI).

`sblim-sfcc`

Contains Small Footprint CIM Client library runtime libraries.

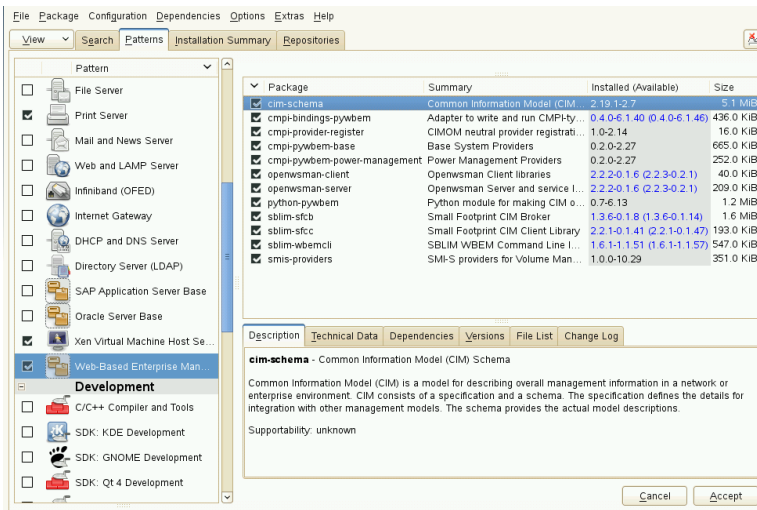
sblim-wbemcli

Contains WBEM command line interface. It is a standalone command line WBEM client especially suited for basic systems management tasks.

smis-providers

Contains providers to instrument the volumes and snapshots on the Linux filesystem. These are based on SNIA's SMI-S volume management profile and Copy Services profile respectively.

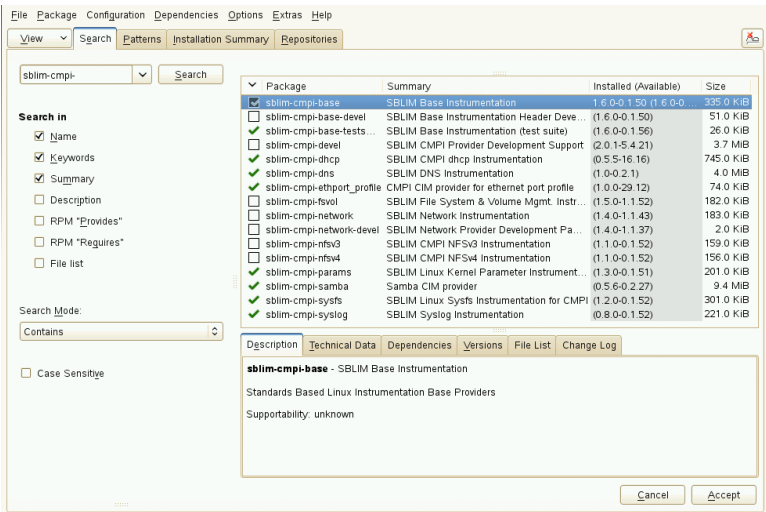
Figure 30.1 *Package Selection for Web-Based Enterprise Management Pattern*



30.2.1 Installing additional providers

SUSE® Linux Enterprise Server software repository includes additional CIM providers that are not found in the Web-Based Enterprise Management installation pattern. You can easily get their list and installation status by searching the pattern `sblim-cmpi-` in the YaST software installation module. These providers cover various tasks of system management, such as dhcp, NFS, or kernel parameters setting. It is useful to install those providers which you are going to use with SFCB.

Figure 30.2 *Package selection of additional CIM providers*



30.2.2 Starting, Stopping and Checking Status for SFCB

CIM server `sfcbd` daemon is installed together with Web-Based Enterprise Management software and is started by default at system startup. The following table explains how to start, stop and check status for `sfcbd`.

Table 30.1 *Commands for Managing `sfcbd`*

Task	Linux Command
Start <code>sfcbd</code>	Enter <code>rcsfcb start</code> as root in the command line.
Stop <code>sfcbd</code>	Enter <code>rcsfcb stop</code> as root in the command line.
Check <code>sfcbd</code> status	Enter <code>rcsfcb status</code> as root in the command line.

30.2.3 Ensuring Secure Access

The default setup of SFCB is relatively secure. However, check that the access to SFCB components is as secure as required for your organization.

Certificates

Secure Socket Layers (SSL) transports require a certificate for secure communication to occur. When SFCB is installed, it has a self-signed certificate generated.

You can replace the path to the default certificate with a path to a commercial or self-signed one by changing the `sslCertificateFilePath`: *path_filename* setting in `/etc/sfcb/sfcb.cfg`. The file must be in PEM format.

The default generated server certificate is in the following location:

```
/etc/sfcb/server.pem
```

NOTE: Paths to SSL Certificates

The default generated certificate files `servercert.pem` and `serverkey.pem` are stored under `/etc/ssl/servercerts` directory. Files `/etc/sfcb/client.pem`, `/etc/sfcb/file.pem` and `/etc/sfcb/server.pem` are symbolic links to these files.

If you want to generate a new certificate, enter the following command as `root` in the command line:

```
tux@mercury:~> sh /usr/share/sfcb/genSslCert.sh
Generating SSL certificates in .
Generating a 2048 bit RSA private key
.....+++
.+++
writing new private key to '/var/tmp/sfcb.0Bjt69/key.pem'
-----
```

By default, the script generates certificates `client.pem`, `file.pem` and `server.pem` in the current working directory. If you want the script to generate the certificates in `/etc/sfcb` directory, you need to append it to the command. If these files already exist, a warning message is displayed and the old certificates are not overwritten.

```
tux@mercury:~> sh /usr/share/sfcb/genSslCert.sh /etc/sfcb
Generating SSL certificates in .
WARNING: server.pem SSL Certificate file already exists.
        old file will be kept intact.
WARNING: client.pem SSL Certificate trust store already exists.
        old file will be kept intact.
```

You must remove the old certificates from the file system and run the command again.

If you want to change the way SFCB uses certificates, see Section “Authentication” (page 462).

Ports

By default, SFCB is configured to accept all communications through the secure port 5989. The following paragraphs explain the communication port setup and recommended configuration.

Port 5989 (secure)

The secure port that SFCB communications use via HTTPS services. This is the default. With this setting, all communications between the CIMOM and client applications are encrypted when sent over the Internet between servers and workstations. Users must authenticate with the client application to reach SFCB server. We recommend that you keep this setting. In order for the SFCB CIMOM to communicate with the necessary applications, this port must be open on routers and firewalls if they are present between the client application and the nodes being monitored.

Port 5988 (insecure)

The insecure port that SFCB communications use via HTTP services. This setting is disabled by default. With this setting, all communications between the CIMOM and client applications are open for review when sent over the Internet between servers and workstations by anyone, without any authentication. We recommend that you use this setting only when attempting to debug a problem with the CIMOM. As soon as the problem is resolved, disable the non-secure port option back. In order for the SFCB CIMOM to communicate with the necessary applications that require non-secure access, this port must be open in routers and firewalls between the client application and the nodes being monitored.

If you want to change the default port assignments, see Section “Ports” (page 461).

Authentication

SFCB supports HTTP basic authentication and authentication based on client certificates (HTTP over SSL connections). Basic HTTP authentication is enabled by specifying `doBasicAuth=true` in the SFCB configuration file (`/etc/sfcb/sfcb.cfg` by default). SUSE® Linux Enterprise Server installation of SFCB supports Pluggable Authentication Modules (PAM) approach; therefore the local root user can authenticate to the SFCB CIMOM with local root user credentials.

If the `clientCertificate` configuration property is set to *accept* or *require*, the SFCB HTTP adapter will request a certificate from clients when connecting via HTTP over SSL (HTTPS). If *require* is specified, the client **must** provide a valid certificate (according to the client trust store specified via `sslClientTrustStore`). If the client fails to do so, the connection will be rejected by the CIM server.

The setting `clientCertificate=accept` may not be obvious. It is very useful if both basic and client certificate authentication are allowed. If the client is able to provide a valid certificate, HTTPS connection will be established and the basic authentication procedure will not be executed. If this function cannot verify the certificate, the HTTP basic authentication will take place instead.

30.3 SFCB CIMOM Configuration

SFCB is a lightweight implementation of the CIM server, but it is also highly configurable. Several options can control its behavior. Basically, you can control the SFCB server in three ways:

- by setting appropriate environment variables
- by using command line options
- by changing its configuration file

30.3.1 Environment Variables

Several environment variables directly affect the behavior of SFCB. You need to restart the SFCB daemon by `rcsfcb restart` for these changes to take effect.

`SFCB_PAUSE_PROVIDER`

Specifies the provider name. The SFCB server pauses after the provider is loaded for the first time. You can then attach a runtime debugger to the provider's process for debugging purposes.

`SFCB_PAUSE_CODEC`

Specifies the name of the SFCB codec. The SFCB server pauses after the codec is loaded for the first time. You can then attach a runtime debugger to the process.

`SFCB_TRACE`

Specifies the level of debug messages for SFCB. Valid values are 0 (no debug messages), or 1 (key debug messages) to 4 (all debug messages). Default is 1.

`SFCB_TRACE_FILE`

By default, SFCB outputs its debug messages to standard error output (STDERR). Setting this variable causes the debug messages to be written to a specified file instead.

`SBLIM_TRACE`

Specifies the level of debug messages for SBLIM providers. Valid values are 0 (no debug messages), or 1 (key debug messages) to 4 (all debug messages).

`SBLIM_TRACE_FILE`

By default, SBLIM provider outputs its trace messages to STDERR. Setting this variable causes the trace messages to be written to a specified file instead.

30.3.2 Command Line Options

`sfcbd`, the SFCB daemon, has several command line options that switch particular runtime features on or off. Enter these options when SFCB daemon starts.

`-c, --config-file=FILE`

When SFCB daemon starts, it reads its configuration from `/etc/sfcb/sfcb.cfg` by default. With this option, you can specify an alternative configuration file.

`-d, --daemon`

Forces `sfcbd` and its child processes to run in the background.

`-s, --collect-stats`

Turns on runtime statistics collecting. Various `sfcdbd` runtime statistics will be written to the `sfcdbStat` file in the current working directory. By default, no statistics are collected.

`-l, --syslog-level=LOGLEVEL`

Specifies the level of verbosity for syslog. `LOGLEVEL` can be one of `LOG_INFO`, `LOG_DEBUG`, or `LOG_ERR`, which is the default.

`-t, --trace-components=NUM`

Activates component-level tracing messages, where `NUM` is an OR-ed bitmask integer that defines which component to trace. After you specify `-t ?`, it lists all the components and their associated integer bitmask:

```
tux@mercury:~> sfcdbd -t ?
--- Traceable Components:      Int      Hex
--- providerMgr:                1 0x00000001
--- providerDrv:                2 0x00000002
--- cimxmlProc:                 4 0x00000004
--- httpDaemon:                 8 0x00000008
--- upCalls:                    16 0x00000010
--- encCalls:                    32 0x00000020
--- ProviderInstMgr:            64 0x00000040
--- providerAssocMgr:          128 0x00000080
--- providers:                  256 0x00000100
--- indProvider:                512 0x00000200
--- internalProvider:          1024 0x00000400
--- objectImpl:                 2048 0x00000800
--- xmlIn:                       4096 0x00010000
--- xmlOut:                       8192 0x00020000
--- sockets:                    16384 0x00040000
--- memoryMgr:                   32768 0x00080000
--- msgQueue:                     65536 0x00100000
--- xmlParsing:                 131072 0x00200000
--- responseTiming:             262144 0x00400000
--- dbpdaemon:                   524288 0x00800000
--- slp:                         1048576 0x01000000
```

A useful value that reveals the internal functions of `sfcdbd` but does not generate too many messages, is `-t 2019`.

30.3.3 SFCB Configuration File

SFCB reads its runtime configuration from configuration file `/etc/sfcb/sfcb.cfg` after starting up. This behavior can be overridden using `-c` option at start-up.

The configuration file contains `option: value` pairs, one per line. When making changes to this file, you can use any text editor that saves the file in a format that is native to the environment you are using.

Any setting that has the options commented out with a number sign (`#`) uses the default setting.

The following options are recognized:

httpPort

Purpose

Specifies the local port value that `sfcbd` should listen to receive HTTP (insecure) requests from CIM clients. Default is `5988`.

Syntax

```
httpPort: port_number
```

enableHttp

Purpose

Specifies whether SFCB should accept HTTP client connections. Default is `false`.

Syntax

```
enableHttp: option
```

Option	Description
true	Enables HTTP connections.
false	Disables HTTP connections.

httpProcs

Purpose

Specifies the maximum number of simultaneous HTTP client connections before new incoming HTTP requests are blocked. Default is 8 .

Syntax

`httpProcs: max_number_of_connections`

httpsPort

Purpose

Specifies the local port value where sfcbd listens for HTTPS requests from CIM clients. Default is 5989 .

Syntax

`httpsPort: port_number`

enableHttps

Purpose

Specifies if SFCB will accept HTTPS client connections. Default is true .

Syntax

`enableHttps: option`

Option	Description
true	Enables HTTPS connections.
false	Disables HTTPS connections.

httpsProcs

Purpose

Specifies the maximum number of simultaneous HTTPS client connections before new incoming HTTPS requests are blocked. Default is 8 .

Syntax

`httpsProcs: max_number_of_connections`

enableInterOp

Purpose

Specifies if SFCB will provide the *interop* namespace for indication support. Default is `true` .

Syntax

`enableInterOp: option`

Option	Description
true	Enables interop namespace.

Option	Description
false	Disables interop namespace.

provProcs

Purpose

Specifies the maximum number of simultaneous provider processes. After this point, if a new incoming request requires loading a new provider, then one of the existing providers will first be automatically unloaded. Default is 32 .

Syntax

`provProcs: max_number_of_procs`

doBasicAuth

Purpose

Switches basic authentication on or off based on the client user identifier before it accepts the request. Default value is `true` which means that basic client authentication is performed.

Syntax

`doBasicAuth: option`

Option	Description
true	Enables basic authentication.
false	Disables basic authentication.

basicAuthLib

Purpose

Specifies the local library name. The SFCB server loads the library to authenticate the client user identifier. Default is `sfcBasicPAMAuthentication`.

Syntax

`provProcs: max_number_of_procs`

useChunking

Purpose

This option switches the use of HTTP/HTTPS “chunking” on or off. If switched on, the server will return large volumes of response data to the client in smaller “chunks”, rather than to buffer the data and send it back all in one chunk. Default is `true`.

Syntax

`useChunking: option`

Option	Description
true	Enables HTTP/HTTPS data chunking.
false	Disables HTTP/HTTPS data chunking.

keepaliveTimeout

Purpose

Specifies the maximum time in seconds that SFCB HTTP process waits between two requests on one connection before it terminates. Setting it to 0 disables HTTP keep-alive. Default is 0.

Syntax

`keepaliveTimeout: secs`

keepaliveMaxRequest

Purpose

Specifies the maximum number of consecutive requests on one connection. Setting it to 0 disables HTTP keep-alive. Default value is 10 .

Syntax

`keepaliveMaxRequest: number_of_connections`

registrationDir

Purpose

Specifies the registration directory, which contains the provider registration data, the staging area, and the static repository. Default is `/var/lib/sfcb/registration` .

Syntax

`registrationDir: dir`

providerDirs

Purpose

Specifies a space-separated list of directories where SFCB is searching for provider libraries. Default is `/usr/lib64 /usr/lib64 /usr/lib64/cmpi`.

Syntax

`providerDirs: dir`

providerSampleInterval

Purpose

Specifies the interval in seconds at which the provider manager is checking for idle providers. Default is 30.

Syntax

```
providerSampleInterval: secs
```

providerTimeoutInterval

Purpose

Specifies the interval in seconds before an idle provider gets unloaded by the provider manager. Default is 60.

Syntax

```
providerTimeoutInterval: secs
```

providerAutoGroup

Purpose

If the provider registration file does not specify any other group, and the option is set to *true*, all providers in the same shared library are executed in the same process.

Syntax

```
providerAutoGroup: option
```

Option	Description
true	Enables grouping of providers.

Option	Description
false	Disables grouping of providers.

sslCertificateFilePath

Purpose

Specifies the name of the file that contains the server certificate. The file must be in PEM (Privacy Enhanced Mail, RFC 1421 and RFC 1424) format. This file is only required if `enableHttps` is set to `true`. Default is `/etc/sfcb/server.pem`.

Syntax

```
sslCertificateFilePath: path
```

sslKeyFilePath

Purpose

Specifies the name of the file that contains the private key for the server certificate. The file must be in PEM format and may not be protected by passphrase. This file is only required if `enableHttps` is set to `true`. Default is `/etc/sfcb/file.pem`.

Syntax

```
sslKeyFilePath: path
```

sslClientTrustStore

Purpose

Specifies the name of the file that contains either the CA or self-signed certificates of the clients. This file must be in PEM format and is only required if `sslClientCertificate` is set to `accept` or `require`. Default is `/etc/sfcb/client.pem`.

Syntax

`sslClientTrustStore: path`

sslClientCertificate

Purpose

Specifies the way SFCB handles client certificate based authentication. If set to `ignore`, it will not request a certificate from the client. If set to `accept` it will request a certificate from the client but will not fail if the client does not present one. If set to `require`, it will refuse the client connection if the client does not present a certificate. Default value is `ignore`.

Syntax

`sslClientCertificate: option`

Option	Description
<code>ignore</code>	Disables requesting a client certificate.
<code>accept</code>	Disables requesting a client certificate. Will not fail if no certificate is present.
<code>require</code>	Refuses the client connection without a valid certificate.

certificateAuthLib

Purpose

Specifies the name of the local library to request for the user authentication based on client certificate. This is only requested if `sslClientCertificate` is not set to `ignore`. Default value is `sfcCertificateAuthentication`.

Syntax

`certificateAuthLib: file`

traceLevel

Purpose

Specifies the trace level for SFCB. You can override it by setting environment variable `SFCB_TRACE_LEVEL`. Default value is 0.

Syntax

`traceLevel: num_level`

traceMask

Purpose

Specifies the trace mask for SFCB. you can override it by the command line option `--trace-components`. Default value is 0.

Syntax

`traceMask: mask`

traceFile

Purpose

Specifies the trace file for SFCB. You can override it by setting environment variable `SFCB_TRACE_FILE`. Default value is `stderr` (standard error output).

Syntax

`traceFile: output`

30.4 Advanced SFCB Tasks

This chapter covers more advanced topics related to SFCB usage. To understand them, you need to have basic knowledge of the Linux file system and experience with the Linux command line. This chapter includes the following tasks:

- Installing CMPI providers
- Testing SFCB
- Using `wbemcli` CIM client

30.4.1 Installing CMPI Providers

To install a CMPI provider, you need to make sure that its shared library is copied into one of the directories specified by `providerDirs` configuration option, see Section “`providerDirs`” (page 470) . The provider must also be properly registered using `sfcbstage` and `sfcbrepos` commands.

The provider package is usually prepared for SFCB, so that its installation takes care of the proper registration. Most SBLIM providers are prepared for SFCB.

Class Repository

Class repository is a place where SFCB stores information about CIM classes. It usually consists of a directory tree comprised of namespace components. Typical CIM namespaces are `root/cimv2` or `root/interop`, which respectively translate to the class repository directory path on the filesystem

```
/var/lib/sfcb/registration/repository/root/cimv2
```

and

```
/var/lib/sfcb/registration/repository/root/interop
```

Each namespace directory contains the file `classSchemas`. The file has a compiled binary representation of all the CIM classes registered under that namespace. It also contains necessary information about their CIM superclasses.

In addition, each namespace directory may contain a file `qualifiers` which contains all qualifiers for the namespace. When `sfcbd` restarts, the class provider will scan the directory `/var/lib/sfcb/registration/repository/` and all its subdirectories to determine the registered namespaces. Then `classSchemas` files are decoded and the class hierarchy for each namespace is built.

Adding New Classes

SFCB cannot make live CIM class manipulations. You need to add, change or remove classes offline and restart SFCB service with `rcsfcb restart` to register the changes.

To store providers class and registration information, SFCB uses a place called *staging area*. On SUSE® Linux Enterprise Server systems, it is the directory structure under `/var/lib/sfcb/stage/`.

To add a new provider, you need to:

- Copy the provider class definition files to the `./mofs` subdirectory of staging area directory (`/var/lib/sfcb/stage/mofs`).
- Copy a registration file which contains the name of the class or classes and type of provider, and the name of the executable library file into the `./regs` subdirectory.

There are two default “mof” (class definition) files in the staging directory: `indication.mof` and `interop.mof`. MOF files under the root stage directory `/var/lib/sfcb/stage/mofs` will be copied into each namespace after running `sfcbrepos` command. The `interop.mof` will only be compiled into the *interop* namespace.

The directory layout may look like the following example:

```
tux@mercury:~> ls /var/lib/sfcb/stage
default.reg  mofs  regs

tux@mercury:~> ls /var/lib/sfcb/stage/mofs
indication.mof  root

tux@mercury:~> ls /var/lib/sfcb/stage/mofs/root
cimv2  interop  suse  virt

tux@mercury:~> ls -l /var/lib/sfcb/stage/mofs/root/cimv2 | less
Linux_ABIPParameter.mof
```

```

Linux_BaseIndication.mof
Linux_Base.mof
Linux_DHCPElementConformsToProfile.mof
Linux_DHCPEntity.mof
[...]
OMC_StorageSettingWithHints.mof
OMC_StorageVolumeDevice.mof
OMC_StorageVolume.mof
OMC_StorageVolumeStorageSynchronized.mof
OMC_SystemStorageCapabilities.mof

tux@mercury:~> ls -l /var/lib/sfcb/stage/mofs/root/interop
ComputerSystem.mof
ElementConformsToProfile.mof
HostSystem.mof
interop.mof
Linux_DHCPElementConformsToProfile.mof
[...]
OMC_SMIElementSoftwareIdentity.mof
OMC_SMIsubProfileRequiresProfile.mof
OMC_SMIVolumeManagementSoftware.mof
ReferencedProfile.mof
RegisteredProfile.mof

tux@mercury:~> ls -l /var/lib/sfcb/stage/regs
AllocationCapabilities.reg
Linux_ABIPParameter.reg
Linux_BaseIndication.reg
Linux_DHCPGlobal.reg
Linux_DHCPRegisteredProfile.reg
[...]
OMC_Base.sfcb.reg
OMC_CopyServices.sfcb.reg
OMC_PowerManagement.sfcb.reg
OMC_Server.sfcb.reg
RegisteredProfile.reg

tux@mercury:~> cat /var/lib/sfcb/stage/regs/Linux_DHCPRegisteredProfile.reg
[Linux_DHCPRegisteredProfile]
  provider: Linux_DHCPRegisteredProfileProvider
  location: cmpiLinux_DHCPRegisteredProfile
  type: instance
  namespace: root/interop
#
[Linux_DHCPElementConformsToProfile]
  provider: Linux_DHCPElementConformsToProfileProvider
  location: cmpiLinux_DHCPElementConformsToProfile
  type: instance association
  namespace: root/cimv2
#
[Linux_DHCPElementConformsToProfile]
  provider: Linux_DHCPElementConformsToProfileProvider
  location: cmpiLinux_DHCPElementConformsToProfile
  type: instance association

```

```
namespace: root/interop
#
```

SFCB uses a custom provider registration file for each provider.

NOTE: SBLIM providers registration files

All SBLIM providers on the SBLIM Web site already include a registration file that is used to generate the .reg file for SFCB.

The format of SFCB registration file is:

```
[<class-name>]
  provider: <provide-name>
  location: <library-name>
  type: [instance] [association] [method] [indication]
  group: <group-name>
  unload: never
  namespace: <namespace-for-class> ...
```

where:

<class-name>

The CIM class name (required)

<provider-name>

The CMPI provider name (required)

<location-name>

The name of the provider library (required)

type

The type of the provider (required). This can be any combination of: instance, association, method or indication.

<group-name>

Multiple providers can be grouped together and run under a single process to further minimize runtime resources. All providers registered under the same <group-name> will be executed under the same process. By default each provider will be run as a separate process.

`unload`

Specifies the unload policy for the provider. Currently the only supported option is `never`, which specifies that the provider will not be monitored for idle times and will never be unloaded. By default each provider will be unloaded when its idle times exceed the value specified in the configuration file.

`namespace`

List of namespaces for which this provider can be executed. This is required, although for most providers this will be `root/cimv2`.

Once all the class definitions and provider registration files are stored in the staging area, you need to rebuild the SFCB class repository with the command `sfcbrepos -f`.

You can add, change or remove classes this way. After rebuilding the class repository, restart SFCB with command `rcsfcb restart`.

Alternatively, the SFCB package contains a utility that will copy provider class mof files and registration files to the correct locations in the staging area.

```
sfcbstage -r [provider.reg] [class1.mof] [class2.mof] ...
```

After running this command you still need to rebuild the class repository and restart SFCB service.

30.4.2 Testing SFCB

The SFCB package includes two testing scripts: `wbemcat` and `xmltest`.

`wbemcat` sends raw CIM-XML data via HTTP protocol to the specified SFCB host (localhost by default) listening on port 5988. Then it displays the returned results. The following file contains the CIM-XML representation of a standard `EnumerateClasses` request:

```
<?xml version="1.0" encoding="utf-8"?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
  <MESSAGE ID="4711" PROTOCOLVERSION="1.0">
    <SIMPLEREQ>
      <IMETHODCALL NAME="EnumerateClasses">
        <LOCALNAMESPACEPATH>
          <NAMESPACE NAME="root"/>
        </LOCALNAMESPACEPATH>
      </IMETHODCALL>
    </SIMPLEREQ>
  </MESSAGE>
</CIM>
```

```

        <NAMESPACE NAME="cimv2"/>
    </LOCALNAMESPACEPATH>
    <IPARAMVALUE NAME="ClassName">
        <CLASSNAME NAME=""/>
    </IPARAMVALUE>
    <IPARAMVALUE NAME="DeepInheritance">
        <VALUE>TRUE</VALUE>
    </IPARAMVALUE>
    <IPARAMVALUE NAME="LocalOnly">
        <VALUE>FALSE</VALUE>
    </IPARAMVALUE>
    <IPARAMVALUE NAME="IncludeQualifiers">
        <VALUE>FALSE</VALUE>
    </IPARAMVALUE>
    <IPARAMVALUE NAME="IncludeClassOrigin">
        <VALUE>TRUE</VALUE>
    </IPARAMVALUE>
</IMETHODCALL>
<?xml version="1.0" encoding="utf-8"?>
1743 <CIM CIMVERSION="2.0" DTDVERSION="2.0">
1744   <MESSAGE ID="4711" PROTOCOLVERSION="1.0">
1745     <SIMPLEREQ>
1746       <IMETHODCALL NAME="EnumerateClasses">
1747         <LOCALNAMESPACEPATH>
1748           <NAMESPACE NAME="root"/>
1749           <NAMESPACE NAME="cimv2"/>
1750         </LOCALNAMESPACEPATH>
1751         <IPARAMVALUE NAME="ClassName">
1752           <CLASSNAME NAME=""/>
1753         </IPARAMVALUE>
1754         <IPARAMVALUE NAME="DeepInheritance">
1755           <VALUE>TRUE</VALUE>
1756         </IPARAMVALUE>
1757         <IPARAMVALUE NAME="LocalOnly">
1758           <VALUE>FALSE</VALUE>
1759         </IPARAMVALUE>
1760         <IPARAMVALUE NAME="IncludeQualifiers">
1761           <VALUE>FALSE</VALUE>
1762         </IPARAMVALUE>
1763         <IPARAMVALUE NAME="IncludeClassOrigin">
1764           <VALUE>TRUE</VALUE>
1765         </IPARAMVALUE>
1766       </IMETHODCALL>
1767     </SIMPLEREQ>
1768   </MESSAGE>
1769 </CIM></SIMPLEREQ>
  </MESSAGE>
</CIM>

```

Sending this request to SFCB CIMOM returns a list of all supported classes for which there is a registered provider. Suppose you save the file as `cim_xml_test.xml`.

```
tux@mercury:~> wbemcat cim_xml_test.xml | less
HTTP/1.1 200 OK
Content-Type: application/xml; charset="utf-8"
Content-Length: 337565
Cache-Control: no-cache
CIMOperation: MethodResponse

<?xml version="1.0" encoding="utf-8" ?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
<MESSAGE ID="4711" PROTOCOLVERSION="1.0">
<SIMPLERSP>
<IMETHODRESPONSE NAME="EnumerateClasses">
[... ]
<CLASS NAME="Linux_DHCPParamsForEntity" SUPERCLASS="CIM_Component">
<PROPERTY.REFERENCE NAME="GroupComponent" REFERENCECLASS="Linux_DHCPEntity">
</PROPERTY.REFERENCE>
<PROPERTY.REFERENCE NAME="PartComponent" REFERENCECLASS="Linux_DHCPParams">
</PROPERTY.REFERENCE>
</CLASS>
</IRETURNVALUE>
</IMETHODRESPONSE>
</SIMPLERSP>
</MESSAGE>
</CIM>
```

The classes listed will vary depending on what providers are installed on your system.

The second script `xmltest` is also used to send a raw CIM-XML test file to the SFCB CIMOM. It then compares the returned results against a previously saved “OK” result file. If there does not yet exist a corresponding “OK” file, it will be created for later use:

```
tux@mercury:~> xmltest cim_xml_test.xml
Running test cim_xml_test.xml ... OK
Saving response as cim_xml_test.OK
tux@mercury:~> xmltest cim_xml_test.xml
Running test cim_xml_test.xml ... Passed
```

30.4.3 Command Line CIM Client: `wbemcli`

In addition to `wbemcat` and `xmltest`, the SBLIM project includes a more advanced command line CIM client `wbemcli`. The client is used to send CIM requests to SFCB server and display returned results. It is independent of CIMOM library and can be used with all WBEM compliant implementations.

For example, if you need to list all the classes implemented by SBLIM providers registered to your SFCB, send the “EnumerateClasses” (ec) request to SFCB:

```

tux@mercury:~> wbemcli -dx ec http://localhost/root/cimv2
To server: <?xml version="1.0" encoding="utf-8" ?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
<MESSAGE ID="4711" PROTOCOLVERSION="1.0"><SIMPLEREQ><IMETHODCALL \
  NAME="EnumerateClasses"><LOCALNAMESPACEPATH><NAMESPACE NAME="root"> \
    </NAMESPACE><NAMESPACE NAME="cimv2"></NAMESPACE> \
  </LOCALNAMESPACEPATH>
<IPARAMVALUE NAME="DeepInheritance"><VALUE>TRUE</VALUE> \
  </IPARAMVALUE>
<IPARAMVALUE NAME="LocalOnly"><VALUE>FALSE</VALUE></IPARAMVALUE>
<IPARAMVALUE NAME="IncludeQualifiers"><VALUE>FALSE</VALUE> \
  </IPARAMVALUE>
<IPARAMVALUE NAME="IncludeClassOrigin"><VALUE>TRUE</VALUE> \
  </IPARAMVALUE>
</IMETHODCALL></SIMPLEREQ>
</MESSAGE></CIM>
From server: Content-Type: application/xml; charset="utf-8"
From server: Content-Length: 337565
From server: Cache-Control: no-cache
From server: CIMOperation: MethodResponse
From server: <?xml version="1.0" encoding="utf-8" ?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
<MESSAGE ID="4711" PROTOCOLVERSION="1.0">
<SIMPLERSP>
<IMETHODRESPONSE NAME="EnumerateClasses">
<RETURNVALUE>
<CLASS NAME="CIM_ResourcePool" SUPERCLASS="CIM_LogicalElement">
<PROPERTY NAME="Generation" TYPE="uint64">
</PROPERTY>
<PROPERTY NAME="ElementName" TYPE="string">
</PROPERTY>
<PROPERTY NAME="Description" TYPE="string">
</PROPERTY>
<PROPERTY NAME="Caption" TYPE="string">
</PROPERTY>
<PROPERTY NAME="InstallDate" TYPE="datetime">
</PROPERTY>
[.]
<CLASS NAME="Linux_ReiserFileSystem" SUPERCLASS="CIM_UnixLocalFileSystem">
<PROPERTY NAME="FSReservedCapacity" TYPE="uint64">
</PROPERTY>
<PROPERTY NAME="TotalInodes" TYPE="uint64">
</PROPERTY>
<PROPERTY NAME="FreeInodes" TYPE="uint64">
</PROPERTY>
<PROPERTY NAME="ResizeIncrement" TYPE="uint64">
<VALUE>0</VALUE>
</PROPERTY>
<PROPERTY NAME="IsFixedSize" TYPE="uint16">
<VALUE>0</VALUE>
</PROPERTY>
[.]

```

The `-dx` option shows you the actual XML send to SFCB by `wbemcli` as well as the actual XML received. In the above example, the first of many returned classes was `CIM_ResourcePool` followed by `Linux_ReiserFileSystem`. Similar entries will appear for all of the other registered classes.

If you omit the `-dx` option, `wbemcli` will display only a compact representation of the returned data:

```
tux@mercury:~> wbemcli ec http://localhost/root/cimv2
localhost:5988/root/cimv2:CIM_ResourcePool Generation=, ElementName=, \
  Description=, Caption=, InstallDate=, Name=, OperationalStatus=, \
  StatusDescriptions=, Status=, HealthState=, PrimaryStatus=, \
  DetailedStatus=, OperatingStatus=, CommunicationStatus=, InstanceID=, \
  PoolID=, Primordial=, Capacity=, Reserved=, ResourceType=, \
  OtherResourceType=, ResourceSubType=, \AllocationUnits=
localhost:5988/root/cimv2:Linux_ReiserFileSystem FSReservedCapacity=, \
  TotalInodes=, FreeInodes=, ResizeIncrement=, IsFixedSize=, NumberOfFiles=, \
  OtherPersistenceType=, PersistenceType=, FileSystemType=, ClusterSize=, \
  MaxFileNameLength=, CodeSet=, CasePreserved=, CaseSensitive=, \
  CompressionMethod=, EncryptionMethod=, ReadOnly=, AvailableSpace=, \
  FileSystemSize=, BlockSize=, Root=, Name=, CreationClassName=, CSName=, \
  CSCreationClassName=, Generation=, ElementName=, Description=, Caption=, \
  InstanceID=, InstallDate=, OperationalStatus=, StatusDescriptions=, \
  Status=, HealthState=, PrimaryStatus=, DetailedStatus=, OperatingStatus= \
  , CommunicationStatus=, EnabledState=, OtherEnabledState=, RequestedState= \
  , EnabledDefault=, TimeOfLastStateChange=, AvailableRequestedStates=, \
  TransitioningToState=, PercentageSpaceUse=
[...]
```

30.5 For More Details

For more details about WBEM and SFCB, see the following sources:

<http://www.dmtf.org>

Distributed Management Task Force Web site

<http://www.dmtf.org/standards/wbem/>

Web-Based Enterprise Management (WBEM) Web site

<http://www.dmtf.org/standards/cim/>

Common Information Model (CIM) Web site

<http://sblim.wiki.sourceforge.net/>

Standards Based Linux Instrumentation (SBLIM) Web site

<http://sblim.wiki.sourceforge.net/Sfcb>
Small Footprint CIM Broker (SFCB) Web site

<http://sblim.wiki.sourceforge.net/Providers>
SBLIM providers packages

Part V. Troubleshooting

Help and Documentation

SUSE® Linux Enterprise Server comes with various sources of information and documentation, many of which are already integrated into your installed system:

Documentation in `/usr/share/doc`

This traditional help directory holds various documentation files and release notes for your system. Find more detailed information in Section 31.1, “Documentation Directory” (page 488).

Man Pages and Info Pages for Shell Commands

When working with the shell, you do not need to know the options of the commands by heart. Traditionally, the shell provides integrated help by means of man pages and info pages. Read more in Section 31.2, “Man Pages” (page 490) and Section 31.3, “Info Pages” (page 491).

Desktop Help Centers

The help centers of both the KDE desktop (KDE help center) and the GNOME desktop (Yelp) provide central access to the most important documentation resources on your system in searchable form. These resources include online help for installed applications, man pages, info pages, and the Novell/SUSE manuals delivered with your product.

Separate Help Packages for Some Applications

When installing new software with YaST, the software documentation is installed automatically (in most cases) and usually appears in the help center of your desktop. However, some applications, such as GIMP, may have different online help packages that can be installed separately with YaST and do not integrate into the help centers.

31.1 Documentation Directory

The traditional directory to find documentation on your installed Linux system is `/usr/share/doc`. Usually, the directory contains information about the packages installed on your system, plus release notes, manuals, and more.

NOTE: Contents Depends on Installed Packages

In the Linux world, many manuals and other kinds of documentation are available in the form of packages, just like software. How much and which information you find in `/usr/share/docs` also depends on the (documentation) packages installed. If you cannot find the subdirectories mentioned here, check if the respective packages are installed on your system and add them with YaST, if needed.

31.1.1 Novell/SUSE Manuals

We provide HTML and PDF versions of our books in different languages. In the `manual` subdirectory, find HTML versions of most of the Novell/SUSE manuals available for your product. For an overview of all documentation available for your product refer to the preface of the manuals.

If more than one language is installed, `/usr/share/doc/manual` may contain different language versions of the manuals. The HTML versions of the Novell/SUSE manuals are also available in the help center of both desktops. For information on where to find the PDF and HTML versions of the books on your installation media, refer to the SUSE Linux Enterprise Server Release Notes. They are available on your installed system under `/usr/share/doc/release-notes/` or online at your product-specific Web page at <http://www.novell.com/documentation/>.

31.1.2 HOWTOs

If the `howto` package is installed on your system, `/usr/share/doc` also holds the `howto` subdirectory, where you find additional documentation for many tasks relating to the setup and operation of Linux software.

31.1.3 Package Documentation

Under `packages`, find the documentation that is included in the software packages installed on your system. For every package, a subdirectory `/usr/share/doc/packages/packagename` is created. It often contains `README` files for the package and sometimes examples, configuration files, or additional scripts. The following list introduces typical files to be found under `/usr/share/doc/packages`. None of these entries are mandatory and many packages might just include a few of them.

AUTHORS

List of the main developers.

BUGS

Known bugs or malfunctions. Might also contain a link to a Bugzilla Web page where you can search all bugs.

CHANGES , ChangeLog

Summary of changes from version to version. Usually interesting for developers, because it is very detailed.

COPYING , LICENSE

Licensing information.

FAQ

Question and answers collected from mailing lists or newsgroups.

INSTALL

How to install this package on your system. As the package is already installed by the time you get to read this file, you can safely ignore the contents of this file.

README, README.*

General information on the software. For example, for what purpose and how to use it.

TODO

Things that are not implemented yet, but probably will be in the future.

MANIFEST

List of files with a brief summary.

NEWS
Description of what is new in this version.

31.2 Man Pages

Man pages are an essential part of any Linux system. They explain the usage of a command and all available options and parameters. Man pages can be accessed with `man` followed by the name of the command, for example, `man ls`.

Man pages are displayed directly in the shell. To navigate them, move up and down with **Page ↑** and **Page ↓**. Move between the beginning and the end of a document with **Home** and **End**. End this viewing mode by pressing **Q**. Learn more about the `man` command itself with `man man`. Man pages are sorted in categories as shown in Table 31.1, “Man Pages—Categories and Descriptions” (page 490) (taken from the `man` page for `man` itself).

Table 31.1 *Man Pages—Categories and Descriptions*

Number	Description
1	Executable programs or shell commands
2	System calls (functions provided by the kernel)
3	Library calls (functions within program libraries)
4	Special files (usually found in <code>/dev</code>)
5	File formats and conventions (<code>/etc/fstab</code>)
6	Games
7	Miscellaneous (including macro packages and conventions), for example, <code>man(7)</code> , <code>groff(7)</code>
8	System administration commands (usually only for <code>root</code>)

Number	Description
9	Kernel routines (nonstandard)

Each man page consists of several parts labeled *NAME* , *SYNOPSIS* , *DESCRIPTION* , *SEE ALSO* , *LICENSING* , and *AUTHOR* . There may be additional sections available depending on the type of command.

31.3 Info Pages

Info pages are another important source of information on your system. Usually, they are more detailed than man pages. To view the info page for a certain command, enter `info` followed by the name of the command, for example, `info ls`. You can browse an info page with a viewer directly in the shell and display the different sections, called “nodes.” Use `Space` to move forward and `<—` to move backwards. Within a node, you can also browse with `Page ↑` and `Page ↓` but only `Space` and `<—` will take you also to the previous or subsequent node. Press `Q` to end the viewing mode. Not every man page comes with an info page and vice versa.

Common Problems and Their Solutions

32

This chapter describes a range of potential problems and their solutions. Even if your situation is not precisely listed here, there may be one similar enough to offer hints to the solution of your problem.

32.1 Finding and Gathering Information

Linux reports things in a very detailed way. There are several places to look when you encounter problems with your system, most of which are standard to Linux systems in general, and some of which are relevant to SUSE Linux Enterprise Server systems. Most log files can be viewed with YaST (*Miscellaneous > Start-Up Log*).

YaST offers the possibility to collect all system information needed by the support team. Use *Miscellaneous > Support Query* and select the problem category. When all information is gathered, attach it to your support request.

A list of the most frequently checked log files follows with the description of their typical purpose. Paths containing ~ refer to the current user's home directory.

Table 32.1 *Log Files*

Log File	Description
<code>~/.xsession-errors</code>	Messages from the desktop applications currently running.
<code>/var/log/apparmor/</code>	Log files from AppArmor, see Part “Confining Privileges with Novell AppArmor” (↑ <i>Security Guide</i>) for detailed information.
<code>/var/log/audit/audit.log</code>	Log file from Audit to track any access to files, directories, or resources of your system, and trace system calls.
<code>/var/log/boot.msg</code>	Messages from the kernel reported during the boot process.
<code>/var/log/mail.*</code>	Messages from the mail system.
<code>/var/log/messages</code>	Ongoing messages from the kernel and system log daemon (when running).
<code>/var/log/NetworkManager</code>	Log file from NetworkManager to collect problems with network connectivity
<code>/var/log/samba/</code>	Directory containing Samba server and client log messages.
<code>/var/log/SaX.log</code>	Hardware messages from the SaX display and KVM system.
<code>/var/log/warn</code>	All messages from the kernel and system log daemon with the “warning” level or higher.
<code>/var/log/wtmp</code>	Binary file containing user login records for the current machine session. View it with <code>last</code> .

Log File	Description
<code>/var/log/Xorg.*.log</code>	Various start-up and runtime logs from the X Window system. It is useful for debugging failed X start-ups.
<code>/var/log/YaST2/</code>	Directory containing YaST's actions and their results.
<code>/var/log/zypper.log</code>	Log file of zypper.

Apart from log files, your machine also supplies you with information about the running system. See Table 32.2: System Information With the `/proc` File System

Table 32.2 *System Information With the `/proc` File System*

File	Description
<code>/proc/cpuinfo</code>	Contains processor information, including its type, make, model, and performance.
<code>/proc/dma</code>	Shows which DMA channels are currently being used.
<code>/proc/interrupts</code>	Shows which interrupts are in use, and how many of each have been in use.
<code>/proc/iomem</code>	Displays the status of I/O (input/output) memory.
<code>/proc/ioports</code>	Shows which I/O ports are in use at the moment.
<code>/proc/meminfo</code>	Displays memory status.
<code>/proc/modules</code>	Displays the individual modules.
<code>/proc/mounts</code>	Displays devices currently mounted.
<code>/proc/partitions</code>	Shows the partitioning of all hard disks.

File	Description
<code>/proc/version</code>	Displays the current version of Linux.

Apart from the `/proc` file system, the Linux kernel exports information with the `sysfs` module, an in-memory filesystem. This module represents kernel objects, their attributes and relationships. For more information about `sysfs`, see the context of `udev` in Chapter 11, *Dynamic Kernel Device Management with udev* (page 135). Table 32.3 contains an overview of the most common directories under `/sys`.

Table 32.3 *System Information With the `/sys` File System*

File	Description
<code>/sys/block</code>	Contains subdirectories for each block device discovered in the system. Generally, these are mostly disk type devices.
<code>/sys/bus</code>	Contains subdirectories for each physical bus type.
<code>/sys/class</code>	Contains subdirectories grouped together as a functional types of devices (like graphics, net, printer, etc.)
<code>/sys/device</code>	Contains the global device hierarchy.

Linux comes with a number of tools for system analysis and monitoring. See Chapter 2, *System Monitoring Utilities* (↑*System Analysis and Tuning Guide*) for a selection of the most important ones used in system diagnostics.

Each of the following scenarios begins with a header describing the problem followed by a paragraph or two offering suggested solutions, available references for more detailed solutions, and cross-references to other scenarios that are related.

32.2 Installation Problems

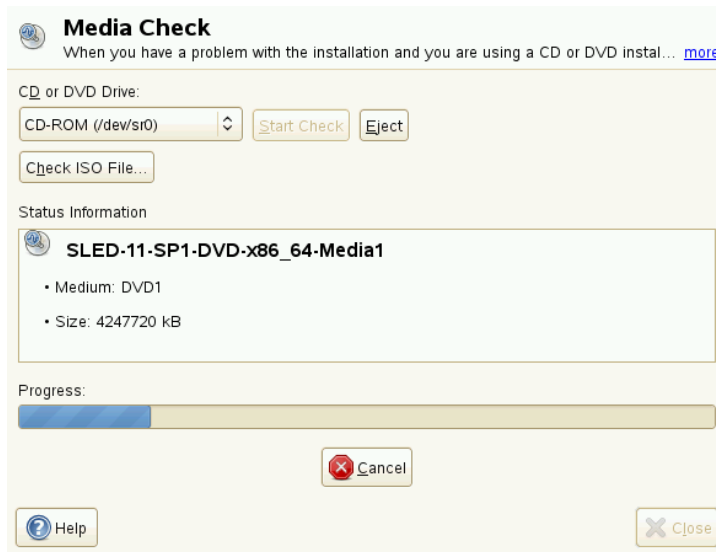
Installation problems are situations when a machine fails to install. It may fail entirely or it may not be able to start the graphical installer. This section highlights some of the

typical problems you may run into, and offers possible solutions or workarounds for these kinds of situations.

32.2.1 Checking Media

If you encounter any problems using the SUSE Linux Enterprise Server installation media, check the integrity of your installation media with *Software > Media Check*. Media problems are more probable with the media you burn yourself. To check the SUSE Linux Enterprise Server medium, insert it into the drive and click *Start Check* in the *Media Check* screen of YaST. This may take several minutes. If errors are detected, do not use this medium for installation.

Figure 32.1 *Checking Media*

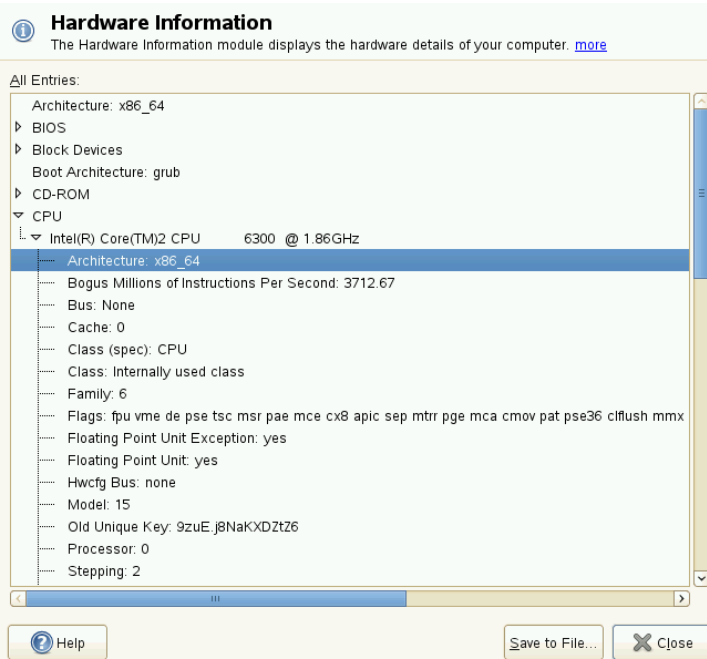


32.2.2 Hardware Information

Display detected hardware and technical data using *Hardware > Hardware Information*. Click any node of the tree for more information about a device. This module is especially useful, when submitting a support request for which you need information about your hardware.

Save the displayed hardware information to a file by clicking *Save to File*. Select the desired directory and filename then click *Save* to create the file.

Figure 32.2 *Displaying Hardware Information*



32.2.3 No Bootable DVD Drive Available

If your computer does not contain a bootable DVD-ROM drive or if the one you have is not supported by Linux, there are several options you can install your machine without a built-in DVD drive:

Booting from a Floppy Disk

Create a boot floppy and boot from floppy disk instead of DVD.

Using an External Boot Device

If it is supported by your BIOS and the installation kernel, boot from external DVD drives.

Network Boot via PXE

If a machine lacks a DVD drive, but provides a working ethernet connection, perform a completely network-based installation. See Section “Remote Installation via VNC—PXE Boot and Wake on LAN” (Chapter 14, *Remote Installation*, ↑*Deployment Guide*) and Section “Remote Installation via SSH—PXE Boot and Wake on LAN” (Chapter 14, *Remote Installation*, ↑*Deployment Guide*) for details.

Booting from a Floppy Disk (SYSLINUX)

On some older computers, there is no bootable DVD drive available, but there is a floppy disk drive present. To install on such a system, create boot disks and boot your system with them.

The boot disks include the loader called SYSLINUX and the program linuxrc. SYSLINUX enables the selection of a kernel during the boot procedure and the specification of any parameters needed for the hardware used. The program linuxrc supports the loading of kernel modules for your hardware and subsequently starts the installation.

When booting from a boot disk, the boot procedure is initiated by the boot loader SYSLINUX (package `syslinux`). When the system is booted, SYSLINUX runs a minimum hardware detection that mainly consists of the following steps:

1. The program checks if the BIOS provides VESA 2.0–compliant framebuffer support and boots the kernel accordingly.
2. The monitor data (DDC info) is read.
3. The first block of the first hard disk (MBR) is read to map BIOS IDs to Linux device names during the boot loader configuration. The program attempts to read the block by means of the the lba32 functions of the BIOS to determine if the BIOS supports these functions.

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

```
verbose 1
```

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

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

External Boot Devices

Linux supports most existing DVD drives. If the system has neither a DVD drive nor a floppy disk, it is still possible that an external DVD drive, connected through USB, FireWire, or SCSI, can be used to boot the system. This depends mainly on the interaction of the BIOS and the hardware used. Sometimes a BIOS update may help if you encounter problems.

32.2.4 Booting from Installation Media Fails

One reason why a machine does not boot the installation media can be an incorrect boot sequence setting in BIOS. The BIOS boot sequence must have DVD drive set as the first entry for booting. Otherwise the machine would try to boot from another medium, typically the hard disk. Guidance for changing the BIOS boot sequence can be found in the documentation provided with your motherboard, or in the following paragraphs.

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

Procedure 32.1 *Changing the BIOS Boot Sequence*

- 1** Enter the BIOS using the proper key as announced by the boot routines and wait for the BIOS screen to appear.
- 2** To change the boot sequence in an AWARD BIOS, look for the *BIOS FEATURES SETUP* entry. Other manufacturers may have a different name for this, such as *ADVANCED CMOS SETUP*. When you have found the entry, select it and confirm with Enter.
- 3** In the screen that opens, look for a subentry called *BOOT SEQUENCE* or *BOOT ORDER*. The boot sequence looks something like C, A or A, C. In the former case, the machine first searches the hard disk (C) then the floppy drive (A) to find a bootable medium. Change the settings by pressing PgUp or PgDown until the sequence is A, CDROM, C.
- 4** Leave the BIOS setup screen by pressing Esc. To save the changes, select *SAVE & EXIT SETUP*, or press F10. To confirm that your settings should be saved, press Y.

Procedure 32.2 *Changing the Boot Sequence in a SCSI BIOS (Adaptec Host Adapter)*

- 1** Open the setup by pressing Ctrl + A.
- 2** Select *Disk Utilities*. The connected hardware components are now displayed.
Make note of the SCSI ID of your DVD drive.
- 3** Exit the menu with Esc.
- 4** Open *Configure Adapter Settings*. Under *Additional Options*, select *Boot Device Options* and press Enter.
- 5** Enter the ID of the DVD drive and press Enter again.
- 6** Press Esc twice to return to the start screen of the SCSI BIOS.
- 7** Exit this screen and confirm with *Yes* to boot the computer.

Regardless of what language and keyboard layout your final installation will be using, most BIOS configurations use the US keyboard layout as depicted in the following figure:

Figure 32.3 *US Keyboard Layout*



32.2.5 Fails to Boot

Some hardware types, mainly very old or very recent ones, fail to install. In many cases, this may happen because support for this type of hardware is missing in the installation kernel, or due to certain functionality included in this kernel, such as ACPI, that can still cause problems on some hardware.

If your system fails to install using the standard *Installation* mode from the first installation boot screen, try the following:

- 1 With the DVD still in the drive, reboot the machine with Ctrl + Alt + Del or using the hardware reset button.
- 2 When the boot screen appears, press F5, use the arrow keys of your keyboard to navigate to *No ACPI* and press Enter to launch the boot and installation process. This option disables the support for ACPI power management techniques.
- 3 Proceed with the installation as described in Chapter 6, *Installation with YaST* (*Deployment Guide*).

If this fails, proceed as above, but choose *Safe Settings* instead. This option disables ACPI and DMA support. Most hardware will boot with this option.

If both of these options fail, use the boot options prompt to pass any additional parameters needed to support this type of hardware to the installation kernel. For more information about the parameters available as boot options, refer to the kernel documentation located in `/usr/src/linux/Documentation/kernel-parameters.txt`.

TIP: Obtaining Kernel Documentation

Install the `kernel-source` package to view the kernel documentation.

There are various other ACPI-related kernel parameters that can be entered at the boot prompt prior to booting for installation:

`acpi=off`

This parameter disables the complete ACPI subsystem on your computer. This may be useful if your computer cannot handle ACPI at all or if you think ACPI in your computer causes trouble.

`acpi=force`

Always enable ACPI even if your computer has an old BIOS dated before the year 2000. This parameter also enables ACPI if it is set in addition to `acpi=off`.

`acpi=noirq`

Do not use ACPI for IRQ routing.

`acpi=ht`

Run only enough ACPI to enable hyper-threading.

`acpi=strict`

Be less tolerant of platforms that are not strictly ACPI specification compliant.

`pci=noacpi`

Disable PCI IRQ routing of the new ACPI system.

`pnpacpi=off`

This option is for serial or parallel problems when your BIOS setup contains wrong interrupts or ports.

`notsc`

Disable the time stamp counter. This option can be used to work around timing problems on your systems. It is a recent feature, if you see regressions on your machine, especially time related or even total hangs, this option is worth a try.

`nohz=off`

Disable the nohz feature. If your machine hangs, this option may help. Otherwise it is of no use.

Once you have determined the right parameter combination, YaST automatically writes them to the boot loader configuration to make sure that the system boots properly next time.

If unexplainable errors occur when the kernel is loaded or during the installation, select *Memory Test* in the boot menu to check the memory. If *Memory Test* returns an error, it is usually a hardware error.

32.2.6 Fails to Launch Graphical Installer

After you insert the medium into your drive and reboot your machine, the installation screen comes up, but after you select *Installation*, the graphical installer does not start.

There are several ways to deal with this situation:

- Try to select another screen resolution for the installation dialogs.
- Select *Text Mode* for installation.
- Do a remote installation via VNC using the graphical installer.

Procedure 32.3 *Change Screen Resolution for Installation*

- 1** Boot for installation.
- 2** Press F3 to open a menu from which to select a lower resolution for installation purposes.
- 3** Select *Installation* and proceed with the installation as described in Chapter 6, *Installation with YaST* (↑*Deployment Guide*).

Procedure 32.4 *Installation in Text Mode*

- 1 Boot for installation.
- 2 Press F3 and select *Text Mode*.
- 3 Select *Installation* and proceed with the installation as described in Chapter 6, *Installation with YaST* (↑*Deployment Guide*).

Procedure 32.5 *VNC Installation*

- 1 Boot for installation.
- 2 Enter the following text at the boot options prompt:

```
vnc=1 vncpassword=some_password
```

Replace *some_password* with the password to use for VNC installation.

- 3 Select *Installation* then press Enter to start the installation .

Instead of starting right into the graphical installation routine, the system continues to run in a text mode, then halts, displaying a message containing the IP address and port number at which the installer can be reached via a browser interface or a VNC viewer application.

- 4 If using a browser to access the installer, launch the browser and enter the address information provided by the installation routines on the future SUSE Linux Enterprise Server machine and hit Enter:

```
http://ip_address_of_machine:5801
```

A dialog opens in the browser window prompting you for the VNC password. Enter it and proceed with the installation as described in Chapter 6, *Installation with YaST* (↑*Deployment Guide*).

IMPORTANT

Installation via VNC works with any browser under any operating system, provided Java support is enabled.

Provide the IP address and password to your VNC viewer when prompted. A window opens, displaying the installation dialogs. Proceed with the installation as usual.

32.2.7 Only Minimalistic Boot Screen Started

You inserted the medium into the drive, the BIOS routines are finished, but the system does not start with the graphical boot screen. Instead it launches a very minimalistic text-based interface. This may happen on any machine not providing sufficient graphics memory for rendering a graphical boot screen.

Although the text boot screen looks minimalistic, it provides nearly the same functionality as the graphical one:

Boot Options

Unlike the graphical interface, the different boot options cannot be selected using the cursor keys of your keyboard. The boot menu of the text mode boot screen offers some keywords to enter at the boot prompt. These keywords map to the options offered in the graphical version. Enter your choice and hit Enter to launch the boot process.

Custom Boot Options

After selecting a boot option, enter the appropriate keyword at the boot prompt or enter some custom boot options as described in Section 32.2.5, “Fails to Boot” (page 502). To launch the installation process, press Enter.

Screen Resolutions

Use the F keys to determine the screen resolution for installation. If you need to boot in text mode, choose F3.

32.3 Boot Problems

Boot problems are situations when your system does not boot properly (does not boot to the expected runlevel and login screen).

32.3.1 Fails to Load the GRUB Boot Loader

If the hardware is functioning properly, it is possible that the boot loader is corrupted and Linux cannot start on the machine. In this case, it is necessary to reinstall the boot loader. To reinstall the boot loader, proceed as follows:

- 1 Insert the installation media into the drive.
- 2 Reboot the machine.
- 3 Select *Installation* from the boot menu.
- 4 Select a language.
- 5 Accept the license agreement.
- 6 In the *Installation Mode* screen, select *Other* and set the installation mode to *Repair Installed System*.
- 7 Once in the YaST System Repair module, select *Expert Tools* then select *Install New Boot Loader*.
- 8 Restore the original settings and reinstall the boot loader.
- 9 Leave YaST System Repair and reboot the system.

Other reasons for the machine not booting may be BIOS-related:

BIOS Settings

Check your BIOS for references to your hard drive. GRUB may simply not be started if the hard drive itself cannot be found with the current BIOS settings.

BIOS Boot Order

Check whether your system's boot order includes the hard disk. If the hard disk option was not enabled, your system may install properly, but fails to boot when access to the hard disk is required.

32.3.2 No Graphical Login

If the machine comes up, but does not boot into the graphical login manager, anticipate problems either with the choice of the default runlevel or the configuration of the X Window System. To check the runlevel configuration, log in as the `root` user and check whether the machine is configured to boot into runlevel 5 (graphical desktop). A quick way to check this is to examine the contents of `/etc/inittab`, as follows:

```
tux@mercury:~> grep "id:" /etc/inittab
id:5:initdefault:
```

The returned line indicates that the machine's default runlevel (`initdefault`) is set to 5 and that it should boot to the graphical desktop. If the runlevel is set to any other number, use the YaST Runlevel Editor module to set it to 5.

IMPORTANT

Do not edit the runlevel configuration manually. Otherwise `SuSEconfig` (run by YaST) will overwrite these changes on its next run. If you need to make manual changes here, disable future `SuSEconfig` changes by setting `CHECK_INITTAB` in `/etc/sysconfig/suseconfig` to `no`.

If the runlevel is set to 5, your desktop or X Windows software is probably misconfigured or corrupted. Examine the log files at `/var/log/Xorg.*.log` for detailed messages from the X server as it attempted to start. If the desktop fails during start, it may log error messages to `/var/log/messages`. If these error messages hint at a configuration problem in the X server, try to fix these issues. If the graphical system still does not come up, consider reinstalling the graphical desktop.

TIP: Starting X Window System Manually

One quick test: the `startx` command should force the X Window System to start with the configured defaults if the user is currently logged in on the console. If that does not work, it should log errors to the console.

32.4 Login Problems

Login problems are those where your machine does, in fact, boot to the expected welcome screen or login prompt, but refuses to accept the username and password, or accepts them but then does not behave properly (fails to start the graphic desktop, produces errors, drops to a command line, etc.).

32.4.1 Valid Username and Password Combinations Fail

This usually occurs when the system is configured to use network authentication or directory services and, for some reason, is unable to retrieve results from its configured servers. The `root` user, as the only local user, is the only user that can still log in to these machines. The following are some common reasons why a machine appears functional but is unable to process logins correctly:

- The network is not working. For further directions on this, turn to Section 32.5, “Network Problems” (page 516).
- DNS is not working at the moment (which prevents GNOME or KDE from working and the system from making validated requests to secure servers). One indication that this is the case is that the machine takes an extremely long time to respond to any action. Find more information about this topic in Section 32.5, “Network Problems” (page 516).
- If the system is configured to use Kerberos, the system's local time may have drifted past the accepted variance with the Kerberos server time (this is typically 300 seconds). If NTP (network time protocol) is not working properly or local NTP servers are not working, Kerberos authentication ceases to function because it depends on common clock synchronization across the network.
- The system's authentication configuration is misconfigured. Check the PAM configuration files involved for any typographical errors or misordering of directives. For additional background information about PAM and the syntax of the configuration files involved, refer to Chapter 2, *Authentication with PAM* (↑*Security Guide*).
- The home partition is encrypted. Find more information about this topic in Section 32.4.3, “Login to Encrypted Home Partition Fails” (page 513).

In all cases that do not involve external network problems, the solution is to reboot the system into single-user mode and repair the configuration before booting again into operating mode and attempting to log in again. To boot into single-user mode:

- 1 Reboot the system. The boot screen appears, offering a prompt.
- 2 Enter `1` at the boot prompt to make the system boot into single-user mode.
- 3 Enter the username and password for `root`.
- 4 Make all the necessary changes.
- 5 Boot into the full multiuser and network mode by entering `telinit 5` at the command line.

32.4.2 Valid Username and Password Not Accepted

This is by far the most common problem users encounter, because there are many reasons this can occur. Depending on whether you use local user management and authentication or network authentication, login failures occur for different reasons.

Local user management can fail for the following reasons:

- The user may have entered the wrong password.
- The user's home directory containing the desktop configuration files is corrupted or write protected.
- There may be problems with the X Window System authenticating this particular user, especially if the user's home directory has been used with another Linux distribution prior to installing the current one.

To locate the reason for a local login failure, proceed as follows:

- 1 Check whether the user remembered his password correctly before you start debugging the whole authentication mechanism. If the user may not remember his password correctly, use the YaST User Management module to change the user's password. Pay attention to the Caps Lock key and unlock it, if necessary.

- 2 Log in as `root` and check `/var/log/messages` for error messages of the login process and of PAM.
- 3 Try to log in from a console (using `Ctrl + Alt + F1`). If this is successful, the blame cannot be put on PAM, because it is possible to authenticate this user on this machine. Try to locate any problems with the X Window System or the desktop (GNOME or KDE). For more information, refer to Section 32.4.4, “Login Successful but GNOME Desktop Fails” (page 514) and Section 32.4.5, “Login Successful but KDE Desktop Fails” (page 515).
- 4 If the user's home directory has been used with another Linux distribution, remove the `Xauthority` file in the user's home. Use a console login via `Ctrl + Alt + F1` and run `rm .Xauthority` as this user. This should eliminate X authentication problems for this user. Try graphical login again.
- 5 If graphical login still fails, do a console login with `Ctrl + Alt + F1`. Try to start an X session on another display—the first one (`:0`) is already in use:

```
startx -- :1
```

This should bring up a graphical screen and your desktop. If it does not, check the log files of the X Window System (`/var/log/Xorg.displaynumber.log`) or the log file for your desktop applications (`.xsession-errors` in the user's home directory) for any irregularities.
- 6 If the desktop could not start because of corrupt configuration files, proceed with Section 32.4.4, “Login Successful but GNOME Desktop Fails” (page 514) or Section 32.4.5, “Login Successful but KDE Desktop Fails” (page 515).

The following are some common reasons why network authentication for a particular user may fail on a specific machine:

- The user may have entered the wrong password.
- The username exists in the machine's local authentication files and is also provided by a network authentication system, causing conflicts.
- The home directory exists but is corrupt or unavailable. Perhaps it is write protected or is on a server that is inaccessible at the moment.

- The user does not have permission to log in to that particular host in the authentication system.
- The machine has changed hostnames, for whatever reason, and the user does not have permission to log in to that host.
- The machine cannot reach the authentication server or directory server that contains that user's information.
- There may be problems with the X Window System authenticating this particular user, especially if the user's home has been used with another Linux distribution prior to installing the current one.

To locate the cause of the login failures with network authentication, proceed as follows:

- 1** Check whether the user remembered their password correctly before you start debugging the whole authentication mechanism.
- 2** Determine the directory server which the machine relies on for authentication and make sure that it is up and running and properly communicating with the other machines.
- 3** Determine that the user's username and password work on other machines to make sure that his authentication data exists and is properly distributed.
- 4** See if another user can log in to the misbehaving machine. If another user can log in without difficulty or if `root` can log in, log in and examine the `/var/log/messages` file. Locate the time stamps that correspond to the login attempts and determine if PAM has produced any error messages.
- 5** Try to log in from a console (using `Ctrl + Alt + F1`). If this is successful, the problem is not with PAM or the directory server on which the user's home is hosted, because it is possible to authenticate this user on this machine. Try to locate any problems with the X Window System or the desktop (GNOME or KDE). For more information, refer to Section 32.4.4, “Login Successful but GNOME Desktop Fails” (page 514) and Section 32.4.5, “Login Successful but KDE Desktop Fails” (page 515).
- 6** If the user's home directory has been used with another Linux distribution, remove the `Xauthority` file in the user's home. Use a console login via `Ctrl + Alt +`

F1 and run `rm .Xauthority` as this user. This should eliminate X authentication problems for this user. Try graphical login again.

- 7 If graphical login still fails, do a console login with `Ctrl + Alt + F1`. Try to start an X session on another display—the first one (`:0`) is already in use:

```
startx -- :1
```

This should bring up a graphical screen and your desktop. If it does not, check the log files of the X Window System (`/var/log/Xorg.displaynumber.log`) or the log file for your desktop applications (`.xsession-errors` in the user's home directory) for any irregularities.

- 8 If the desktop could not start because of corrupt configuration files, proceed with Section 32.4.4, “Login Successful but GNOME Desktop Fails” (page 514) or Section 32.4.5, “Login Successful but KDE Desktop Fails” (page 515).

32.4.3 Login to Encrypted Home Partition Fails

It is recommended to use an encrypted home partition for laptops. If you cannot log in to your laptop, the reason is usually simple: your partition could not be unlocked.

During the boot time, you have to enter the passphrase to unlock your encrypted partition. If you do not enter it, the boot process continues, leaving the partition locked.

To unlock your encrypted partition, proceed as follows:

- 1 Switch to the text console with `Ctrl + Alt + F1`.
- 2 Become `root`.
- 3 Restart the unlocking process again with:

```
/etc/init.d/boot.cryptot restart
```

- 4 Enter your passphrase to unlock your encrypted partition.
- 5 Exit the text console and switch back to the login screen with `Alt + F7`.

6 Log in as usual.

32.4.4 Login Successful but GNOME Desktop Fails

If this is the case, it is likely that your GNOME configuration files have become corrupted. Some symptoms may include the keyboard failing to work, the screen geometry becoming distorted, or even the screen coming up as a bare gray field. The important distinction is that if another user logs in, the machine works normally. It is then likely that the problem can be fixed relatively quickly by simply moving the user's GNOME configuration directory to a new location, which causes GNOME to initialize a new one. Although the user is forced to reconfigure GNOME, no data is lost.

1 Switch to a text console by pressing Ctrl + Alt + F1.

2 Log in with your user name.

3 Move the user's GNOME configuration directories to a temporary location:

```
mv .gconf .gconf-ORIG-RECOVER
mv .gnome2 .gnome2-ORIG-RECOVER
```

4 Log out.

5 Log in again, but do not run any applications.

6 Recover your individual application configuration data (including the Evolution e-mail client data) by copying the `~/ .gconf-ORIG-RECOVER/apps/` directory back into the new `~/ .gconf` directory as follows:

```
cp -a .gconf-ORIG-RECOVER/apps .gconf/
```

If this causes the login problems, attempt to recover only the critical application data and reconfigure the remainder of the applications.

32.4.5 Login Successful but KDE Desktop Fails

There are several reasons why a KDE desktop would not allow users to login. Corrupted cache data can cause login problems as well as corrupt KDE desktop configuration files.

Cache data is used at desktop start-up to increase performance. If this data is corrupted, start-up is slowed down or fails entirely. Removing them forces the desktop start-up routines to start from scratch. This takes more time than a normal start-up, but data is intact after this and the user can login.

To remove the cache files of the KDE desktop, issue the following command as `root`:

```
rm -rf /tmp/kde-user /tmp/ksocket-user
```

Replace `user` with your username. Removing these two directories just removes the corrupted cache files. No real data is harmed using this procedure.

Corrupted desktop configuration files can always be replaced with the initial configuration files. If you want to recover the user's adjustments, carefully copy them back from their temporary location after the configuration has been restored, using the default configuration values.

To replace a corrupted desktop configuration with the initial configuration values, proceed as follows:

- 1 Switch to a text console by pressing `Ctrl + Alt + F1`.
- 2 Log in with your username.
- 3 Move the KDE configuration directory and the `.skel` files to a temporary location:

- For KDE3 use these commands:

```
mv .kde .kde-ORIG-RECOVER
mv .skel .skel-ORIG-RECOVER
```

- For KDE4 use these commands:

```
mv .kde4 .kde4-ORIG-RECOVER
mv .skel .skel-ORIG-RECOVER
```

4 Log out.

5 Log in again.

6 After the desktop has started successfully, copy the user's own configurations back into place:

```
cp -a KDEDIR/share .kde/share
```

Replace *KDEDIR* with the directory from Step 3 (page 515).

IMPORTANT

If the user's own adjustments caused the login to fail and continue to do so, repeat the procedure as described above, but do not copy the `.kde/share` directory.

32.5 Network Problems

Many problems of your system may be network-related, even though they do not seem to be at first. For example, the reason for a system not allowing users to log in may be a network problem of some kind. This section introduces a simple check list you can apply to identify the cause of any network problem encountered.

Procedure 32.6 *How to Identify Network Problems*

When checking the network connection of your machine, proceed as follows:

- 1** If you use an ethernet connection, check the hardware first. Make sure that your network cable is properly plugged into your computer and router (or hub, etc.). The control lights next to your ethernet connector are normally both be active.

If the connection fails, check whether your network cable works with another machine. If it does, your network card causes the failure. If hubs or switches are included in your network setup, they may be faulty, as well.

- 2 If using a wireless connection, check whether the wireless link can be established by other machines. If not, contact the wireless network's administrator.
- 3 Once you have checked your basic network connectivity, try to find out which service is not responding. Gather the address information of all network servers needed in your setup. Either look them up in the appropriate YaST module or ask your system administrator. The following list gives some of the typical network servers involved in a setup together with the symptoms of an outage.

DNS (Name Service)

A broken or malfunctioning name service affects the network's functionality in many ways. If the local machine relies on any network servers for authentication and these servers cannot be found due to name resolution issues, users would not even be able to log in. Machines in the network managed by a broken name server would not be able to “see” each other and communicate.

NTP (Time Service)

A malfunctioning or completely broken NTP service could affect Kerberos authentication and X server functionality.

NFS (File Service)

If any application needs data stored in an NFS mounted directory, it will not be able to start or function properly if this service was down or misconfigured. In the worst case scenario, a user's personal desktop configuration would not come up if their home directory containing the `.gconf` or `.kde` subdirectories could not be found due to a faulty NFS server.

Samba (File Service)

If any application needs data stored in a directory on a faulty Samba server, it will not be able to start or function properly.

NIS (User Management)

If your SUSE Linux Enterprise Server system relies on a faulty NIS server to provide the user data, users will not be able to log in to this machine.

LDAP (User Management)

If your SUSE Linux Enterprise Server system relies on a faulty LDAP server to provide the user data, users will not be able to log in to this machine.

Kerberos (Authentication)

Authentication will not work and login to any machine fails.

CUPS (Network Printing)

Users cannot print.

- 4 Check whether the network servers are running and whether your network setup allows you to establish a connection:

IMPORTANT

The debugging procedure described below only applies to a simple network server/client setup that does not involve any internal routing. It assumes both server and client are members of the same subnet without the need for additional routing.

- 4a Use `ping IP address or hostname` (replace *hostname* with the hostname of the server) to check whether each one of them is up and responding to the network. If this command is successful, it tells you that the host you were looking for is up and running and that the name service for your network is configured correctly.

If `ping` fails with `destination host unreachable`, either your system or the desired server is not properly configured or down. Check whether your system is reachable by running `ping IP address or your_hostname` from another machine. If you can reach your machine from another machine, it is the server that is not running at all or not configured correctly.

If `ping` fails with `unknown host`, the name service is not configured correctly or the hostname used was incorrect. For further checks on this matter, refer to Step 4b (page 518). If `ping` still fails, either your network card is not configured correctly or your network hardware is faulty.

- 4b Use `host hostname` to check whether the hostname of the server you are trying to connect to is properly translated into an IP address and vice versa. If this command returns the IP address of this host, the name service is up and running. If the `host` command fails, check all network configuration files relating to name and address resolution on your host:

`/etc/resolv.conf`

This file is used to keep track of the name server and domain you are currently using. It can be modified manually or automatically adjusted by YaST or DHCP. Automatic adjustment is preferable. However, make sure that this file has the following structure and all network addresses and domain names are correct:

```
search fully_qualified_domain_name
nameserver ipaddress_of_nameserver
```

This file can contain more than one name server address, but at least one of them must be correct to provide name resolution to your host. If needed, adjust this file using the YaST Network Setting module (Host-name/DNS tab).

If your network connection is handled via DHCP, enable DHCP to change hostname and name service information by selecting *Change Hostname via DHCP* and *Update Name Servers and Search List via DHCP* in the YaST DNS and Hostname module.

`/etc/nsswitch.conf`

This file tells Linux where to look for name service information. It should look like this:

```
...
hosts: files dns
networks: files dns
...
```

The `dns` entry is vital. It tells Linux to use an external name server. Normally, these entries are automatically managed by YaST, but it would be prudent to check.

If all the relevant entries on the host are correct, let your system administrator check the DNS server configuration for the correct zone information. For detailed information about DNS, refer to Chapter 21, *The Domain Name System* (page 289). If you have made sure that the DNS configuration of your host and the DNS server are correct, proceed with checking the configuration of your network and network device.

- 4c** If your system cannot establish a connection to a network server and you have excluded name service problems from the list of possible culprits, check the configuration of your network card.

Use the command `ifconfig network_device` (executed as `root`) to check whether this device was properly configured. Make sure that both `inet address` and `Mask` are configured correctly. An error in the IP address or a missing bit in your network mask would render your network configuration unusable. If necessary, perform this check on the server as well.

- 4d** If the name service and network hardware are properly configured and running, but some external network connections still get long time-outs or fail entirely, use `traceroute fully_qualified_domain_name` (executed as `root`) to track the network route these requests are taking. This command lists any gateway (hop) that a request from your machine passes on its way to its destination. It lists the response time of each hop and whether this hop is reachable at all. Use a combination of `traceroute` and `ping` to track down the culprit and let the administrators know.

Once you have identified the cause of your network trouble, you can resolve it yourself (if the problem is located on your machine) or let the system administrators of your network know about your findings so they can reconfigure the services or repair the necessary systems.

32.5.1 NetworkManager Problems

If you have a problem with network connectivity, narrow it down as described in Procedure 32.6, “How to Identify Network Problems” (page 516). If NetworkManager seems to be the culprit, proceed as follows to get logs providing hints on why NetworkManager fails:

- 1 Open a shell and log in as `root`.

- 2 Restart the NetworkManager:

```
rcnetwork restart -o nm
```

- 3 Open a web page, for example, <http://www.opensuse.org> as normal user to see, if you can connect.

- 4 Collect any information about the state of NetworkManager in `/var/log/NetworkManager`.

For more information about NetworkManager, refer to Chapter 23, *Using NetworkManager* (page 331).

32.6 Data Problems

Data problems are when the machine may or may not boot properly but, in either case, it is clear that there is data corruption on the system and that the system needs to be recovered. These situations call for a backup of your critical data, enabling you to recover the system state from before your system failed. SUSE Linux Enterprise Server offers dedicated YaST modules for system backup and restoration as well as a rescue system that can be used to recover a corrupted system from the outside.

32.6.1 Managing Partition Images

Sometimes you need to perform a backup from an entire partition or even hard disk. Linux comes with the `dd` tool which can create an exact copy of your disc. Combined with `gzip` you save some space.

Procedure 32.7 *Backing and Restoring up Hard discs*

- 1 Start a Shell as user `root`.
- 2 Select your source device. Typically this is something like `/dev/sda` (labeled as *SOURCE*).
- 3 Decide where you want to store your image (labeled as *BACKUP_PATH*). It must be different from your source device. In other words: if you make a backup from `/dev/sda`, your image file need not to be stored under `/dev/sda`.

- 4 Run the commands to create a compressed image file:

```
dd if=/dev/SOURCE | gzip > /BACKUP_PATH/image.gz
```

- 5 Restore the hard disk with the following commands:

```
gzip -dc /BACKUP_PATH/image.gz | dd of=/dev/SOURCE
```

If you only need a partition to backup, replace the *SOURCE* placeholder with your respective partition. In this case, your image file can lie on the same hard disk, but on a different partition.

32.6.2 Backing Up Critical Data

System backups can be easily managed using the YaST System Backup module:

- 1** As `root`, start YaST and select *System > System Backup*.
- 2** Create a backup profile holding all details needed for the backup, filename of the archive file, scope, and type of the backup:
 - 2a** Select *Profile Management > Add*.
 - 2b** Enter a name for the archive.
 - 2c** Enter the path to the location of the backup if you want to keep a local backup. For your backup to be archived on a network server (via NFS), enter the IP address or name of the server and the directory that should hold your archive.
 - 2d** Determine the archive type and click *Next*.
 - 2e** Determine the backup options to use, such as whether files not belonging to any package should be backed up and whether a list of files should be displayed prior to creating the archive. Also determine whether changed files should be identified using the time-consuming MD5 mechanism.

Use *Expert* to enter a dialog for the backup of entire hard disk areas. Currently, this option only applies to the Ext2 file system.
 - 2f** Finally, set the search constraints to exclude certain system areas from the backup area that do not need to be backed up, such as lock files or cache files. Add, edit, or delete items until your needs are met and leave with *OK*.

- 3 Once you have finished the profile settings, you can start the backup right away with *Create Backup* or configure automatic backup. It is also possible to create other profiles tailored for various other purposes.

To configure automatic backup for a given profile, proceed as follows:

- 1 Select *Automatic Backup* from the *Profile Management* menu.
- 2 Select *Start Backup Automatically*.
- 3 Determine the backup frequency. Choose *daily*, *weekly*, or *monthly*.
- 4 Determine the backup start time. These settings depend on the backup frequency selected.
- 5 Decide whether to keep old backups and how many should be kept. To receive an automatically generated status message of the backup process, check *Send Summary Mail to User root*.
- 6 Click *OK* to apply your settings and have the first backup start at the time specified.

32.6.3 Restoring a System Backup

Use the YaST System Restoration module to restore the system configuration from a backup. Restore the entire backup or select specific components that were corrupted and need to be reset to their old state.

- 1 Start *YaST > System > System Restoration*.
- 2 Enter the location of the backup file. This could be a local file, a network mounted file, or a file on a removable device, such as a floppy or a DVD. Then click *Next*.

The following dialog displays a summary of the archive properties, such as the filename, date of creation, type of backup, and optional comments.
- 3 Review the archived content by clicking *Archive Content*. Clicking *OK* returns you to the *Archive Properties* dialog.
- 4 *Expert Options* opens a dialog in which to fine-tune the restore process. Return to the *Archive Properties* dialog by clicking *OK*.
- 5 Click *Next* to open the view of packages to restore. Press *Accept* to restore all files in the archive or use the various *Select All*, *Deselect All*, and *Select Files* buttons to fine-tune your selection. Only use the *Restore RPM Database* option if the RPM database is corrupted or deleted and this file is included in the backup.
- 6 After you click *Accept*, the backup is restored. Click *Finish* to leave the module after the restore process is completed.

32.6.4 Recovering a Corrupted System

There are several reasons why a system could fail to come up and run properly. A corrupted file system following a system crash, corrupted configuration files, or a corrupted boot loader configuration are the most common ones.

SUSE Linux Enterprise Server offers two different methods to resolve these situations. You can either use the YaST System Repair functionality or boot the rescue system. The following sections cover both types of system repair methods.

Using YaST System Repair

NOTE: Keyboard and Language Settings

If you change the language settings after booting, your keyboard is adapted as well.

Before launching the YaST System Repair module, determine in which mode to run it to best fit your needs. Depending on the severity and cause of your system failure (and your expertise), there are three different modes to choose from:

Automatic Repair

If your system failed due to an unknown cause and you basically do not know which part of the system is to blame for the failure, use *Automatic Repair*. An extensive automated check will be performed on all components of your installed system. For a detailed description of this procedure, refer to Section “Automatic Repair” (page 525).

Customized Repair

If your system failed and you already know which component is to blame, you can cut the lengthy system check with *Automatic Repair* short by limiting the scope of the system analysis to those components. For example, if the system messages prior to the failure seem to indicate an error with the package database, you can limit the analysis and repair procedure to checking and restoring this aspect of your system. For a detailed description of this procedure, refer to Section “Customized Repair” (page 527).

Expert Tools

If you already have a clear idea of what component failed and how this should be fixed, you can skip the analysis runs and directly apply the tools necessary for the repair of the relevant component. For details, refer to Section “Expert Tools” (page 528).

Choose one of the repair modes as described above and proceed with the system repair as outlined in the following sections.

Automatic Repair

To start the automatic repair mode of YaST System Repair, proceed as follows:

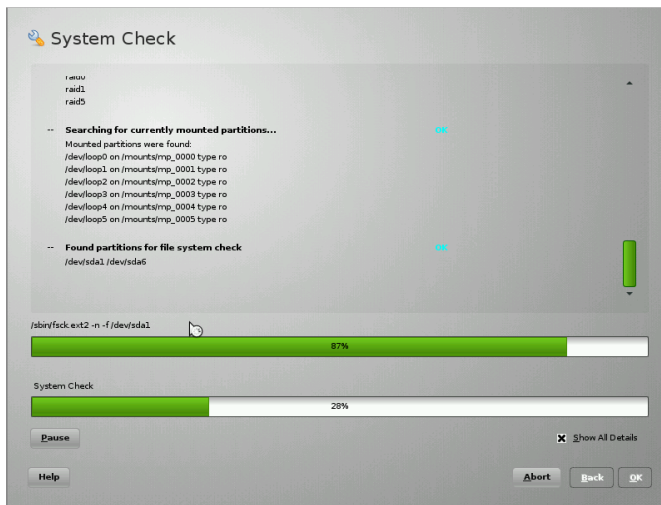
- 1 Insert the installation medium of SUSE Linux Enterprise Server into your DVD drive.
- 2 Reboot the system.
- 3 On the boot screen, select *Repair Installed System*.

4 Confirm the license agreement and click *Next*.

5 Select *Automatic Repair*.

YaST now launches an extensive analysis of the installed system. The progress of the procedure is displayed at the bottom of the screen with two progress bars. The upper bar shows the progress of the currently running test. The lower bar shows the overall progress of the analysis. The log window in the top section tracks the currently running test and its result. See Figure 32.4, “Automatic Repair Mode” (page 526).

Figure 32.4 *Automatic Repair Mode*



The following main test runs are performed with every run. They contain, in turn, a number of individual subtests:

Check Partition Tables

Checks the validity and coherence of the partition tables of all detected hard disks.

Check Swap Areas

The swap partitions of the installed system are detected, tested, and offered for activation, where applicable. This offer should be accepted for the sake of a higher system repair speed.

Check File Systems

All detected file systems are subjected to a file system–specific check.

Check fstab Entries

The entries in the file are checked for completeness and consistency. All valid partitions are mounted.

Check Package Database

This checks whether all packages necessary for the operation of a minimal installation are present. While it is optionally possible to also analyze the base packages, this takes a long time because of their vast number.

Check Boot Loader Configuration

The boot loader configuration of the installed system (GRUB or LILO) is checked for completeness and coherence. Boot and root devices are examined and the availability of the initrd modules is checked.

- 6 Whenever an error is encountered, the procedure stops and a dialog opens outlining the details and possible solutions.

Read the screen messages carefully before accepting the proposed fix. If you decide to decline a proposed solution, your system remains unchanged.

- 7 After the repair process has been terminated successfully, click *OK* and *Finish* and remove the installation media. The system automatically reboots.

Customized Repair

To launch the *Customized Repair* mode and selectively check certain components of your installed system, proceed as follows:

- 1 Insert the installation medium of SUSE Linux Enterprise Server into your DVD drive.
- 2 Reboot the system.
- 3 At the boot screen, select *Repair Installed System*.
- 4 Confirm the license agreement and click *Next*.
- 5 Select *Customized Repair*.

Choosing *Customized Repair* shows a list of test runs that are all marked for execution at first. The total range of tests matches that of automatic repair. If you already know where no damage is present, unmark the corresponding tests. Clicking *Next* starts a narrower test procedure that probably has a significantly shorter running time.

Not all test groups can be applied individually. The analysis of the fstab entries is always bound to an examination of the file systems, including existing swap partitions. YaST automatically resolves such dependencies by selecting the smallest number of necessary test runs. YaST does not support encrypted partitions. If you have one, YaST informs you about it.

- 6 Whenever an error is encountered, the procedure stops and a dialog opens outlining the details and possible solutions.

Read the screen messages carefully before accepting the proposed fix. If you decide to decline a proposed solution, your system remains unchanged.

- 7 After the repair process has been terminated successfully, click *OK* and *Finish* and remove the installation media. The system automatically reboots.

Expert Tools

If you are knowledgeable with SUSE Linux Enterprise Server and already have a very clear idea of what needs to be repaired in your system, directly apply the tools, skipping the system analysis.

To make use of the *Expert Tools* feature of the YaST System Repair module, proceed as follows:

- 1 Insert the installation medium of SUSE Linux Enterprise Server into your DVD drive.
- 2 Reboot the system.
- 3 At the boot screen, select *Repair Installed System*.
- 4 Confirm the license agreement and click *Next*.
- 5 Select *Expert Tools* and choose a repair option.

- 6 After the repair process has been terminated successfully, click *OK* and *Finish* and remove the installation media. The system automatically reboots.

The *Expert Tools* provides the following options to repair your faulty system:

Install New Boot Loader

This starts the YaST boot loader configuration module. Find details in Section 8.2, “Configuring the Boot Loader with YaST” (page 93).

Boot Installed System

Try to boot an already-installed Linux system.

Start Partitioning Tool

This starts the expert partitioning tool in YaST.

Repair File System

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

Recover Lost Partitions

It is possible to attempt to reconstruct damaged partition tables. A list of detected hard disks is presented first for selection. Clicking *OK* starts the examination. This can take a while depending on the speed of your computer and the size and speed of the hard disk.

IMPORTANT: *Reconstructing a Partition Table*

The reconstruction of a partition table is tricky. YaST attempts to recognize lost partitions by analyzing the data sectors of the hard disk. The lost partitions are added to the rebuilt partition table when recognized. This is, however, not successful in all imaginable cases.

Save System Settings to Floppy

This option saves important system files to a floppy disk. If one of these files becomes damaged, it can be restored from disk.

Verify Installed Software

This checks the consistency of the package database and the availability of the most important packages. Any damaged installed packages can be reinstalled with this tool.

Using the Rescue System

SUSE Linux Enterprise Server contains a rescue system. The rescue system is a small Linux system that can be loaded into a RAM disk and mounted as root file system, allowing you to access your Linux partitions from the outside. Using the rescue system, you can recover or modify any important aspect of your system:

- Manipulate any type of configuration file.
- Check the file system for defects and start automatic repair processes.
- Access the installed system in a “change root” environment.
- Check, modify, and reinstall the boot loader configuration.
- Resize partitions using the parted command. Find more information about this tool at the GNU Parted website <http://www.gnu.org/software/parted/parted.html>.

The rescue system can be loaded from various sources and locations. The simplest option is to boot the rescue system from the original installation medium:

- 1 Insert the installation medium into your DVD drive.
- 2 Reboot the system.
- 3 At the boot screen, press F4 and choose *DVD-ROM*. Then choose *Rescue System* from the main menu.
- 4 Enter `root` at the `Rescue:` prompt. A password is not required.

If your hardware setup does not include a DVD drive, you can boot the rescue system from a network source. The following example applies to a remote boot scenario—if using another boot medium, such as a DVD, modify the `info` file accordingly and boot as you would for a normal installation.

- 1 Enter the configuration of your PXE boot setup and add the lines
`install=protocol://instsource` and `rescue=1`. If you need to start the repair system, use `repair=1` instead. As with a normal installation,

protocol stands for any of the supported network protocols (NFS, HTTP, FTP, etc.) and *instsource* for the path to your network installation source.

- 2 Boot the system using “Wake on LAN”, as described in Section “Wake on LAN” (Chapter 14, *Remote Installation*, ↑*Deployment Guide*).
- 3 Enter `root` at the `Rescue :` prompt. A password is not required.

Once you have entered the rescue system, you can make use of the virtual consoles that can be reached with `Alt + F1` to `Alt + F6`.

A shell and many other useful utilities, such as the `mount` program, are available in the `/bin` directory. The `sbin` directory contains important file and network utilities for reviewing and repairing the file system. This directory also contains the most important binaries for system maintenance, such as `fdisk`, `mkfs`, `mkswap`, `mount`, `mount`, `init`, and `shutdown`, and `ifconfig`, `ip`, `route`, and `netstat` for maintaining the network. The directory `/usr/bin` contains the `vi` editor, `find`, `less`, and `ssh`.

To see the system messages, either use the command `dmesg` or view the file `/var/log/messages`.

Checking and Manipulating Configuration Files

As an example for a configuration that might be fixed using the rescue system, imagine you have a broken configuration file that prevents the system from booting properly. You can fix this using the rescue system.

To manipulate a configuration file, proceed as follows:

- 1 Start the rescue system using one of the methods described above.
- 2 To mount a root file system located under `/dev/sda6` to the rescue system, use the following command:

```
mount /dev/sda6 /mnt
```

All directories of the system are now located under `/mnt`

- 3 Change the directory to the mounted root file system:

```
cd /mnt
```

4 Open the problematic configuration file in the vi editor. Adjust and save the configuration.

5 Unmount the root file system from the rescue system:

```
umount /mnt
```

6 Reboot the machine.

Repairing and Checking File Systems

Generally, file systems cannot be repaired on a running system. If you encounter serious problems, you may not even be able to mount your root file system and the system boot may end with a “kernel panic”. In this case, the only way is to repair the system from the outside. It is strongly recommended to use the YaST System Repair for this task (see Section “Using YaST System Repair” (page 524) for details). However, if you need to do a manual file system check or repair, boot the rescue system. It contains the utilities to check and repair the `ext2`, `ext3`, `ext4`, `reiserfs`, `xfs`, `dosfs`, and `vfat` file systems.

Accessing the Installed System

If you need to access the installed system from the rescue system to, you need to do this in a *change root* environment. For example, to modify the boot loader configuration, or to execute a hardware configuration utility.

To set up a change root environment based on the installed system, proceed as follows:

1 First mount the root partition from the installed system and the device file system (change the device name to your current settings):

```
mount /dev/sda6 /mnt
mount --bind /dev /mnt/dev
```

2 Now you can “change root” into the new environment:

```
chroot /mnt
```

3 Then mount `/proc` and `/sys`:

```
mount /proc
mount /sys
```

- 4 Finally, mount the remaining partitions from the installed system:

```
mount -a
```

- 5 Now you have access to the installed system. Before rebooting the system, unmount the partitions with `umount -a` and leave the “change root” environment with `exit`.

WARNING: Limitations

Although you have full access to the files and applications of the installed system, there are some limitations. The kernel that is running is the one that was booted with the rescue system, not with the change root environment. It only supports essential hardware and it is not possible to add kernel modules from the installed system unless the kernel versions are exactly the same (which is unlikely). So you cannot access a sound card, for example. It is also not possible to start a graphical user interface.

Also note that you leave the “change root” environment when you switch the console with `Alt + F1` to `Alt + F6`.

Modifying and Reinstalling the Boot Loader

Sometimes a system cannot boot because the boot loader configuration is corrupted. The start-up routines cannot, for example, translate physical drives to the actual locations in the Linux file system without a working boot loader.

To check the boot loader configuration and reinstall the boot loader, proceed as follows:

- 1 Perform the necessary steps to access the installed system as described in Section “Accessing the Installed System” (page 532).
- 2 Check whether the following files are correctly configured according to the GRUB configuration principles outlined in Chapter 8, *The Boot Loader GRUB* (page 81) and apply fixes if necessary.

- `/etc/grub.conf`
- `/boot/grub/device.map`

- `/boot/grub/menu.lst`
- `/etc/sysconfig/bootloader`

3 Reinstall the boot loader using the following command sequence:

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

4 Unmount the partitions, log out from the “change root” environment, and reboot the system:

```
umount -a  
exit  
reboot
```

32.7 IBM System z: Using initrd as a Rescue System

If the kernel of the SUSE® Linux Enterprise Server for IBM System z is upgraded or modified, it is possible to reboot the system accidentally in an inconsistent state, so standard procedures of IPLing the installed system fail. This most commonly occurs if a new or updated SUSE Linux Enterprise Server kernel has been installed and the `zipl` program has not been run to update the IPL record. In this case, use the standard installation package as a rescue system from which the `zipl` program can be executed to update the IPL record.

32.7.1 IPLing the Rescue System

IMPORTANT: Making the Installation Data Available

For this method to work, the SUSE Linux Enterprise Server for IBM System z installation data must be available. For details, refer to Section “Making the Installation Data Available” (Chapter 4, *Installation on IBM System z*, ↑*Deployment Guide*). Additionally, you need the channel number of the device and the partition number within the device that contains the root file system of the SUSE Linux Enterprise Server installation.

First, IPL the SUSE Linux Enterprise Server for IBM System z installation system as described in Section “Preparing for Installation” (Chapter 4, *Installation on IBM System z*, ↑*Deployment Guide*). A list of choices for the network adapter to use is then presented.

Select *Start Installation or System* then *Start Rescue System* to start the rescue system. Depending on the installation environment, you now must specify the parameters for the network adapter and the installation source. The rescue system is loaded and the following login prompt is shown at the end:

```
Skipped services in runlevel 3:  nfs nfsboot
```

```
Rescue login:
```

You can now login as `root` without a password.

32.7.2 Configuring Disks

In this state, no disks are configured. You need to configure them before you can proceed.

Procedure 32.8 *Configuring DASDs*

- 1 Configure DASDs with the following command:

```
dasd_configure 0.0.0150 1 0
```

0.0.0150 is the channel to which the DASD is connected. The 1 means activate the disk (a 0 at this place would deactivate the disk). The 0 stands for “no DIAG mode” for the disk (a 1 here would enable DAIG access to the disk).

- 2 Now the DASD is online (check with `cat /proc/partitions`) and can be used for subsequent commands.

Procedure 32.9 *Configuring a zFCP Disk*

- 1 To configure a zFCP disk, it is necessary to first configure the zFCP adapter. Do this with the following command:

```
zfcphost_configure 0.0.4000 1
```

0.0.4000 is the channel to which the adapter is attached and 1 stands for activate (a 0 here would deactivate the adapter).

- 2 After the adapter is activated, a disk can be configured. Do this with the following command:

```
zfcplib_configure 0.0.4000 1234567887654321 8765432100000000 1
```

0.0.4000 is the previously-used channel ID, 1234567887654321 is the WWPN (World wide Port Number), and 8765432100000000 is the LUN (logical unit number). The 1 stands for activating the disk (a 0 here would deactivate the disk).

- 3 Now the zFCP disk is online (check with `cat /proc/partitions`) and can be used for subsequent commands.

32.7.3 Mounting the Root Device

If all needed disks are online, you should now be able to mount the root device. Assuming that the root device is on the second partition of the DASD device (`/dev/dasda2`), the corresponding command is `mount /dev/dasda2 /mnt`.

IMPORTANT: File System Consistency

If the installed system has not been shut down properly, it may be advisable to check the file system consistency prior to mounting. This prevents any accidental loss of data. Using this example, issue the command `fsck /dev/dasda2` to ensure that the file system is in a consistent state.

By just issuing the command `mount`, it is possible to check whether the file system could be mounted correctly.

Example 32.1 *Output of the Mount Command*

```
SuSE Instsys suse:/ # mount
shmfs on /newroot type shm (rw,nr_inodes=10240)
devpts on /dev/pts type devpts (rw)
virtual-proc-filessystem on /proc type proc (rw)
/dev/dasda2 on /mnt type reiserfs (rw)
```

32.7.4 Changing to the Mounted File System

For the `zipl` command to read the configuration file from the root device of the installed system and not from the rescue system, change the root device to the installed system with the `chroot` command:

Example 32.2 *chroot to the Mounted File System*

```
SuSE Instsys suse:/ # cd /mnt
SuSE Instsys suse:/mnt # chroot /mnt
```

32.7.5 Executing zipl

Now execute `zipl` to rewrite the IPL record with the correct values:

Example 32.3 *Installing the IPL Record with zipl*

```
sh-2.05b# zipl
building bootmap : /boot/zipl/bootmap
adding Kernel Image : /boot/kernel/image located at 0x00010000
adding Ramdisk : /boot/initrd located at 0x00800000
adding Parmline : /boot/zipl/parmfile located at 0x00001000
Bootloader for ECKD type devices with z/OS compatible layout installed.
Syncing disks....
...done
```

32.7.6 Exiting the Rescue System

To exit the rescue system, first leave the shell opened by the `chroot` command with `exit`. To prevent any loss of data, flush all unwritten buffers to disk with the `sync` command. Now change to the root directory of the rescue system and unmount the root device of SUSE Linux Enterprise Server for IBM System z installation.

Example 32.4 *Unmounting the File System*

```
SuSE Instsys suse:/mnt # cd /
SuSE Instsys suse:/ # umount /mnt
```

Finally, halt the rescue system with the `halt` command. The SUSE Linux Enterprise Server system can now be IPLed as described in Section “IBM System z: IPLing the Installed System” (Chapter 6, *Installation with YaST*, ↑*Deployment Guide*).

