openSUSE

11.1

www.novell.com

March 06, 2009

Reference

A NOVELL BUSINESS

Reference

Contents

About This Guide

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

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

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

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

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

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

Mobility
: Get an introduction to mobile computing with openSUSE, get to know the various options for wireless computing and power management and learn how to use a tablet PC.

Security
: Become acquainted with openSUSE security features and learn how to set up and configure services that will make your system secure.

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/opensuse111` or to the following section.

1 Available Documentation

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

Start-Up (↑Start-Up)
: Guides you through the installation and basic configuration of your system. For newcomers, the manual also introduces basic Linux concepts such as the file system, the user concept and access permissions and gives an overview of the features openSUSE offers to support mobile computing. Provides help and advice in troubleshooting.

KDE User Guide (↑KDE User Guide)
: Introduces the KDE desktop of openSUSE. It guides you through using and configuring the desktop and helps you perform key tasks. It is intended mainly for users who want to make efficient use of KDE as their default desktop.

GNOME User Guide (↑GNOME User Guide)
: Introduces the GNOME desktop of openSUSE. It guides you through using and configuring the desktop and helps you perform key tasks. It is intended mainly for end users who want to make efficient use of GNOME desktop as their default desktop.

Application Guide (↑Application Guide)
: Learn how to use and configure key desktop applications on openSUSE. This guide introduces browsers and e-mail clients as well as office applications and collaboration tools. It also covers graphics and multimedia applications.

Reference (page 1)
: Gives you a general understanding of openSUSE and covers advanced system administration tasks. It is intended mainly for system administrators and home users

with basic system administration knowledge. It provides detailed information about advanced deployment scenarios, administration of your system, the interaction of key system components and the set-up of various network and file services openSUSE offers.

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

Lessons For Lizards
: A community book project for the openSUSE distribution. A snapshot of the manual written by the open source community is released on an equal footing with the Novell/SUSE manuals. The lessons are written in a cook book style and cover more specific or exotic topics than the traditional manuals. For more information, see `http://developer.novell.com/wiki/index.php/Lessons_for_Lizards`.

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

KDE Quick Start (↑KDE Quick Start)
: Gives a short introduction to the KDE desktop and some key applications running on it.

GNOME Quick Start (↑GNOME Quick Start)
: Gives a short introduction to the GNOME desktop and some key applications running on it.

Novell AppArmor Quick Start
: Helps you understand the main concepts behind Novell® AppArmor.

Find HTML versions of most openSUSE 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.

For information where to find the books on your installation media, refer to the Release Notes of this product. The Release Notes are available from your installed system under

/usr/share/doc/release-notes/ or in the help centers of your KDE or GNOME desktop.

2 Feedback

Several feedback channels are available:

- To report bugs for a product component or to submit enhancements requests, please use https://bugzilla.novell.com/. If you are new to Bugzilla, you might find the *Submitting Bug Reports* article—available under http://en.opensuse.org/Submitting_Bug_Reports helpful. Frequently asked questions on reporting bugs are available under http://en.opensuse.org/Bug_Reporting_FAQ.

- We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation and enter your comments there.

3 Documentation Conventions

The following typographical conventions are used in this manual:

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

- *`placeholder`*: replace *`placeholder`* with the actual value

- `PATH`: the environment variable PATH

- `ls`, `--help`: commands, options, and parameters

- `user`: users or groups

- Alt, Alt + F1: a key to press or a key combination; keys are shown in uppercase as on a keyboard

- *File*, *File* > *Save As*: menu items, buttons

- *Dancing Penguins* (Chapter *Penguins*, ↑Another Manual): This is a reference to a chapter in another manual.

4 About the Making of This Manual

This book is written in Novdoc, a subset of DocBook (see `http://www.docbook.org`). The XML source files were validated by `xmllint`, processed by `xsltproc`, and converted into XSL-FO using a customized version of Norman Walsh's stylesheets. The final PDF is formatted through XEP from RenderX.

5 Source Code

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

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

6 Acknowledgments

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

Have a lot of fun!

Your SUSE Team

Part I. Advanced Deployment Scenarios

1 Remote Installation

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

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

NOTE

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

1.1 Installation Scenarios for Remote Installation

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

IMPORTANT

The configuration of the X Window System is not part of any remote installation process. After the installation has finished, log in to the target system as `root`, enter `telinit 3`, and start SaX2 to configure the graphics hardware as described in Section "Setting Up Graphics Card and Monitor" (Chapter 2, *Setting Up Hardware Components with YaST*, ↑Start-Up).

1.1.1 Simple Remote Installation via VNC—Static Network Configuration

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

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

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

To perform this kind of installation, proceed as follows:

1. Set up the installation source as described in Section 1.2, "Setting Up the Server Holding the Installation Sources" (page 12). Choose an NFS, HTTP, or FTP network server. For an SMB installation source, refer to Section 1.2.5, "Managing an SMB Installation Source" (page 20).

2 Boot the target system using the first CD or DVD of the openSUSE media kit.

3 When the boot screen of the target system appears, use the boot options prompt to set the appropriate VNC options and the address of the installation source. This is described in detail in Section 1.4, “Booting the Target System for Installation” (page 33).

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

4 On the controlling workstation, open a VNC viewing application or Web browser and connect to the target system as described in Section 1.5.1, “VNC Installation” (page 36).

5 Perform the installation as described in Chapter 1, *Installation with YaST* (↑Start-Up). Reconnect to the target system after it reboots for the final part of the installation.

6 Finish the installation.

1.1.2 Simple Remote Installation via VNC—Dynamic Network Configuration

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

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

- Remote installation source: NFS, HTTP, FTP, or SMB with working network connection.

- Target system with working network connection.

- Controlling system with working network connection and VNC viewer software or Java-enabled browser (Firefox, Konqueror, Internet Explorer, or Opera).

- Physical boot medium (CD, DVD, or custom boot disk) for booting the target system.

- Running DHCP server providing IP addresses.

To perform this kind of installation, proceed as follows:

1. Set up the installation source as described in Section 1.2, "Setting Up the Server Holding the Installation Sources" (page 12). Choose an NFS, HTTP, or FTP network server. For an SMB installation source, refer to Section 1.2.5, "Managing an SMB Installation Source" (page 20).

2. Boot the target system using the first CD or DVD of the openSUSE media kit.

3. When the boot screen of the target system appears, use the boot options prompt to set the appropriate VNC options and the address of the installation source. This is described in detail in Section 1.4, "Booting the Target System for Installation" (page 33).

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

4. On the controlling workstation, open a VNC viewing application or Web browser and connect to the target system as described in Section 1.5.1, "VNC Installation" (page 36).

5. Perform the installation as described in Chapter 1, *Installation with YaST* (↑Start-Up). Reconnect to the target system after it reboots for the final part of the installation.

6. Finish the installation.

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

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

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

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

To perform this type of installation, proceed as follows:

1. Set up the installation source as described in Section 1.2, "Setting Up the Server Holding the Installation Sources" (page 12). Choose an NFS, HTTP, or FTP network server or configure an SMB installation source as described in Section 1.2.5, "Managing an SMB Installation Source" (page 20).
2. Set up a TFTP server to hold a boot image that can be pulled by the target system. This is described in Section 1.3.2, "Setting Up a TFTP Server" (page 25).
3. Set up a DHCP server to provide IP addresses to all machines and reveal the location of the TFTP server to the target system. This is described in Section 1.3.1, "Setting Up a DHCP Server" (page 22).
4. Prepare the target system for PXE boot. This is described in further detail in Section 1.3.5, "Preparing the Target System for PXE Boot" (page 32).

5. Initiate the boot process of the target system using Wake on LAN. This is described in Section 1.3.7, “Wake on LAN” (page 32).

6. On the controlling workstation, open a VNC viewing application or Web browser and connect to the target system as described in Section 1.5.1, “VNC Installation” (page 36).

7. Perform the installation as described in Chapter 1, *Installation with YaST* (↑Start-Up). Reconnect to the target system after it reboots for the final part of the installation.

8. Finish the installation.

1.1.4 Simple Remote Installation via SSH—Static Network Configuration

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

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

- Remote installation source: NFS, HTTP, FTP, or SMB with working network connection.

- Target system with working network connection.

- Controlling system with working network connection and working SSH client software.

- Physical boot medium (CD, DVD, or custom boot disk) for the target system.

- Valid static IP addresses already assigned to the installation source and the controlling system.

- Valid static IP address to assign to the target system.

To perform this kind of installation, proceed as follows:

1. Set up the installation source as described in Section 1.2, “Setting Up the Server Holding the Installation Sources” (page 12). Choose an NFS, HTTP, or FTP network server. For an SMB installation source, refer to Section 1.2.5, “Managing an SMB Installation Source” (page 20).

2. Boot the target system using the first CD or DVD of the openSUSE media kit.

3. When the boot screen of the target system appears, use the boot options prompt to set the appropriate parameters for network connection, address of the installation source, and SSH enablement. This is described in detail in Section 1.4.2, “Using Custom Boot Options” (page 33).

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

4. On the controlling workstation, open a terminal window and connect to the target system as described in Section “Connecting to the Installation Program” (page 38).

5. Perform the installation as described in Chapter 1, *Installation with YaST* (↑Start-Up). Reconnect to the target system after it reboots for the final part of the installation.

6. Finish the installation.

1.1.5 Simple Remote Installation via SSH—Dynamic Network Configuration

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

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

- Remote installation source: NFS, HTTP, FTP, or SMB with working network connection.
- Target system with working network connection.
- Controlling system with working network connection and working SSH client software.
- Physical boot medium (CD or DVD) for booting the target system.
- Running DHCP server providing IP addresses.

To perform this kind of installation, proceed as follows:

1. Set up the installation source as described in Section 1.2, “Setting Up the Server Holding the Installation Sources” (page 12). Choose an NFS, HTTP, or FTP network server. For an SMB installation source, refer to Section 1.2.5, “Managing an SMB Installation Source” (page 20).

2. Boot the target system using the first CD or DVD of the openSUSE media kit.

3. When the boot screen of the target system appears, use the boot options prompt to pass the appropriate parameters for network connection, location of the installation source, and SSH enablement. See Section 1.4.2, “Using Custom Boot Options” (page 33) for detailed instructions on the use of these parameters.

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

4. On the controlling workstation, open a terminal window and connect to the target system as described in Section “Connecting to the Installation Program” (page 38).

5. Perform the installation as described in Chapter 1, *Installation with YaST* (↑Start-Up). Reconnect to the target system after it reboots for the final part of the installation.

6. Finish the installation.

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

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

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

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

To perform this type of installation, proceed as follows:

1. Set up the installation source as described in Section 1.2, “Setting Up the Server Holding the Installation Sources” (page 12). Choose an NFS, HTTP, or FTP network server. For the configuration of an SMB installation source, refer to Section 1.2.5, “Managing an SMB Installation Source” (page 20).
2. Set up a TFTP server to hold a boot image that can be pulled by the target system. This is described in Section 1.3.2, “Setting Up a TFTP Server” (page 25).
3. Set up a DHCP server to provide IP addresses to all machines and reveal the location of the TFTP server to the target system. This is described in Section 1.3.1, “Setting Up a DHCP Server” (page 22).
4. Prepare the target system for PXE boot. This is described in further detail in Section 1.3.5, “Preparing the Target System for PXE Boot” (page 32).
5. Initiate the boot process of the target system using Wake on LAN. This is described in Section 1.3.7, “Wake on LAN” (page 32).

6 On the controlling workstation, start an SSH client and connect to the target system as described in Section 1.5.2, "SSH Installation" (page 38).

7 Perform the installation as described in Chapter 1, *Installation with YaST* (↑Start-Up). Reconnect to the target system after it reboots for the final part of the installation.

8 Finish the installation.

1.2 Setting Up the Server Holding the Installation Sources

Depending on the operating system running on the machine to use as network installation source for openSUSE, there are several options for the server configuration. The easiest way to set up an installation server is to use YaST on SUSE Linux 9.3 and higher.

TIP

You can even use a Microsoft Windows machine as installation server for your Linux deployment. See Section 1.2.5, "Managing an SMB Installation Source" (page 20) for details.

1.2.1 Setting Up an Installation Server Using YaST

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

1 Log in as `root` to the machine that should act as installation server.

2 Install the `yast2-instserver` package.

3 Start *YaST* > *Miscellaneous* > *Installation Server*.

4 Select the server type (HTTP, FTP, or NFS). The selected server service is started automatically every time the system starts. If a service of the selected type is already running on your system and you want to configure it manually for the server, deactivate the automatic configuration of the server service with *Do Not Configure Any Network Services*. In both cases, define the directory in which the installation data should be made available on the server.

5 Configure the required server type. This step relates to the automatic configuration of server services. It is skipped when automatic configuration is deactivated.

Define an alias for the root directory of the FTP or HTTP server on which the installation data should be found. The installation source will later be located under `ftp://`*`Server-IP/Alias/Name`* (FTP) or under `http://`*`Server-IP/Alias/Name`* (HTTP). *`Name`* stands for the name of the installation source, which is defined in the following step. If you selected NFS in the previous step, define wild cards and export options. The NFS server will be accessible under `nfs://`*`Server-IP/Name`*. Details of NFS and exports can be found in Chapter 24, *Sharing File Systems with NFS* (page 367).

TIP: Firewall Settings

Make sure that the firewall settings of your server system allow traffic on the ports for HTTP, NFS, and FTP. If they currently do not, start the YaST firewall module and open the respective ports.

---

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

> **TIP**
>
> Consider announcing your installation source via OpenSLP if your network setup supports this option. This saves you from entering the network installation path on every target machine. The target systems are just booted using the SLP boot option and find the network installation source without any further configuration. For details on this option, refer to Section 1.4, "Booting the Target System for Installation" (page 33).

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

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

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

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

1.2.2 Setting Up an NFS Installation Source Manually

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

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

1. Log in as `root`.

2. Create a directory that should later hold all installation data and change into this directory. For example:

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

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

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

3. For each CD contained in the media kit execute the following commands:

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

   ```
   cp -a /media/path_to_your_CD-ROM_drive .
   ```

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

   3b Rename the directory to the CD number:

   ```
   mv path_to_your_CD-ROM_drive CDx
   ```

   Replace *`x`* with the actual number of your CD.

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

1. Log in as `root`.

2. Start *YaST* > *Network Services* > *NFS Server*.

3. Select *Start* and *Open Port in Firewall* and click *Next*.

4. Select *Add Directory* and browse for the directory containing the installation sources, in this case, *`productversion`*.

5. Select *Add Host* and enter the hostnames of the machines to which to export the installation data. Instead of specifying hostnames here, you could also use wild cards, ranges of network addresses, or just the domain name of your network. Enter the appropriate export options or leave the default, which works fine in most setups. For more information about the syntax used in exporting NFS shares, read the `exports` man page.

6. Click *Finish*. The NFS server holding the openSUSE installation sources is automatically started and integrated into the boot process.

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

1. Log in as `root`.

2. Open the file `/etc/exports` and enter the following line:

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

   This exports the directory `/productversion` to any host that is part of this network or to any host that can connect to this server. To limit the access to this server, use netmasks or domain names instead of the general wild card `*`. Refer to the `export` man page for details. Save and exit this configuration file.

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

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

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

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

1. Log in as `root`.

2. Enter the directory `/etc/slp.reg.d/`.

3 Create a configuration file called `install.suse.nfs.reg` containing the following lines:

```
# Register the NFS Installation Server
service:install.suse:nfs://$HOSTNAME/path_to_instsource/CD1,en,65535
description=NFS Installation Source
```

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

4 Save this configuration file and start the OpenSLP daemon with `rcslpd start`.

For more information about OpenSLP, refer to the package documentation located under `/usr/share/doc/packages/openslp/` or refer to Chapter 20, *SLP Services in the Network* (page 321). More Information about NFS is found in Chapter 24, *Sharing File Systems with NFS* (page 367).

1.2.3 Setting Up an FTP Installation Source Manually

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

1 Create a directory holding the installation sources as described in Section 1.2.2, "Setting Up an NFS Installation Source Manually" (page 14).

2 Configure the FTP server to distribute the contents of your installation directory:

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

2b Enter the FTP server root directory:

```
cd /srv/ftp
```

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

```
mkdir instsource
```

Replace *instsource* with the product name.

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

```
mount --bind path_to_instsource /srv/ftp/instsource
```

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

2e Start vsftpd with `vsftpd`.

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

3a Create a configuration file called `install.suse.ftp.reg` under `/etc/slp.reg.d/` that contains the following lines:

```
# Register the FTP Installation Server
service:install.suse:ftp://$HOSTNAME/srv/ftp/instsource/CD1,en,65535
description=FTP Installation Source
```

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

3b Save this configuration file and start the OpenSLP daemon with `rcslpd start`.

TIP: Configuring an FTP Server with YaST

If you prefer using YaST over manually configuring the FTP installation server, refer to Chapter 27, *Setting up a FTP server with YaST* (page 435) for more information on how to use the YaST FTP server module.

1.2.4 Setting Up an HTTP Installation Source Manually

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

1 Create a directory holding the installation sources as described in Section 1.2.2, "Setting Up an NFS Installation Source Manually" (page 14).

2 Configure the HTTP server to distribute the contents of your installation directory:

2a Install the Web server Apache as described in Section 26.1.2, "Installation" (page 394).

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

```
mkdir instsource
```

Replace *`instsource`* with the product name.

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

```
ln -s /path_instsource /srv/www/htdocs/instsource
```

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

```
Options None
```

with

```
Options Indexes FollowSymLinks
```

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

3. Announce the installation source via OpenSLP, if this is supported by your network setup:

   3a. Create a configuration file called `install.suse.http.reg` under `/etc/slp.reg.d/` that contains the following lines:

```
# Register the HTTP Installation Server
service:install.suse:http://$HOSTNAME/srv/www/htdocs/instsource/CD1/,en,65535
description=HTTP Installation Source
```

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

   3b. Save this configuration file and start the OpenSLP daemon using `rcslpd restart`.

1.2.5 Managing an SMB Installation Source

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

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

1. Log in to your Windows machine.

2. Start Explorer and create a new folder that will hold the entire installation tree and name it `INSTALL`, for example.

3. Export this share according the procedure outlined in your Windows documentation.

4. Enter this share and create a subfolder, called *`product`*. Replace *`product`* with the actual product name.

5. Enter the `INSTALL/`*`product`* folder and copy each CD or DVD to a separate folder, such as `CD1` and `CD2`.

To use a SMB mounted share as installation source, proceed as follows:

1. Boot the installation target.

2. Select *Installation*.

3. Press F4 for a selection of installation sources.

4. Choose SMB and enter the Windows machine's name or IP address, the share name (`INSTALL/`*`product`*`/CD1`, in this example), username, and password.

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

1.2.6 Using ISO Images of the Installation Media on the Server

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

1. Download the ISO images and save them to the machine to use as the installation server.

2. Log in as `root`.

3. Choose and create an appropriate location for the installation data, as described in Section 1.2.2, "Setting Up an NFS Installation Source Manually" (page 14), Section 1.2.3, "Setting Up an FTP Installation Source Manually" (page 17), or Section 1.2.4, "Setting Up an HTTP Installation Source Manually" (page 19).

4. Create subdirectories for each CD or DVD.

5. To mount and unpack each ISO image to the final location, issue the following command:

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

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

6 Repeat the previous step to mount all ISO images needed for your product.

7 Start your installation server as usual, as described in Section 1.2.2, "Setting Up an NFS Installation Source Manually" (page 14), Section 1.2.3, "Setting Up an FTP Installation Source Manually" (page 17), or Section 1.2.4, "Setting Up an HTTP Installation Source Manually" (page 19).

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

```
path_to_iso path_to_instsource/product
    medium auto loop
```

1.3 Preparing the Boot of the Target System

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

1.3.1 Setting Up a DHCP Server

There are two ways to set up a DHCP server. For openSUSE, YaST provides a graphical interface to the process. Users can also manually edit the configuration files. For more information about DHCP servers, see also Chapter 22, *DHCP* (page 349).

Setting Up a DHCP Server with YaST

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

1. Log in as `root` to the machine hosting the DHCP server.

2. Start *YaST* > *Network Services* > *DHCP Server*.

3. Complete the setup wizard for basic DHCP server setup.

4. Select *Expert Settings* and select *Yes* when warned about leaving the start-up dialog.

5. In the *Configured Declarations* dialog, select the subnet in which the new system should be located and click *Edit*.

6. In the *Subnet Configuration* dialog select *Add* to add a new option to the subnet's configuration.

7. Select `filename` and enter `pxelinux.0` as the value.

8. Add another option (`next-server`) and set its value to the address of the TFTP server.

9. Select *OK* and *Finish* to complete the DHCP server configuration.

To configure DHCP to provide a static IP address to a specific host, enter the *Expert Settings* of the DHCP server configuration module (Step 4 (page 23)) and add a new declaration of the host type. Add the options `hardware` and `fixed-address` to this host declaration and provide the appropriate values.

Setting Up a DHCP Server Manually

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

1. Log in as `root` to the machine hosting the DHCP server.

2 Append the following lines to a subnet configuration of your DHCP server's configuration file located under `/etc/dhcpd.conf`:

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

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

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

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

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

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

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

1.3.2 Setting Up a TFTP Server

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

Setting Up a TFTP Server Using YaST

1. Log in as `root`.
2. Install the `yast2-tftp-server` package.
3. Start *YaST* > *Network Services* > *TFTP Server* and install the requested package.
4. Click *Enable* to make sure that the server is started and included in the boot routines. No further action from your side is required to secure this. xinetd starts tftpd at boot time.
5. Click *Open Port in Firewall* to open the appropriate port in the firewall running on your machine. If there is no firewall running on your server, this option is not available.
6. Click *Browse* to browse for the boot image directory. The default directory `/tftpboot` is created and selected automatically.
7. Click *Finish* to apply your settings and start the server.

Setting Up a TFTP Server Manually

1. Log in as `root` and install the packages `tftp` and `xinetd`.
2. If unavailable, create `/srv/tftpboot` and `/srv/tftpboot/pxelinux.cfg` directories.

3. Add the appropriate files needed for the boot image as described in Section 1.3.3, “Using PXE Boot” (page 26).

4. Modify the configuration of xinetd located under `/etc/xinetd.d/` to make sure that the TFTP server is started on boot:

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

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

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

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

1.3.3 Using PXE Boot

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

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

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

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

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

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

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

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

5 Edit the `/srv/tftpboot/pxelinux.cfg/default` file and remove the lines beginning with `gfxboot`, `readinfo`, and `framebuffer`.

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

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

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

`install=nfs://`*`ip_instserver`*`/`*`path_instsource`*`/CD1`
: This entry defines the NFS server and the installation source for the client installation. Replace *`ip_instserver`* with the actual IP address of your installation server. *`path_instsource`* should be replaced with the actual path to the installation sources. HTTP, FTP, or SMB sources are addressed in a similar manner, except for the protocol prefix, which should read `http`, `ftp`, or `smb`.

IMPORTANT

If you need to pass other boot options to the installation routines, such as SSH or VNC boot parameters, append them to the `install` entry. An overview of parameters and some examples are given in Section 1.4, "Booting the Target System for Installation" (page 33).

TIP: Changing Kernel and Initrd Filenames

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

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

```
default harddisk

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

# repair
label repair
  kernel linux
  append initrd=initrd splash=silent repair=1 showopts

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

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

```
#  memory test
label memtest
  kernel memtest

# hard disk
label harddisk
  localboot 0

implicit     0
display      message
prompt       1
timeout      100
```

Replace *`ip_instserver`* and *`path_instsource`* with the values used in your setup.

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

1.3.4 PXELINUX Configuration Options

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

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

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

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

`LABEL` *`label`* `KERNEL` *`image`* `APPEND` *`options`*`...`

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

Note that GRUB uses the following syntax:

```
title mytitle
  kernel my_kernel my_kernel_options
  initrd myinitrd
```

PXELINUX uses the following syntax:

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

Labels are mangled as if they were filenames and they must be unique after mangling. For example, the two labels “v2.1.30” and “v2.1.31” would not be distinguishable under PXELINUX because both mangle to the same DOS filename.

The kernel does not have to be a Linux kernel; it can be a boot sector or a COMBOOT file.

`APPEND -`

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

`LOCALBOOT` *`type`*

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

Argument	Description
`0`	Perform a normal boot
`4`	Perform a local boot with the Universal Network Driver Interface (UNDI) driver still resident in memory

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

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

`TIMEOUT` *`time-out`*

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

`PROMPT` *`flag_val`*

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

```
F2  filename
F1  filename
..etc...
F9  filename
F10 filename
```

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

1.3.5 Preparing the Target System for PXE Boot

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

WARNING: BIOS Boot Order

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

1.3.6 Preparing the Target System for Wake on LAN

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

1.3.7 Wake on LAN

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

IMPORTANT: Wake on LAN across Different Network Segments

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

1.4 Booting the Target System for Installation

Basically, there are two different ways to customize the boot process for installation apart from those mentioned under Section 1.3.7, "Wake on LAN" (page 32) and Section 1.3.3, "Using PXE Boot" (page 26). You can either use the default boot options and function keys or use the boot options prompt of the installation boot screen to pass any boot options that the installation kernel might need on this particular hardware.

1.4.1 Using the Default Boot Options

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

If problems occur, use *Installation—ACPI Disabled* or *Installation—Safe Settings*. For more information about troubleshooting the installation process, refer to Section "Installation Problems" (Chapter 13, *Common Problems and Their Solutions*, ↑Start-Up).

The menu bar at the bottom screen offers some advanced functionality needed in some setups. Using the F keys, you can specify additional options to pass to the installation routines without having to know the detailed syntax of these parameters (see Section 1.4.2, "Using Custom Boot Options" (page 33)). A detailed description of the available function keys is available at Section "The Boot Screen" (Chapter 1, *Installation with YaST*, ↑Start-Up).

1.4.2 Using Custom Boot Options

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

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

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

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

***Table 1.1** Installation (Boot) Scenarios Used in This Chapter*

Installation Scenario	Parameters Needed for Booting	Boot Options
Chapter 1, *Installation with YaST* (↑Start-Up)	None: system boots automatically	None needed
Section 1.1.1, "Simple Remote Installation via VNC—Static Network Configuration" (page 4)	• Location of the installation server • Network device • IP address • Netmask • Gateway • VNC enablement • VNC password	• `install=(nfs,http,ftp,smb):///path_to_instmedia` • `netdevice=some_netdevice` (only needed if several network devices are available) • `hostip=some_ip` • `netmask=some_netmask` • `gateway=ip_gateway` • `vnc=1` • `vncpassword=some_password`
Section 1.1.2, "Simple Remote Installation via VNC—Dynamic Network Configuration" (page 5)	• Location of the installation server • VNC enablement • VNC password	• `install=(nfs,http,ftp,smb):///path_to_instmedia` • `vnc=1` • `vncpassword=some_password`
Section 1.1.3, "Remote Installation via	• Location of the installation server	Not applicable; process managed through PXE and DHCP

Installation Scenario	Parameters Needed for Booting	Boot Options
VNC—PXE Boot and Wake on LAN” (page 7)	• Location of the TFTP server • VNC enablement • VNC password	
Section 1.1.4, “Simple Remote Installation via SSH—Static Network Configuration” (page 8)	• Location of the installation server • Network device • IP address • Netmask • Gateway • SSH enablement • SSH password	• `install=(nfs,http,ftp,smb):///`*`path_to_instmedia`* • `netdevice=`*`some_netdevice`* (only needed if several network devices are available) • `hostip=`*`some_ip`* • `netmask=`*`some_netmask`* • `gateway=`*`ip_gateway`* • `usessh=1` • `sshpassword=`*`some_password`*
Section 1.1.5, “Simple Remote Installation via SSH—Dynamic Network Configuration” (page 9)	• Location of the installation server • SSH enablement • SSH password	• `install=(nfs,http,ftp,smb):///`*`path_to_instmedia`* • `usessh=1` • `sshpassword=`*`some_password`*
Section 1.1.6, “Remote Installation via SSH—PXE Boot and Wake on LAN” (page 11)	• Location of the installation server • Location of the TFTP server • SSH enablement	Not applicable; process managed through PXE and DHCP

Installation Scenario	Parameters Needed for Booting	Boot Options
	• SSH password	

TIP: More Information about linuxrc Boot Options

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

1.5 Monitoring the Installation Process

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

1.5.1 VNC Installation

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

Preparing for VNC Installation

All you need to do on the installation target to prepare for a VNC installation is to provide the appropriate boot options at the initial boot for installation (see Section 1.4.2, “Using Custom Boot Options” (page 33)). The target system boots into a text-based environment and waits for a VNC client to connect to the installation program.

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

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

1 Start the KDE file and Web browser Konqueror.

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

Connecting to the Installation Program

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

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

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

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

1 Start the VNC viewer.

2 Enter the IP address and display number of the installation target as provided by the SLP browser or the installation program itself:

```
ip_address:display_number
```

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

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

To perform a VNC installation, proceed as follows:

1. Launch your preferred Web browser.

2. Enter the following at the address prompt:

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

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

1.5.2 SSH Installation

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

Preparing for SSH Installation

Apart from installing the appropriate software package (OpenSSH for Linux and PuTTY for Windows), you just need to pass the appropriate boot options to enable SSH for installation. See Section 1.4.2, "Using Custom Boot Options" (page 33) for details. OpenSSH is installed by default on any SUSE Linux–based operating system.

Connecting to the Installation Program

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

2. At a command line, enter the following command:

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

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

3. When prompted for a username, enter `root`.

4. When prompted for the password, enter the password that has been set with the SSH boot option. After you have successfully authenticated, a command line prompt for the installation target appears.

5. Enter `yast` to launch the installation program. A window opens showing the normal YaST screens as described in Chapter 1, *Installation with YaST* (↑Start-Up).

Advanced Disk Setup

2

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

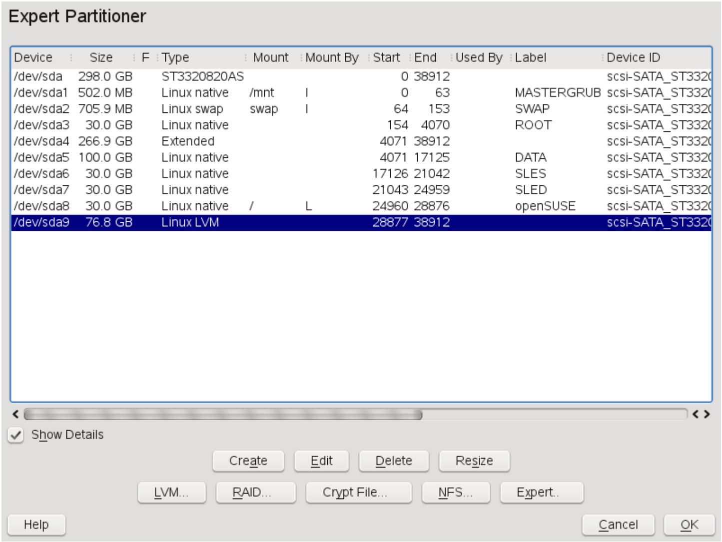
2.1 Using the YaST Partitioner

With the expert partitioner, shown in Figure 2.1, “The YaST Partitioner” (page 42), manually modify the partitioning of one or several hard disks. Partitions can be added, deleted, resized, and edited. Also access the soft RAID and LVM configuration from this YaST module.

WARNING: Repartitioning the Running System

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

Figure 2.1 The YaST Partitioner



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

Several functional views are available on the lefthand *System View*. Use these views to gather information about existing storage configurations, or to configure functions like `RAID`, `Volume Management`, `Crypt Files`, or `NFS`.

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

2.1.1 Partition Types

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

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

If you need more than four partitions, create an extended partition as the fourth partition or earlier. This extended partition should span the entire remaining free cylinder range. Then create multiple logical partitions within the extended partition. The maximum number of logical partitions is 15 on SCSI, SATA, and Firewire disks and 63 on (E)IDE disks. It does not matter which types of partitions are used for Linux. Primary and logical partitions both work fine.

2.1.2 Creating a Partition

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

1. Select *Add*. If several hard disks are connected, a selection dialog appears in which to select a hard disk for the new partition.

2. Specify the partition type (primary or extended). Create up to four primary partitions or up to three primary partitions and one extended partition. Within the extended partition, create several logical partitions (see Section 2.1.1, "Partition Types" (page 43)).

3. Select the file system to use and a mount point. YaST suggests a mount point for each partition created. To use a different mount method, like mount by label, select *Fstab Options*.

4 Specify additional file system options if your setup requires them. This is necessary, for example, if you need persistent device names. For details on the available options, refer to Section 2.1.3, "Editing a Partition" (page 44).

5 Click *OK* > *Apply* to apply your partitioning setup and leave the partitioning module.

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

2.1.3 Editing a Partition

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

1 Select the partition.

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

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

   File System
      Change the file system or format the partition here. Changing the file system or reformatting partitions irreversibly deletes all data from the partition.

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

      Ext3 is the default file system for the Linux partitions. ReiserFS, JFS, XFS, and Ext3 are journaling file systems. These file systems are able to restore the system very quickly after a system crash, because write processes are logged during the operation. Furthermore, ReiserFS is very fast in handling lots of small files. Ext2 is not a journaling file system. However, it is rock solid and good for smaller partitions, because it does not require much disk space for management.

Encrypt File System
: If you activate the encryption, all data is written to the hard disk in encrypted form. This increases the security of sensitive data, but slightly reduces the system speed, because the encryption takes some time. More information about the encryption of file systems is provided in Chapter 11, *Encrypting Partitions and Files* (↑Security Guide).

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

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

  When using the mount option *LABEL* to mount a partition, define an appropriate label for the selected partition. For example, you could use the partition label `HOME` for a partition intended to mount to `/home`.

  If you intend to use quota on the file system, use the mount option *Enable Quota Support*. This must be done before you can define quotas for users in the YaST *User Management* module. For further information on how to configure user quota, refer to Section “Managing Quotas” (Chapter 5, *Managing Users with YaST*, ↑Start-Up).

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

3 Select *OK* > *Apply* to activate the partition.

NOTE: Resize Filesystems

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

2.1.4 More Partitioning Tips

The following section comprises a few hints and tips on partitioning that should help you in taking the right decisions while setting up your system.

TIP: Cylinder Numbers

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

Using `swap`

Swap is used to extend the physically available memory. This makes it possible to use more memory than physical ram available. The memory management system of kernels before 2.4.10 needed swap as a safety measure. In those times, if you did not have twice the size of your ram in swap, the performance of the system suffered. This does not hold true anymore as these limitations no longer exist.

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

If an application tries to allocate as much memory as it can possibly get, there are some problems with swap. There are three major cases to look at:

System with no swap
: The application gets all memory that can be freed by any means. All caches are freed, and thus all other applications are slowed down. After a few minutes, the out of memory killer mechanism of the kernel will become active and kill the process.

System with medium sized swap (128 MB–512 MB)
: At first, the system is slowed down like a system without swap. After all physical ram has been used up, swap space is used as well. At this point, the system becomes very slow and it becomes impossible to run commands from remote. Depending on the speed of the hard disks that run the swap space, the system stays in this condition for about 10 to 15 minutes until the out of memory killer of the kernel resolves the issue. Note, that you will need a certain amount of swap if the computer should perform a "suspend to disk". In that case, the swap size should be reasonably big to contain the necessary data from memory (512 MB–1GB).

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

If your system does not run wild, but needs more swap after some time, it is possible to extend the swap space online. If you prepared a partition for swap space, just add this partition with YaST. If you do not have a partition available, you may also just use a swap file to extend the swap. Swap files are generally slower than partitions, but compared to physical ram, both are extremely slow and the actual speed difference is not as important as one would think in the first place.

Procedure 2.1 *Adding a Swap File Manually*

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

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

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

2 Initialize this swap file with the command

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

3. Activate the swap with the command

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

   To disable this swap file, use the command

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

4. Check the current available swap spaces with the command

   ```
   cat /proc/swaps
   ```

   Note, that at this point this is only temporary swap space. After the next reboot, it is not used anymore.

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

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

2.1.5 Partitioning and LVM

From the expert partitioner, access the LVM configuration with *Volume Management*. However, if a working LVM configuration already exists on your system, it is automatically activated as soon as you enter the LVM configuration for the first time in a session. In this case, any disks containing a partition belonging to an activated volume group cannot be repartitioned because the Linux kernel cannot reread the modified partition table of a hard disk when any partition on this disk is in use. However, if you already have a functioning LVM configuration on your system, physical repartitioning should not be necessary. Instead, change the configuration of the logical volumes.

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

WARNING: File System for Booting

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

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

2.2 LVM Configuration

This section briefly describes the principles behind the Logical Volume Manager (LVM) and its basic features that make it useful under many circumstances. In Section 2.2.2, "LVM Configuration with YaST" (page 51), learn how to set up LVM with YaST.

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

2.2.1 The Logical Volume Manager

The LVM enables flexible distribution of hard disk space over several file systems. It was developed because sometimes the need to change the segmentation of hard disk space arises only after the initial partitioning during installation has already been done. Because it is difficult to modify partitions on a running system, LVM provides a virtual pool (volume group, VG for short) of memory space from which logical volumes (LVs) can be created as needed. The operating system accesses these LVs instead of the physical partitions. Volume groups can span more than only one disk so that several disks or parts of them may constitute one single VG. This way, LVM provides a kind of abstraction from the physical disk space that allows its segmentation to be changed in a much easier and safer way than physical repartitioning does. Background information regarding physical partitioning can be found in Section 2.1.1, "Partition Types" (page 43) and Section 2.1, "Using the YaST Partitioner" (page 41).

Figure 2.2 Physical Partitioning versus LVM

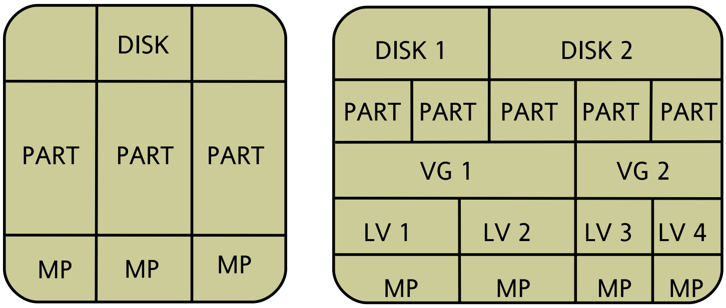


Figure 2.2, "Physical Partitioning versus LVM" (page 50) compares physical partitioning (left) with LVM segmentation (right). On the left side, one single disk has been divided into three physical partitions (PART), each with a mount point (MP) assigned so that the operating system can access them. On the right side, two disks have been divided into two and three physical partitions each. Two LVM volume groups (VG 1 and VG 2) have been defined. VG 1 contains two partitions from DISK 1 and one from DISK 2. VG 2 contains the remaining two partitions from DISK 2. In LVM, the physical disk partitions that are incorporated in a volume group are called physical volumes (PVs). Within the volume groups, four LVs (LV 1 through LV 4) have been defined, which can be used by the operating system via the associated mount points. The border between different LVs need not be aligned with any partition border. See the border between LV 1 and LV 2 in this example.

LVM features:

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

- It is possible to activate a "striping mode" that distributes the data stream of a LV over several PVs. If these PVs reside on different disks, this can improve the reading and writing performance just like RAID 0.

- The snapshot feature enables consistent backups (especially for servers) in the running system.

With these features, using LVM already makes sense for heavily used home PCs or small servers. If you have a growing data stock, as in the case of databases, music archives, or user directories, LVM is just the right thing for you. This would allow file systems that are larger than the physical hard disk. Another advantage of LVM is that up to 256 LVs can be added. However, keep in mind that working with LVM is different from working with conventional partitions. Instructions and further information about configuring LVM is available in the official LVM HOWTO at `http://tldp.org/HOWTO/LVM-HOWTO/`.

Starting from kernel version 2.6, LVM version 2 is available, which is downward-compatible with the previous LVM and enables the continued management of old volume groups. When creating new volume groups, decide whether to use the new format or the downward-compatible version. LVM 2 does not require any kernel patches. It makes use of the device mapper integrated in kernel 2.6. This kernel only supports LVM version 2. Therefore, when talking about LVM, this section always refers to LVM version 2.

2.2.2 LVM Configuration with YaST

The YaST LVM configuration can be reached from the YaST Expert Partitioner (see Section 2.1, "Using the YaST Partitioner" (page 41)) below *Volume Management*. The Expert Partitioner allows you to edit and delete existing partitions and also create new ones that should be used with LVM. The first task is to create PVs that provide space to a volume group:

1. Select a hard disk from *Hard Disks*.

2. Change to the *Partitions* tab.

3. Click *Add* and enter the desired size of the PV on this disk.

4. Use *Do not Format the Partition* and change the *File System ID* to *0x8E Linux LVM*. Do not mount this partition.

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

Creating Volume Groups

If no volume group exists on your system yet, you have to add one (see Figure 2.3, "Creating a Volume Group" (page 53)). It is possible to create additional groups with *Add Volume Group*, but usually one single volume group is sufficient.

1. Enter a name for the VG, e.g. `system`.

2. Select the desired *Physical Extend Size*. This value defines the size of a physical block in the volume group. All the disk space in a volume group is handled in chunks of this size.

   TIP: Logical Volumes and Block Sizes

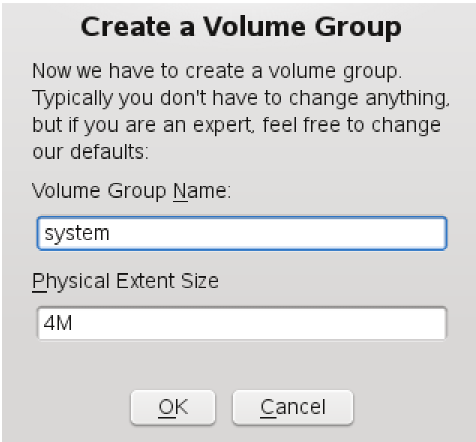
   The possible size of a LV depends on the block size used in the volume group. The default is 4 MB and allows for a maximum size of 256 GB for physical and LVs. The physical extent size should be increased, for example, to 8, 16, or 32 MB, if you need LVs larger than 256 GB.

   ---

3. Add the prepared PVs to the VG by selecting the device and clicking on *Add*. Selecting several devices is possible by holding *Strg* pressed while selecting the devices.

4. Select *Finish* to make the VG available to further configuration steps.

Figure 2.3 Creating a Volume Group

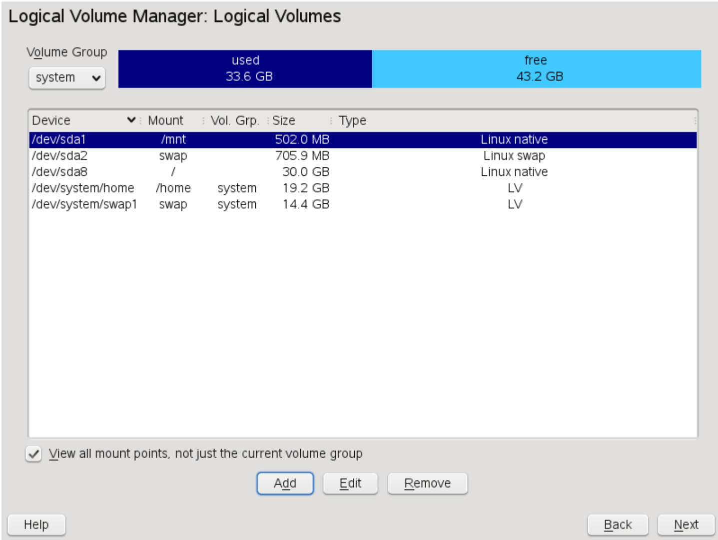


If you have multiple volume groups defined, and want to add or remove PVs, select the volume group in *Volume Management*. Then change to the *Overview* tab and select *Resize*. In the following menu, you can add or remove PVs to the selected volume group.

Configuring Logical Volumes

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

Figure 2.4 *Logical Volume Management*



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

1. Enter the name of the LV. For a partition that should be mounted to `/home`, a selfexplaining name like `HOME` could be used.

2. Select the size and the number of stripes of the LV. If you have only one PV, selecting more than one stripes is not useful.

3. Choose the filesystem to use on the LV as well as the mount point.

By using stripes it is possible to distribute the data stream in the LV among several PVs (striping). If these PVs reside on different hard disks, this generally results in a better reading and writing performance (like RAID 0). However, a striping LV with `n` stripes can only be created correctly if the hard disk space required by the LV can be distributed evenly to `n` PVs. If, for example, only two PVs are available, a LV with three stripes is impossible.

WARNING: Striping

YaST has no chance at this point to verify the correctness of your entries concerning striping. Any mistake made here is apparent only later when the LVM is implemented on disk.

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

2.3 Soft RAID Configuration

The purpose of RAID (redundant array of independent disks) is to combine several hard disk partitions into one large *virtual* hard disk to optimize performance, data security, or both. Most RAID controllers use the SCSI protocol because it can address a larger number of hard disks in a more effective way than the IDE protocol and is more suitable for parallel processing of commands. There are some RAID controllers that support IDE or SATA hard disks. Soft RAID provides the advantages of RAID systems without the additional cost of hardware RAID controllers. However, this requires some CPU time and has memory requirements that make it unsuitable for real high performance computers.

openSUSE® offers the option of combining several hard disks into one soft RAID system with the help. RAID implies several strategies for combining several hard disks in a RAID system, each with different goals, advantages, and characteristics. These variations are commonly known as *RAID levels*.

Common RAID levels are:

RAID 0
: This level improves the performance of your data access by spreading out blocks of each file across multiple disk drives. Actually, this is not really a RAID, because it does not provide data backup, but the name *RAID 0* for this type of system has become the norm. With RAID 0, two or more hard disks are pooled together. The performance is very good, but the RAID system is destroyed and your data lost if even one hard disk fails.

RAID 1
: This level provides adequate security for your data, because the data is copied to another hard disk 1:1. This is known as *hard disk mirroring*. If a disk is destroyed, a copy of its contents is available on another one. All of them except one could be damaged without endangering your data. However, if damage is not detected, it also may happen that damaged data is mirrored to the correct disk and data corruption happens that way. The writing performance suffers a little in the copying process compared to when using single disk access (10 to 20 % slower), but read access is significantly faster in comparison to any one of the normal physical hard disks, because the data is duplicated so can be parallel scanned. Generally it can be said that Level 1 provides nearly twice the read transaction rate of single disks and almost the same write transaction rate as single disks.

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

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

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

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

2.3.1 Soft RAID Configuration with YaST

The YaST *RAID* configuration can be reached from the YaST Expert Partitioner, described in Section 2.1, “Using the YaST Partitioner” (page 41). This partitioning tool enables you to edit and delete existing partitions and create new ones that should be used with soft RAID. There, create RAID partitions:

1 Select a hard disk from *Hard Disks*.

2 Change to the *Partitions* tab.

3 Click *Add* and enter the desired size of the raid partition on this disk.

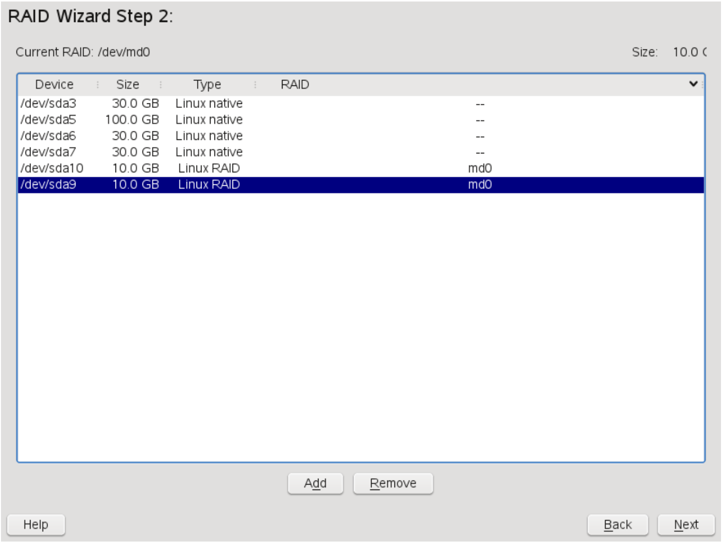
4 Use *Do not Format the Partition* and change the *File System ID* to *0xFD Linux RAID*. Do not mount this partition.

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

For RAID 0 and RAID 1, at least two partitions are needed—for RAID 1, usually exactly two and no more. If RAID 5 is used, at least three partitions are required. It is recommended to take only partitions of the same size. The RAID partitions should be located on different hard disks to decrease the risk of losing data if one is defective (RAID 1 and 5) and to optimize the performance of RAID 0. After creating all the partitions to use with RAID, click *RAID* > *Add RAID* to start the RAID configuration.

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

Figure 2.5 RAID Partitions



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

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

2.3.2 Troubleshooting

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

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

2.3.3 For More Information

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

- `/usr/share/doc/packages/mdadm/Software-RAID.HOWTO.html`
- `http://en.tldp.org/HOWTO/Software-RAID-HOWTO.html`

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

Part II. Managing and Updating Software

Installing or Removing Software

3

Change the software collection of your system using YaST. This YaST module is available in three toolkit flavors: Qt, GTK+, and ncurses; Qt and GTK+ flavors are described here.

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

3.1 Definition of Terms

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

(Repository) Alias
: A short name for a repository used by various zypper commands. The alias can be chosen by the user when adding a repository and has to be unique.

Product
: Represents a whole product, for example openSUSE.

Pattern
: A pattern is an installable list of packages needed for a special purpose. Examples are `Base System`, providing the openSUSE basic system, or `GNOME Base System`, containing all packages needed to run the GNOME Desktop environment.

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

Patch
: A patch consists of one or more packages—either full packages or patchrpm or deltarpm packages— and may also introduce dependencies to packages that are not installed yet.

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

patchrpm
: A patchrpm consists only of files that have been updated since it was first released for openSUSE 11.1. Its download size is usually considerably smaller than the size of a package.

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

3.2 Using the Qt Interface

Start the YaST Qt interface on the command line with `yast2 --qt`.

3.2.1 Installing Software

Software is available via RPM packages. Each package contains the program itself, the configuration files, and additional documentation. If you want to add more software to the system, proceed as follows:

1. Click *Software* > *Software Management* to start the YaST package manager.

2. In the search field enter the name of the software you want to install (for example, `xpdf`, a lightweight PDF viewer) and press Enter.

3. The package is listed in the right frame. Select it for installation. Once done, you can search for more packages and select them for installation in one go.

4. Click *Accept*.

5. When all selected packages are installed, YaST asks you whether you want to install or remove additional packages. Press *No* to close YaST.

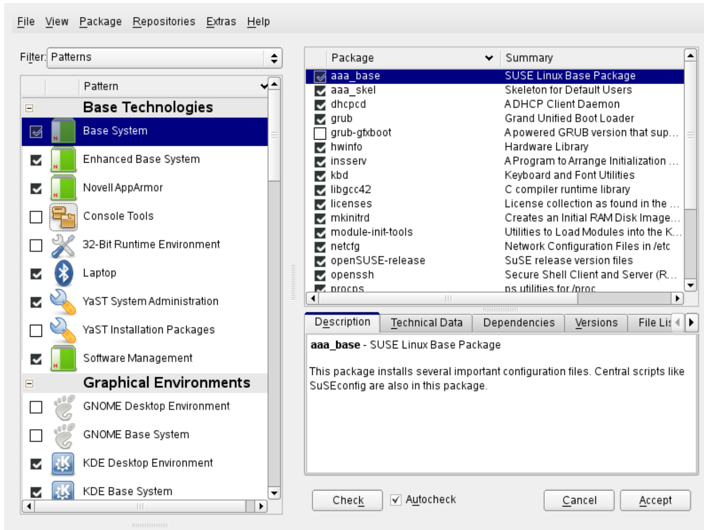
By specifying various search criteria, you can restrict the search to display a few or even only one package. You can also define special search patterns using wild cards and regular expressions in *Search Mode*.

TIP: Quick Search

In addition to the *Search* filter, all lists of the package manager feature a quick search. Click the respective list to gain focus (for example, the *Package* list) and enter a letter to move the cursor to the first package in the list whose name begins with this letter.

If you do not know the name of the software you are interested in, you can browse through the software catalog in various ways. For example, you can filter by patterns, package groups, languages, repositories, or installation summary. Filter by patterns, if you are looking for software for a specific task:

1. From the filter list in the upper left corner, select *Patterns*. Now you see various pattern sets listed in the area below.



2 From the patterns select one or more patterns you are interested in. If you click on the name of a pattern, for example on *Base Development*, you see the packages it contains, in the right frame. If you activate it, the status markers at the beginning of the line will change: all packages get marked either with the status *Keep* or *Install*. The meaning of all the symbols and font color changes is explained in *Help* > *Symbols*.

3 Click *Accept*.

Alternatively, filter by package groups. The package groups feature offers a more detailed view of the software grouped by categories. Often packages depend on other packages; if you select a package, you might be requested to install additional packages to resolve possible package dependencies.

Filtering by languages is similar to filtering by package groups. Using the languages view enables you to select packages like translated program messages, documentation, or special fonts which are needed to support a specific language.

For installing corresponding source packages, use `zypper`. For more information, see Section 7.1.2, “Installing and Removing Software with Zypper” (page 88).

Using the installation summary filter you see an overview of the packages you have scheduled for installation. It is convenient for double-checking if many packages are pending for installation.

3.2.2 Checking Software Dependencies

The software of one package might only work properly if the required package is also installed. If packages with identical or similar functionality use the same system resource, they should not be installed at the same time, because this would cause a package conflict.

When the package manager starts, it examines the system and displays the installed packages. When you select to install and remove packages, the package manager automatically checks the dependencies and selects any other required packages (resolution of dependencies). If you select or deselect conflicting packages, the package manager indicates this and suggests possible solutions to this problem (resolution of conflicts).

Check Dependencies and *Autocheck* are located under the information window. If you click *Check Dependencies*, the package manager checks if the current package selection results in any unresolved package dependencies or conflicts. In the event of unresolved dependencies, the required additional packages are selected automatically. For package conflicts, the package manager opens a dialog that shows the conflict and offers various options for solving the problem.

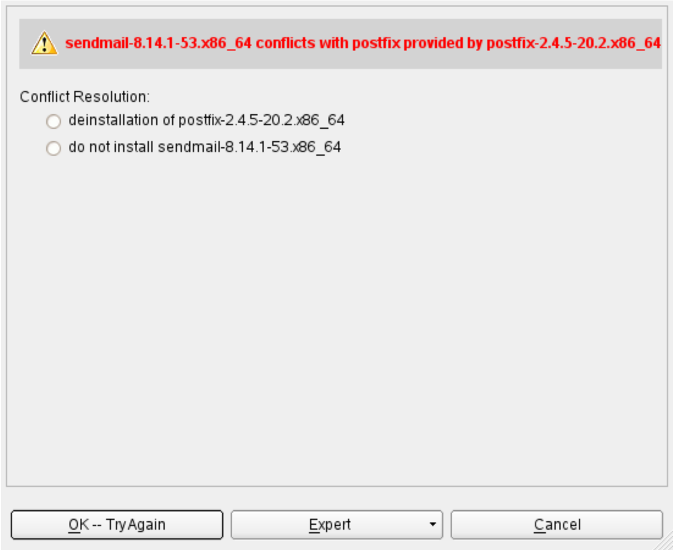
If you activate *Autocheck*, any change of a package status triggers an automatic check. This is a useful feature, because the consistency of the package selection is monitored permanently. However, this process consumes resources and can slow down the package manager. For this reason, *Autocheck* is not activated by default. In either case, a consistency check is performed when you confirm your selection with *Accept*.

For example, `sendmail` and `postfix` may not be installed concurrently. Figure 3.1, "Conflict Management of the Package Manager" (page 68) shows the conflict message prompting you to take a decision. `postfix` is already installed. Accordingly, you can refrain from installing `sendmail`, remove `postfix`, or take the risk and ignore the conflict.

WARNING: Handling Package Conflicts

Unless you are very experienced, follow the suggestions of YaST when handling package conflicts, because otherwise the stability and functionality of your system could be endangered by the existing conflict.

Figure 3.1 *Conflict Management of the Package Manager*



3.2.3 Packages and Software Repositories

If you want to search for packages originating from one particular software repository, use the *Repositories* filter. In the default configuration, this filter shows a list of all packages from the selected installation source. To restrict the list, use a secondary filter.

To view a list of the all installed packages from the selected repository, select the filter *Repositories* then select *Installation Summary* from *Secondary Filter* list and deactivate all check boxes except *Keep*.

If you are interested in the opposite and want to detect packages not belonging to any repository, also use the *Repositories* filter and select *Unmaintained Packages* as the *Secondary Filter*.

3.2.4 Removing Software

If you want to remove software from the system, proceed as follows:

1 Make use of a search strategy explained in Section 3.2.1, "Installing Software" (page 64).

2 Depending on the search strategy, you can either select a complete set or single packages one by one. For patterns, both ways are possible.

3 Click *Accept* and either watch the de-installation progress or adjust your selection, if YaST complains about dependency issues.

3.3 Using the GTK+ Interface

Change the software collection of your system using YaST. In YaST's software management tool search for software components you want to add or remove. YaST resolves all the dependencies for you. Add additional software repositories to your setup to install packages not shipped with the installation media and let YaST manage them. Keep your system up-to-date by managing software updates with openSUSE Updater.

Start the YaST GTK+ interface on the command line with `yast2 --gtk`.

3.3.1 Installing Software

Software is available via RPM packages. Each package contains the program itself, the configuration files, and additional documentation. If you want to add more software to the system, proceed as follows:

1 Click *Software* > *Software Management* to start the YaST package manager.

2 In the package search field at the right window border enter the name of the software you want to install (for example, `xpdf`, a leightweight PDF viewer). YaST starts searching for the package while you enter the name. When the search is finished, select the desired package in the main pane and click *Install*.

3 You are able to search for more packages and list them the same way.

4 When finished, click *Apply* to perform the installation of the listed packages.

If you do not know the name of the software you are interested in, you can browse through the software catalog in various ways. For example, you can group by patterns, package groups, languages, or repositories. Group by patterns, if you are looking for software for a specific task:

1 From the grouping menu in the upper left corner, select *Patterns*. Now you see various pattern sets listed in the area below.

2 From the patterns select one or more patterns you are interested in. If you click the name of a pattern, for example, *Base Development*, you see the packages it contains in the right frame. If you activate this pattern by clicking *Install All*, the packages will get listed in the changes overview on the right side.

3 Click *Apply* to install all selected packages.

Alternatively, group by package groups. The package groups feature offers a more detailed view of the software grouped by categories. Packages often depend on other packages; if you select a package, you might be requested to install additional packages to resolve possible package dependencies.

Grouping by languages is similar to grouping by package groups. Using the languages view enables you to select packages like translated program messages, documentation, or special fonts which are needed to support a specific language.

For installing corresponding source packages, use `zypper`. For more information, see Section 7.1.2, "Installing and Removing Software with Zypper" (page 88).

3.3.2 Checking Software Dependencies

The software of one package might only work properly if the required package is also installed. If packages with identical or similar functionality use the same system resource, they should not be installed at the same time, because this would cause a package conflict.

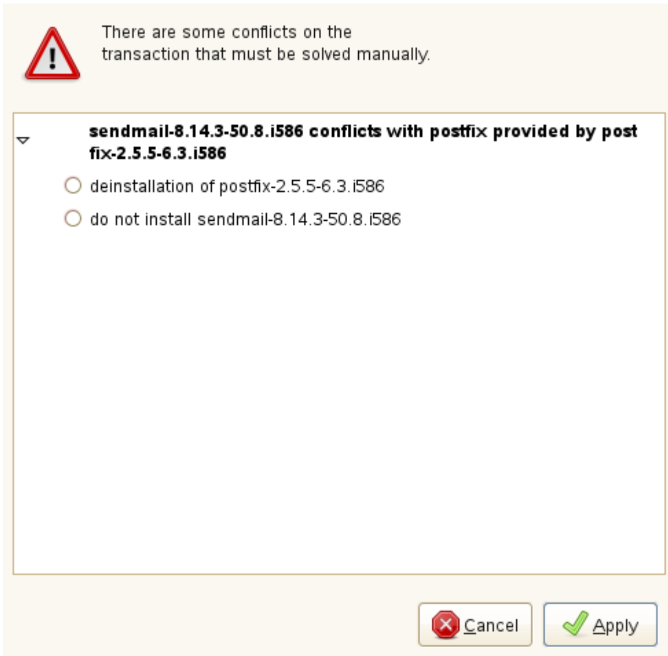
When the package manager starts, it examines the system and displays the installed packages. When you select a packag to install and remove it, the package manager automatically checks the dependencies and selects any other required packages (resolution of dependencies). If you select or deselect conflicting packages, the package manager indicates this and suggests possible solutions to this problem (resolution of conflicts).

For example, `sendmail` and `postfix` should not be installed concurrently. Figure 3.2, "Conflict Management of the Package Manager" (page 72) shows a conflict message prompting you to make a decision. `postfix` is already installed. Accordingly, you can decide wheter to install `sendmail` or remove `postfix`.

WARNING: Handling Package Conflicts

Unless you are very experienced, follow the suggestions of YaST when handling package conflicts, because otherwise the stability and functionality of your system could be endangered by the existing conflict.

Figure 3.2 *Conflict Management of the Package Manager*



3.3.3 Packages and Software Repositories

If you want to search for packages originating from one particular software repository, use grouping by *Repositories*. This view shows a list of all packages from the selected installation source.

To view a list of the all installed packages from the selected repository, click *Installed*. From this listing select packages for removing. To accomplish the opposite action, click *Available* and select packages for installation.

3.3.4 Removing Software

If you want to remove software from the system, proceed as follows:

1. Make use of a search strategy explained in Section 3.3.1, “Installing Software” (page 69).

2. In the *Packages Listing* mark the packages you want to remove. To mark all of them with one click, right-click in the *Packages Listing* pane and and choose *Select All*.

3. Click *Remove*.

   If you try to remove a package which is required by installed software, the conflict manager will complain about dependency issues and you must resolve the conflicts first as descripted in Section 3.3.2, “Checking Software Dependencies” (page 71).

   When all conflicts are resolved, the package scheduled for removing is listed in the *Changes* pane on the right side.

4. Click *Apply* to perform all actions listed in the *Changes* pane.

3.4 Managing Software Repositories and Services

Add additional software repositories to your system to install third-party software. By default, the product repository such as openSUSE-DVD 11.1 and a matching update repository are configured once you registered your system; for more information about registration, see Section “Registration” (Chapter 1, *Installation with YaST*, ↑Start-Up). Depending on the initially selected product, a separate language add-on repository with translations, dictionaries, etc. might also be configured.

Here also manage subscriptions to so-called *Services*. A Service in this context is a *Repository Index Service* (RIS) that can offer one or more software repositories. Such a Service can be changed dynamically by its administrator or vendor.

WARNING: Trusting External Software Sources

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

To ensure the integrity software repositories can be signed with the GPG Key of the repository maintainer. You can manage these keys in YaST—for more information, see GPG Keys (page 75). Whenever you add a new repository, YaST offers to import its key. Verify it as any other GPG key and pay attention that it does not change. If you detect a key change, something could be wrong with the repository and you should better disable it as an installation source until you know the cause of the key change.

To add product repositories either click *Software Repositories* directly in the *Software* pane of the YaST control center, or from within the *Software Management*, proceed as follows:

1 In the *Software Management* start screen, select *Repositories* from the upper left drop-down menu and then click *Edit* to display an overview of configured software repositories.

2 Click *Add* to select the media type holding the repository, for example, *DVD* or *USB Mass Storage* with the language add-ons. Then click *Next* and provide additional information about the medium.

3 YaST asks to insert the medium.

4 Confirm with *Continue*. It takes some moments until YaST has downloaded and parsed the metadata of the repository. Once done you can install software from this repository as described in Section 3.2.1, "Installing Software" (page 64) resp. Section 3.3.1, "Installing Software" (page 69).

If you want to add a repository of the openSUSE® Build Service, such as a version of Mozilla (`http://download.opensuse.org/repositories/mozilla/`), use the *Community Repositories* configuration dialog of YaST:

1 In the *Software Management* start screen, click *Repositories* from the upper left drop-down menu and then *Edit* to display an overview of configured software repositories.

2 Click *Add* to select the media type holding the repository, for example, *Community Repositories* with the Mozilla projects. Then click *Next*.

3 From the list of repositories activate the wanted items such as *openSUSE BuildService - Mozilla*.

Confirm with *OK*.

4. The new software repository is now listed in the *Configured Software Repositories* overview. Click *OK* again to install additional packages from this repository as described in Section 3.2.1, “Installing Software” (page 64) resp. Section 3.3.1, “Installing Software” (page 69).

In the *Configured Software Repositories* overview find several configuration options:

Properties
: By default, after adding a new repository, the repository is *Enabled* and the *Automatically Refresh* is active. This means, YaST will pull in updated meta data automatically and is always aware of new versions.

  The *Priority* of a repository is a value between `0` and `99`, where `0` is the highest priority. If a package is available in more than one repository the repository with the highest priority wins. This is useful if you want to give a local repository (for example, a DVD) a higher priority to avoid downloading packages unnecessarily from the Internet although they have the same or a higher version number.

GPG Keys
: Clicking *GPG Keys*, you open the GPG public keys management interface. In the *GPG Keys* subdialog, you can add new keys manually, delete or edit existing keys.

Refresh
: *Refresh* lets you update the repository meta data in various ways.

4

1-Click Install

It is also possible to install software packages from within a Web browser without subscribing to a repository first. For example, you can search in the openSUSE Build Service for software you want to install. This procedure is part of the *Package Search* and is called “1-Click Install”.

1. Start the openSUSE Build Service at `http://software.opensuse.org/search`.

2. Search for a package you want to install, for example, the OpenStreetMap editor `josm`, and select your system version, such as `openSUSE 11.1` or `SLE_11`, from the drop-down menu.

3. Click *Search*.

4. From the result list select the preferred item by clicking the *1-Click Install* button.

5. In the Web browser's download dialog, select the YaST Meta Package Handler.

6. In the *Additional Software Repositories* dialog choose the repository settings. If you are interested in updates, keep *Remain Subscribed to These Repositories after Installation* activated. Otherwise, uncheck this option.

   If you remain subscribed to these repositories, other package management tools, such as YaST or zypper can install or update software from it. Click *Next* when finished.

7. Confirm the next dialogs,*Software to be Installed* and *Proposal*, with *Next*. Study any warning carefully.

8. Enter the `root` password, if you actually want to install the selected software components. Several progress pop-ups appear. After reading the “Installation was successful” message, click *Finish*.

TIP: Disabling 1-Click Install Feature

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

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

---

5 YaST Online Update

openSUSE offers a continuous stream of software security updates for your product. By default openSUSE Updater is used to keep your system up-to-date. Refer to Section “Keeping the System Up-to-date” (Chapter 3, *Installing, Removing and Updating Software*, ↑Start-Up) for further information on openSUSE Updater. This chapter covers the alternative tool for updating software packages: YaST Online Update.

The current patches for openSUSE® are available from an update software repository. If you have registered your product during the installation, an update repository is already configured. If you have not registered openSUSE, 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 3.4, “Managing Software Repositories and Services” (page 73).

openSUSE provides updates with different relevance levels. `Security` updates fix severe security hazards and should definitely be installed. `Recommended` updates fix issues that could compromise your computer, whereas `Optional` updates fix non-security relevant issues or provide enhancements.

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

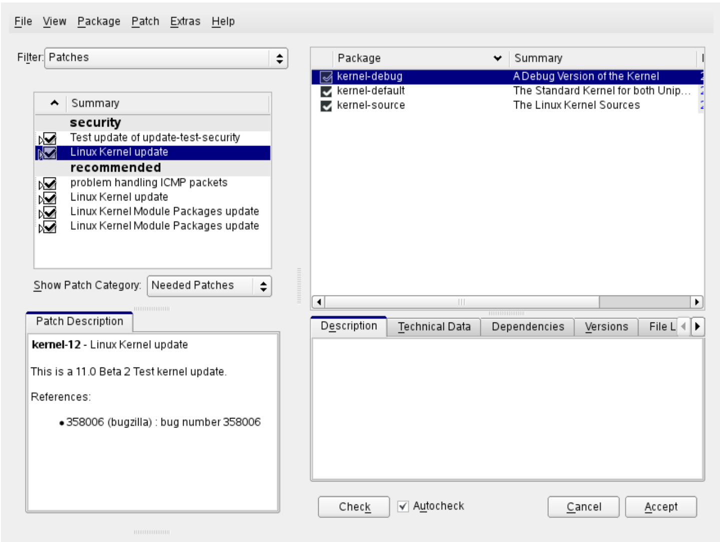
TIP: Disabling deltarpms

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

5.1 Installing Patches Manually Using the Qt Interface

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

Figure 5.1 *YaST Online Update*



The patch display lists the available patches for openSUSE. 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 have been fulfilled (because they have already been updated from another source).

All Patches
: All patches available for openSUSE.

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.

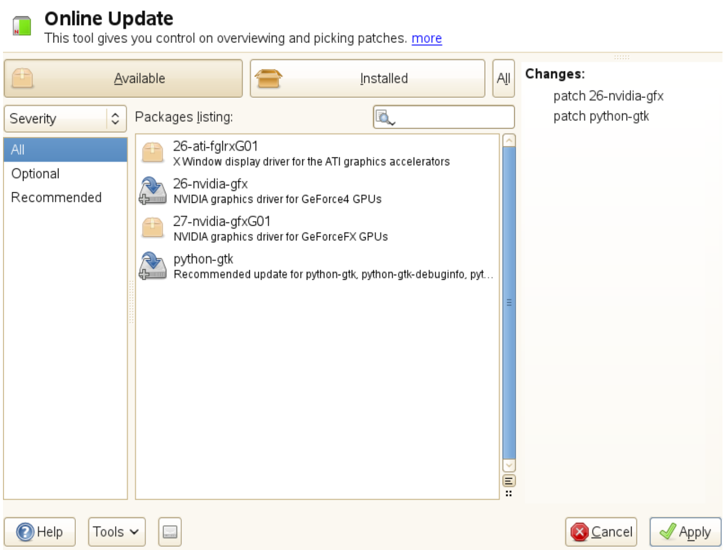
If you install an up-to-date package from a repository other than the update repository, the requirements of a patch for this package may be fulfilled with this installation. In this case a check mark is displayed in front of the patch summary. The patch will be visible in the list until you mark it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

5.2 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 carried out once you *Apply* them.

Figure 5.2 *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 area of the left pane. Here you can see a detailed patch description as well as the versions available. You can also choose to *Install* optional patches—security and recommended patches are already preselected for installation.

5.3 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.

6 Installing Add-On Products

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

6.1 Add-Ons

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

1. Click *Software* > *Add-On Products* to see an overview of installed add-on products.

2. Select various types of product media, such as CD, FTP or a local directory, by clicking *Add*. You can also use ISO images instead of CD or DVD media.

3. To add an ISO image, select *Local ISO Image* and click*Next*.

4. Enter the *Path to ISO Image* and choose a *Repository Name*. Click *Next*.

5. After successfully adding the add-on media, the software manager window appears. If the add-on provides a new pattern, see the new item in the *Patterns* filter. To view the list of all packages from the selected software repository, select the

filter *Software Repositories* and choose the repository to view. To view packages from a selected add-on by package groups, select the secondary filter *Package Groups* in the YaST Qt interface.

6.2 Binary Drivers

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

Managing Software with Command Line Tools 7

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

7.1 Using Zypper

Zypper is a command line tool for installing and updating packages. It is especially useful for accomplishing remote software management tasks or managing software from shell scripts.

zypper has a help overview built in:

```
zypper help
```

7.1.1 General Usage

The general syntax of zypper is:

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

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

```
zypper patch
```

Additionally, you can choose from one or more global options by typing them just before the command. For example, `--non-interactive` means, run the command without asking anything, decide on your own:

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

To use the options specific to a particular command, type them right after the command. For example, `--auto-agree-with-licenses` means, apply all needed patches to the system without asking to confirm any licenses—all of them were read in advance:

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

Some of the commands require one or more arguments:

```
zypper install mplayer
```

Some of the options also require an argument. The following 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
```

7.1.2 Installing and Removing Software with Zypper

To install a package from registered repositories, use:

```
zypper install package_name
```

To install a specific version of a package, use:

```
zypper install package_name=version
```

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

```
zypper install package_name*
```

You can also install a local or remote RPM directly—Zypper will also install all packages that *`package_name`* is dependent on automatically with the following:

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

To remove an installed package, use:

```
zypper remove package_name
```

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

```
zypper install emacs -vim
```

Or:

```
zypper remove emacs +vim
```

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

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

WARNING: Do not Remove 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 the system may stop working.

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.

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

```
zypper source-install package_name
```

The following command will also install the build dependencies of the specified package. If you do not want this, add the switch `--no-build-deps` as follows:

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

Of course, this will only work if you have the repository with the source packages added to your repository list. Enter `zypper search -t srcpackage` to get a list of source packages available in your repositories. For more information about adding repositories, see Section 7.1.4, "Managing Repositories" (page 90).

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

```
zypper verify
```

7.1.3 Updating Software with Zypper

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

```
zypper patch
```

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

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

```
zypper update
```

To update individual packages, use the update command with arguments:

```
zypper update package_name
```

Or the installation command:

```
zypper install package_name
```

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

```
zypper list-updates
```

Similarily, to list all needed patches, use:

```
zypper list-patches
```

7.1.4 Managing Repositories

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.

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

```
zypper removerepo 3
```

To add a repository, run

```
zypper addrepo URI Alias
```

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

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

```
zypper renamerepo 1 main
```

7.1.5 Querying

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

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

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

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

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

7.1.6 For More Information

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

7.2 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`.

7.2.1 Verifying Package Authenticity

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

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

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

7.2.2 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`.

7.2.3 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`.

7.2.4 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.

7.2.5 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 7.1, "The Most Important RPM Query Options" (page 97).

***Table 7.1** The Most Important RPM Query Options*

`-i`	Package information
`-l`	File list
`-f FILE`	Query the package that contains the file *`FILE`* (the full path must be specified with *`FILE`*)
`-s`	File list with status information (implies `-l`)
`-d`	List only documentation files (implies `-l`)

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

For example, the command `rpm -q -i wget` displays the information shown in Example 7.1, "rpm -q -i wget" (page 98).

***Example 7.1** rpm -q -i wget*

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

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

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

results in:

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

If only part of the filename is known, use a shell script as shown in Example 7.2, "Script to Search for Packages" (page 99). Pass the partial filename to the script shown as a parameter when running it.

Example 7.2 *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. The above example shows information about the package `rpm`.

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

Table 7.2 *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`.

7.2.6 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 package with YaST, you should have files similar to the following listing:

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

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

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

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

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

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

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

`--short-circuit`
: Skip some steps.

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

7.2.7 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.

7.2.8 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 3, *Installing or Removing Software* (page 63)).

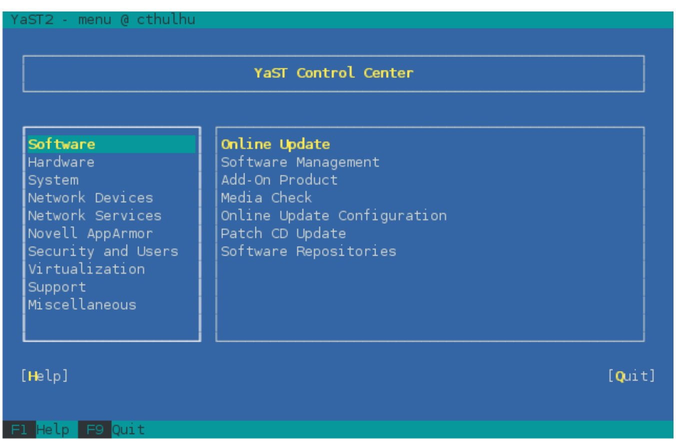
Part III. Administration

8 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 8.1 *Main Window of YaST in Text Mode*



When YaST is started in text mode, the YaST Control Center appears first (see Figure 8.1). The main window consists of three areas. The left frame, which is surrounded by a thick white border, features the categories to which the various modules belong. The active category is indicated by a colored background. The right frame, which is surrounded by a thin white border, provides an overview of the modules available in the active category. The bottom frame contains the buttons for *Help* and *Quit*.

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

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

8.1 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 8.2, “Restriction of Key Combinations” (page 110) 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.

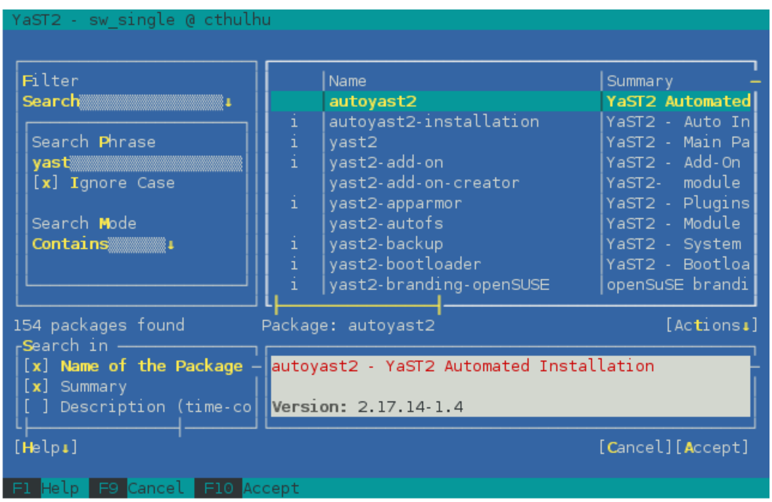
Buttons, Radio Buttons, and Check Boxes
: To select buttons with empty square brackets (check boxes) or empty parentheses (radio buttons), press Space or Enter. Alternatively, radio buttons and check boxes can be selected directly with Alt + yellow_letter. In this case, you do not need to

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

Function Keys
: The F keys (F1 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. 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 8.2 *The Software Installation Module*



8.2 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.

8.3 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
```

8.3.1 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`.

8.3.2 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 7.1, “Using Zypper” (page 87).

8.3.3 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.
```

9 Printer Operation

openSUSE® supports printing with many types of printers, including remote network printers. Printers can be configured with YaST or manually. For configuration instructions, refer to Section “Setting Up a Printer” (Chapter 2, *Setting Up Hardware Components with YaST*, ↑Start-Up). 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 9.8, “Troubleshooting” (page 123).

CUPS is the standard print system in openSUSE. 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 openSUSE only for reasons of compatibility.

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

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

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

Proprietary Printers (Also Called GDI Printers)
: These printers do not support any of the common printer languages. They use their own undocumented printer languages, which are subject to change when a new edition of a model is released. Usually only Windows drivers are available for these printers. See Section 9.8.1, "Printers without Standard Printer Language Support" (page 123) for more information.

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

`http://www.linuxfoundation.org/en/OpenPrinting/`
: The OpenPrinting.org printer database.

`http://www.cs.wisc.edu/~ghost/`
: The Ghostscript Web page.

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

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

9.1 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.

9.2 Methods and Protocols for Connecting Printers

There are various possibilities for connecting a printer to the system. The configuration of the CUPS print system does not distinguish between a local printer and a printer connected to the system over the network. In Linux, local printers must be connected as described in the manual of the printer manufacturer. CUPS supports serial, USB, parallel, and SCSI connections. For more information about the printer connection, read the article *CUPS in a Nutshell* in the Support Database at `http://en.opensuse.org/SDB:CUPS_in_a_Nutshell`.

WARNING: Changing Cable Connections in a Running System

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

9.3 Installing the Software

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

To configure a PostScript printer, the best approach is to get a suitable PPD file. Many PPD files are available in the package `manufacturer-PPDs`, which is automatically installed within the scope of the standard installation. See Section 9.7.2, “PPD Files in Various Packages” (page 121) and Section 9.8.2, “No Suitable PPD File Available for a PostScript Printer” (page 124).

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 2, *Setting Up Hardware Components with YaST*, ↑Start-Up)). 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 openSUSE and, second, print commands may work differently and the system may no longer be able to address devices of other manufacturers. For this reason, the installation of manufacturer software is not recommended.

9.4 Network Printers

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

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

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

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

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

SMB (Windows Share)
: CUPS also supports printing on printers connected to Windows shares. The protocol used for this purpose is SMB. SMB uses the port numbers `137`, `138`, and `139`. Example device URIs are

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

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

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

9.4.1 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-level1.ppd.gz -E
```

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

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

1. First, list all options:

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

   Example:

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

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

2. Change the option with `lpadmin`:

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

3. Check the new setting:

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

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

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

9.5 Graphical Printing Interfaces

Tools such as xpp and the KDE program KPrinter provide a graphical interface for choosing queues and setting both CUPS standard options and printer-specific options made available through the PPD file. You can even use KPrinter as the standard printing interface of non-KDE applications. In the print dialog of these applications, specify either `kprinter` or `kprinter --stdin` as the print command. The command to use depends on how the application transmits the data—just try which one works. If set up correctly, the application should open the KPrinter dialog whenever a print job is issued from it, so you can use the dialog to select a queue and set other printing options. This requires that the application's own print setup does not conflict with that of KPrinter and that printing options are only changed through KPrinter after it has been enabled. More information on KPrinter is available in Chapter 7, *Managing Print Jobs* (↑KDE User Guide).

9.6 Printing from the Command Line

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

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

9.7 Special Features in openSUSE

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

9.7.1 CUPS and Firewall

After having performed a default installation of openSUSE, SuSEfirewall2 is active and the network interfaces are configured to be in the `External Zone` which blocks incoming traffic. These default settings have to be adjusted when using CUPS. More information about the SUSEfirewall2 configuration is available in Section “SuSEfirewall2” (Chapter 14, *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.

9.7.2 PPD Files in Various Packages

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

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

CUPS PPD Files in the `cups` Package

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

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

PPD Files in the `cups-drivers` Package

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

YaST generally prefers a `manufacturer-PPD` file. However, when no suitable `manufacturer-PPD` file exists, a 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.

9.8 Troubleshooting

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

9.8.1 Printers without Standard Printer Language Support

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

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

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

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

9.8.2 No Suitable PPD File Available for a PostScript Printer

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

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

9.8.3 Parallel Ports

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

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

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

If interrupt `7` is free, it can be activated with the entry shown in Example 9.1, "/etc/modprobe.conf: Interrupt Mode for the First Parallel Port" (page 125). 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 9.1 */etc/modprobe.conf: Interrupt Mode for the First Parallel Port*

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

9.8.4 Network Printer Connections

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

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

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

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

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

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

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

If `lpd` does not respond, it may not be active or there may be basic network problems. If `lpd` responds, the response should show why printing is not possible on the `queue` on `host`. If you receive a response like that in Example 9.2, "Error Message from lpd" (page 126), the problem is caused by the remote `lpd`.

Example 9.2 *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 9.3, “Broadcast from the CUPS Network Server” (page 126).

Example 9.3 *Broadcast from the CUPS Network Server*

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

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

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

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

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

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

Troubleshooting a Network Printer or Print Server Box

Spoolers running in a print server box sometimes cause problems when they have to deal with a lot of print jobs. Because this is caused by the spooler in the print server box, there is nothing you can do about it. As a work-around, circumvent the spooler in the print server box by addressing the printer connected to the print server box directly with TCP socket. See Section 9.4, “Network Printers” (page 116).

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

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

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

Enter a command like

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

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

9.8.5 Defective Printouts without Error Message

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

9.8.6 Disabled Queues

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

9.8.7 CUPS Browsing: Deleting Print Jobs

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

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

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

9.8.8 Defective Print Jobs and Data Transfer Errors

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

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

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

2. The print job may still be in the queue, because jobs are only removed after they are sent completely to the printer. Use `lpstat -o` or `lpstat -h cups.example.com -o` to check which queue is currently printing. Delete the print job with `cancel` *`queue-jobnumber`* or `cancel -h cups.example.com` *`queue-jobnumber`*.

3. Some data may still be transferred to the printer even though the print job has been deleted from the queue. Check if a CUPS back-end process is still running for the respective queue and terminate it. For example, for a printer connected to the parallel port, the command `fuser -k /dev/lp0` can be used to terminate all processes that are still accessing the printer (more precisely: the parallel port).

4. Reset the printer completely by switching it off for some time. Then insert the paper and turn on the printer.

9.8.9 Debugging the CUPS Print System

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

1. Set `LogLevel debug` in `/etc/cups/cupsd.conf`.

2. Stop `cupsd`.

3. Remove `/var/log/cups/error_log*` to avoid having to search through very large log files.

4. Start `cupsd`.

5. Repeat the action that led to the problem.

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

9.8.10 For More Information

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

The X Window System 10

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 openSUSE®.

10.1 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 2, *Setting Up Hardware Components with YaST*, ↑Start-Up). Alternatively it can be configured manually by editing the its configuration files.

WARNING: Faulty X Configurations can Damage Your Hardware

Be very careful when configuring your X Window System. Never start the X Window System until the configuration is finished. A misconfigured system can cause irreparable damage to your hardware (this applies especially to fixed-frequency monitors). The creators of this book and openSUSE 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 openSUSE's SaX2 have failed. If your setup involves proprietary binary-only drivers, `X -configure` cannot work.

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

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

The section types available are listed in Table 10.1, "Sections in /etc/X11/xorg.conf" (page 132).

***Table 10.1** Sections in /etc/X11/xorg.conf*

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

Type	Meaning
`Monitor`	The monitor used. Important elements of this section are the `Identifier`, which is referred to later in the `Screen` definition, the refresh rate `VertRefresh`, and the synchronization frequency limits (`HorizSync` and `VertRefresh`). 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 SaX2 on the basis of the values given by the user and normally do not need to be changed. Intervene manually at this point if, for example, you want to connect a fixed frequency monitor. Find details of the meaning of individual number values in the HOWTO files in `/usr/share/doc/howto/en/html/XFree86-Video-Timings-HOWTO` (available in the `howtoenh` package). To calculate VESA modes manually, you can use the tool `cvt`. For example, to calculate a modeline for a 1680x1050@60Hz monitor, use the command `cvt 1680 1050 60`.
`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 `i810` driver, find more information about the available options in the manual page `man 4 i810`.
`Screen`	Combines a `Monitor` and a `Device` to form all the necessary settings for X.Org. In the `Display` subsection, specify the size of the virtual screen (`Virtual`), the `ViewPort`, and the `Modes` used with this screen. Note that some drivers demand that all of the used configurations must be present in the `Display` section at some place. For ex-

Type	Meaning
	ample, if you use a laptop and want to use an external monitor that is bigger than the internal LCD, it might be necessary to add a bigger resolution than supported by the internal LCD at the end of the `Modes` line.
`ServerLayout`	The layout of a single or multihead configuration. This section binds the input devices `InputDevice` and the display devices `Screen`.
`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.

10.1.1 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 10.1, "Screen Section of the File /etc/X11/xorg.conf" (page 135).

Example 10.1 *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.

❻ 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.

❼ 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.

10.1.2 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.

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

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

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

10.1.3 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.

10.2 Installing and Configuring Fonts

The installation of additional fonts in openSUSE is very easy. Simply copy the fonts to any directory located in the X11 font path (see Section 10.2.1, "X11 Core Fonts" (page 139)). To the enable use of the fonts, the installation directory should be a subdirectory of the directories configured in `/etc/fonts/fonts.conf` (see Section 10.2.2, "Xft" (page 140)) or included into this file with `/etc/fonts/suse-font-dirs.conf`.

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

```
<!-- Font directory list -->
<dir>/usr/share/fonts</dir>
<dir>/usr/X11R6/lib/X11/fonts</dir>
<dir>/opt/kde3/share/fonts</dir>
<dir>/usr/local/share/fonts</dir>
<dir>~/.fonts</dir>
<include ignore_missing="yes">conf.d</include>
```

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

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

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

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

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

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

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

10.2.1 X11 Core Fonts

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

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

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

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

By default, openSUSE 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 openSUSE contain at least the glyphs needed for European languages (formerly encoded as iso-8859-*).

10.2.2 Xft

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

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

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

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

and end with

```
</fontconfig>
```

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

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

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

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

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

to disable antialiasing for all fonts or

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

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

to disable antialiasing for specific fonts.

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

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

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

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

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

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

The output of this command could look like the following:

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

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

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

Table 10.2 *Parameters of fc-list*

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

10.3 For More Information

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

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

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

11 System Monitoring Utilities

A number of programs and mechanisms, some of which are presented here, can be used to examine the status of your system. Also described are some utilities that are useful for routine work, along with their most important parameters.

For each of the commands introduced, examples of the relevant outputs are presented. In these examples, the first line is the command itself (after the > or # sign prompt). Omissions are indicated with square brackets (`[...]`) and long lines are wrapped where necessary. Line breaks for long lines are indicated by a backslash (\).

```
# command -x -y
output line 1
output line 2
output line 3 is annoyingly long, so long that \
    we have to break it
output line 3
[...]
output line 98
output line 99
```

The descriptions have been kept short to allow as many utilities as possible to be mentioned. Further information for all the commands can be found in the man pages. Most of the commands also understand the parameter `--help`, which produces a brief list of the possible parameters.

11.1 Debugging

11.1.1 Specifying the Required Library: ldd

Use the command `ldd` to find out which libraries would load the dynamic executable specified as argument.

```
tux@mercury:~> ldd /bin/ls
        linux-vdso.so.1 =>  (0x00007fffbe7fe000)
        librt.so.1 => /lib64/librt.so.1 (0x00007f55b639d000)
        libacl.so.1 => /lib64/libacl.so.1 (0x00007f55b6195000)
        libc.so.6 => /lib64/libc.so.6 (0x00007f55b5e3d000)
        libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f55b5c21000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f55b65a6000)
        libattr.so.1 => /lib64/libattr.so.1 (0x00007f55b5a1c000)
```

Static binaries do not need any dynamic libraries.

```
tux@mercury:~> ldd /bin/sash
        not a dynamic executable
tux@mercury:~> file /bin/sash
/bin/sash: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), for GNU/Linux
 2.6.4, statically linked, stripped
```

11.1.2 Library Calls of a Program Run: ltrace

The command `ltrace` enables you to trace the library calls of a process. This command is used in a similar fashion to `strace`. The parameter `-c` outputs the number and duration of the library calls that have occurred:

```
tux@mercury:~> ltrace -c find ~
% time     seconds  usecs/call     calls      function
------ ----------- ----------- --------- --------------------
 34.37    6.758937         245     27554 __errno_location
 33.53    6.593562         788      8358 __fprintf_chk
 12.67    2.490392         144     17212 strlen
 11.97    2.353302         239      9845 readdir64
  2.37    0.466754          27     16716 __ctype_get_mb_cur_max
  1.17    0.230765          27      8358 memcpy
[...]
  0.00    0.000036          36         1 textdomain
------ ----------- ----------- --------- --------------------
100.00   19.662715                105717 total
```

11.1.3 System Calls of a Program Run: strace

The utility `strace` enables you to trace all the system calls of a process currently running. Enter the command in the normal way, adding `strace` at the beginning of the line:

```
tux@mercury:~> strace ls
execve("/bin/ls", ["ls"], [/* 61 vars */]) = 0
uname({sys="Linux", node="mercury", ...}) = 0
brk(0)                                  = 0x805c000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or \
    directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=89696, ...}) = 0
mmap2(NULL, 89696, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7ef2000
close(3)                                = 0
open("/lib/librt.so.1", O_RDONLY)       = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0000\36\0"..., 512) \
   = 512
fstat64(3, {st_mode=S_IFREG|0755, st_size=36659, ...}) = 0
[...]
stat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 0), ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) \
    = 0xb7ca7000
write(1, "bin  Desktop  Documents  music\tM"..., 55bin  Desktop  Documents \
   \  music       Music  public_html  tmp
) = 55
close(1)                                = 0
munmap(0xb7ca7000, 4096)                = 0
exit_group(0)                           = ?
```

For example, to trace all attempts to open a particular file, use the following:

```
tux@mercury:~> strace -e open ls .bashrc
open("/etc/ld.so.cache", O_RDONLY)      = 3
open("/lib/librt.so.1", O_RDONLY)       = 3
open("/lib/libacl.so.1", O_RDONLY)      = 3
open("/lib/libc.so.6", O_RDONLY)        = 3
open("/lib/libpthread.so.0", O_RDONLY)  = 3
open("/lib/libattr.so.1", O_RDONLY)     = 3
[...]
```

To trace all the child processes, use the parameter `-f`. The behavior and output format of strace can be largely controlled. For information, see `man strace`.

11.2 Files and File Systems

11.2.1 Determine the File Type: file

The command `file` determines the type of a file or a list of files by checking `/etc/magic`.

```
tux@mercury:~> file /usr/bin/file
/usr/bin/file: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), \
    for GNU/Linux 2.6.4, dynamically linked (uses shared libs), stripped
```

The parameter `-f` *`list`* specifies a file with a list of filenames to examine. The `-z` allows `file` to look inside compressed files:

```
tux@mercury:~> file /usr/share/man/man1/file.1.gz
usr/share/man/man1/file.1.gz: gzip compressed data, from Unix, max compression
tux@mercury:~> file -z /usr/share/man/man1/file.1.gz
/usr/share/man/man1/file.1.gz: ASCII troff or preprocessor input text \
    (gzip compressed data, from Unix, max compression)
```

11.2.2 File Systems and Their Usage: mount, df, and du

The command `mount` shows which file system (device and type) is mounted at which mount point:

```
tux@mercury:~> mount
/dev/sda3 on / type reiserfs (rw,acl,user_xattr)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
udev on /dev type tmpfs (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/sda1 on /boot type ext2 (rw,acl,user_xattr)
/dev/sda4 on /local type reiserfs (rw,acl,user_xattr)
/dev/fd0 on /media/floppy type subfs (rw,nosuid,nodev,noatime,fs=floppyfss,p
```

Obtain information about total usage of the file systems with the command `df`. The parameter `-h` (or `--human-readable`) transforms the output into a form understandable for common users.

```
tux@mercury:~> df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda3              11G  3.2G  6.9G  32% /
udev                  252M  104K  252M   1% /dev
/dev/sda1              16M  6.6M  7.8M  46% /boot
/dev/sda4              27G   34M   27G   1% /local
```

Display the total size of all the files in a given directory and its subdirectories with the command `du`. The parameter `-s` suppresses the output of detailed information. `-h` again transforms the data into a human-readable form:

```
tux@mercury:~> du -sh /local
1.7M    /local
```

11.2.3 Additional Information about ELF Binaries

Read the content of binaries with the `readelf` utility. This even works with ELF files that were built for other hardware architectures:

```
tux@mercury:~> readelf --file-header /bin/ls
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF64
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Advanced Micro Devices X86-64
  Version:                           0x1
  Entry point address:               0x402430
  Start of program headers:          64 (bytes into file)
  Start of section headers:          98616 (bytes into file)
  Flags:                             0x0
  Size of this header:               64 (bytes)
  Size of program headers:           56 (bytes)
  Number of program headers:         9
  Size of section headers:           64 (bytes)
  Number of section headers:         31
  Section header string table index: 30
```

11.2.4 File Properties: stat

The command `stat` displays file properties:

```
tux@mercury:~> stat /etc/profile
  File: '/etc/profile'
  Size: 8080            Blocks: 16         IO Block: 4096   regular file
Device: 806h/2054d      Inode: 64942       Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2007-07-16 23:28:18.000000000 +0200
Modify: 2006-09-19 14:45:01.000000000 +0200
Change: 2006-12-05 14:54:55.000000000 +0100
```

The parameter `--filesystem` produces details of the properties of the file system in which the specified file is located:

```
tux@mercury:~> stat /etc/profile --filesystem
  File: "/etc/profile"
    ID: 0        Namelen: 255     Type: reiserfs
Block size: 4096       Fundamental block size: 4096
Blocks: Total: 2622526    Free: 1809771    Available: 1809771
Inodes: Total: 0          Free: 0
```

11.3 Hardware Information

11.3.1 PCI Resources: lspci

The command `lspci` lists the PCI resources:

```
mercury:~ # lspci
00:00.0 Host bridge: Intel Corporation 82845G/GL[Brookdale-G]/GE/PE \
    DRAM Controller/Host-Hub Interface (rev 01)
00:01.0 PCI bridge: Intel Corporation 82845G/GL[Brookdale-G]/GE/PE \
    Host-to-AGP Bridge (rev 01)
00:1d.0 USB Controller: Intel Corporation 82801DB/DBL/DBM \
    (ICH4/ICH4-L/ICH4-M) USB UHCI Controller #1 (rev 01)
00:1d.1 USB Controller: Intel Corporation 82801DB/DBL/DBM \
    (ICH4/ICH4-L/ICH4-M) USB UHCI Controller #2 (rev 01)
00:1d.2 USB Controller: Intel Corporation 82801DB/DBL/DBM \
    (ICH4/ICH4-L/ICH4-M) USB UHCI Controller #3 (rev 01)
00:1d.7 USB Controller: Intel Corporation 82801DB/DBM \
    (ICH4/ICH4-M) USB2 EHCI Controller (rev 01)
00:1e.0 PCI bridge: Intel Corporation 82801 PCI Bridge (rev 81)
00:1f.0 ISA bridge: Intel Corporation 82801DB/DBL (ICH4/ICH4-L) \
    LPC Interface Bridge (rev 01)
00:1f.1 IDE interface: Intel Corporation 82801DB (ICH4) IDE \
```

```
    Controller (rev 01)
00:1f.3 SMBus: Intel Corporation 82801DB/DBL/DBM (ICH4/ICH4-L/ICH4-M) \
    SMBus Controller (rev 01)
00:1f.5 Multimedia audio controller: Intel Corporation 82801DB/DBL/DBM \
    (ICH4/ICH4-L/ICH4-M) AC'97 Audio Controller (rev 01)
01:00.0 VGA compatible controller: Matrox Graphics, Inc. G400/G450 (rev 85)
02:08.0 Ethernet controller: Intel Corporation 82801DB PRO/100 VE (LOM) \
    Ethernet Controller (rev 81)
```

Using `-v` results in a more detailed listing:

```
mercury:~ # lspci -v
[...]
02:08.0 Ethernet controller: Intel Corporation 82801DB PRO/100 VE (LOM)\
    Ethernet Controller (rev 81)
        Subsystem: Fujitsu Siemens Computer GmbH: Unknown device 1001
        Flags: bus master, medium devsel, latency 66, IRQ 11
        Memory at d1000000 (32-bit, non-prefetchable) [size=4K]
        I/O ports at 3000 [size=64]
        Capabilities: [dc] Power Management version 2
```

Information about device name resolution is obtained from the file `/usr/share/pci.ids`. PCI IDs not listed in this file are marked “Unknown device.”

The parameter `-vv` produces all the information that could be queried by the program. To view the pure numeric values, use the parameter `-n`.

11.3.2 USB Devices: lsusb

The command `lsusb` lists all USB devices. With the option `-v`, print a more detailed list. The detailed information is read from the directory `/proc/bus/usb/`. The following is the output of `lsusb` with these USB devices attached: hub, memory stick, hard disk, and mouse.

```
mercury:/ # lsusb
Bus 004 Device 007: ID 0ea0:2168 Ours Technology, Inc. Transcend JetFlash \
    2.0 / Astone USB Drive
Bus 004 Device 006: ID 04b4:6830 Cypress Semiconductor Corp. USB-2.0 IDE \
    Adapter
Bus 004 Device 005: ID 05e3:0605 Genesys Logic, Inc.
Bus 004 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 005: ID 046d:c012 Logitech, Inc. Optical Mouse
Bus 001 Device 001: ID 0000:0000
```

11.4 Networking

11.4.1 Show the Network Status: netstat

`netstat` shows network connections, routing tables (`-r`), interfaces (`-i`), masquerade connections (`-M`), multicast memberships (`-g`), and statistics (`-s`).

```
tux@mercury:~> netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags  MSS Window  irtt Iface
192.168.2.0     *               255.255.254.0   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         192.168.2.254   0.0.0.0         UG       0 0          0 eth0
```

```
tux@mercury:~> netstat -i
Kernel Interface table
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR  TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0   1500   0 1624507 129056      0      0   7055      0      0      0 BMNRU
lo    16436   0   23728      0      0      0  23728      0      0      0 LRU
```

When displaying network connections or statistics, you can specify the socket type to display: TCP (`-t`), UDP (`-u`), or raw (`-r`). The `-p` option shows the PID and name of the program to which each socket belongs.

The following example lists all TCP connections and the programs using these connections.

```
mercury:~ # netstat -t -p
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address  Foreign Address       State       PID/Pro

tcp      0      0 mercury:33513    www.novell.com:www-http ESTABLISHED 6862/fi
tcp      0    352 mercury:ssh      mercury2.:trc-netpoll     ESTABLISHED
19422/s
tcp      0      0 localhost:ssh  localhost:17828         ESTABLISHED -
```

In the following, statistics for the TCP protocol are displayed:

```
tux@mercury:~> netstat -s -t
Tcp:
    2427 active connections openings
    2374 passive connection openings
    0 failed connection attempts
    0 connection resets received
    1 connections established
    27476 segments received
```

```
    26786 segments send out
    54 segments retransmited
    0 bad segments received.
    6 resets sent
[...]
    TCPAbortOnLinger: 0
    TCPAbortFailed: 0
    TCPMemoryPressures: 0
```

11.5 The /proc File System

The `/proc` file system is a pseudo file system in which the kernel reserves important information in the form of virtual files. For example, display the CPU type with this command:

```
tux@mercury:~> cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 15
model           : 4
model name      : Intel(R) Pentium(R) 4 CPU 3.40GHz
stepping        : 3
cpu MHz         : 2800.000
cache size      : 2048 KB
physical id     : 0
[...]
```

Query the allocation and use of interrupts with the following command:

```
tux@mercury:~> cat /proc/interrupts
           CPU0
  0:    3577519          XT-PIC  timer
  1:        130          XT-PIC  i8042
  2:          0          XT-PIC  cascade
  5:     564535          XT-PIC  Intel 82801DB-ICH4
  7:          1          XT-PIC  parport0
  8:          2          XT-PIC  rtc
  9:          1          XT-PIC  acpi, uhci_hcd:usb1, ehci_hcd:usb4
 10:          0          XT-PIC  uhci_hcd:usb3
 11:      71772          XT-PIC  uhci_hcd:usb2, eth0
 12:     101150          XT-PIC  i8042
 14:      33146          XT-PIC  ide0
 15:     149202          XT-PIC  ide1
NMI:          0
LOC:          0
ERR:          0
MIS:          0
```

Some of the important files and their contents are:

`/proc/devices`
: Available devices

`/proc/modules`
: Kernel modules loaded

`/proc/cmdline`
: Kernel command line

`/proc/meminfo`
: Detailed information about memory usage

`/proc/config.gz`
: `gzip`-compressed configuration file of the kernel currently running

Further information is available in the text file `/usr/src/linux/Documentation/filesystems/proc.txt` (this file is available when the package `kernel-source` is installed). Find information about processes currently running in the `/proc/`*NNN* directories, where *NNN* is the process ID (PID) of the relevant process. Every process can find its own characteristics in `/proc/self/`:

```
tux@mercury:~> ls -l /proc/self
lrwxrwxrwx 1 root root 64 2007-07-16 13:03 /proc/self -> 5356
tux@mercury:~> ls -l /proc/self/
total 0
dr-xr-xr-x 2 tux users 0 2007-07-16 17:04 attr
-r-------- 1 tux users 0 2007-07-16 17:04 auxv
-r--r--r-- 1 tux users 0 2007-07-16 17:04 cmdline
lrwxrwxrwx 1 tux users 0 2007-07-16 17:04 cwd -> /home/tux
-r-------- 1 tux users 0 2007-07-16 17:04 environ
lrwxrwxrwx 1 tux users 0 2007-07-16 17:04 exe -> /bin/ls
dr-x------ 2 tux users 0 2007-07-16 17:04 fd
-rw-r--r-- 1 tux users 0 2007-07-16 17:04 loginuid
-r--r--r-- 1 tux users 0 2007-07-16 17:04 maps
-rw------- 1 tux users 0 2007-07-16 17:04 mem
-r--r--r-- 1 tux users 0 2007-07-16 17:04 mounts
-rw-r--r-- 1 tux users 0 2007-07-16 17:04 oom_adj
-r--r--r-- 1 tux users 0 2007-07-16 17:04 oom_score
lrwxrwxrwx 1 tux users 0 2007-07-16 17:04 root -> /
-rw------- 1 tux users 0 2007-07-16 17:04 seccomp
-r--r--r-- 1 tux users 0 2007-07-16 17:04 smaps
-r--r--r-- 1 tux users 0 2007-07-16 17:04 stat
[...]
```

```
dr-xr-xr-x 3 tux users 0 2007-07-16 17:04 task
-r--r--r-- 1 tux users 0 2007-07-16 17:04 wchan
```

The address assignment of executables and libraries is contained in the `maps` file:

```
tux@mercury:~> cat /proc/self/maps
08048000-0804c000 r-xp 00000000 03:03 17753      /bin/cat
0804c000-0804d000 rw-p 00004000 03:03 17753      /bin/cat
0804d000-0806e000 rw-p 0804d000 00:00 0          [heap]
b7d27000-b7d5a000 r--p 00000000 03:03 11867      /usr/lib/locale/en_GB.utf8/
b7d5a000-b7e32000 r--p 00000000 03:03 11868      /usr/lib/locale/en_GB.utf8/
b7e32000-b7e33000 rw-p b7e32000 00:00 0
b7e33000-b7f45000 r-xp 00000000 03:03 8837       /lib/libc-2.3.6.so
b7f45000-b7f46000 r--p 00112000 03:03 8837       /lib/libc-2.3.6.so
b7f46000-b7f48000 rw-p 00113000 03:03 8837       /lib/libc-2.3.6.so
b7f48000-b7f4c000 rw-p b7f48000 00:00 0
b7f52000-b7f53000 r--p 00000000 03:03 11842      /usr/lib/locale/en_GB.utf8/
[...]
b7f5b000-b7f61000 r--s 00000000 03:03 9109       /usr/lib/gconv/gconv-module
b7f61000-b7f62000 r--p 00000000 03:03 9720       /usr/lib/locale/en_GB.utf8/
b7f62000-b7f76000 r-xp 00000000 03:03 8828       /lib/ld-2.3.6.so
b7f76000-b7f78000 rw-p 00013000 03:03 8828       /lib/ld-2.3.6.so
bfd61000-bfd76000 rw-p bfd61000 00:00 0          [stack]
ffffe000-fffff000 ---p 00000000 00:00 0          [vdso]
```

11.5.1 procinfo

Important information from the `/proc` file system is summarized by the command `procinfo`:

```
tux@mercury:~> procinfo
Linux 2.6.18.8-0.5-default (geeko@buildhost) (gcc 4.1.2 20061115) #1 2CPU

Memory:      Total        Used        Free      Shared     Buffers
Mem:       2060604     2011264       49340           0      200664
Swap:      2104472         112     2104360

Bootup: Tue Jul 10 10:29:15 2007    Load average: 0.86 1.10 1.11 3/118 21547

user  :       2:43:13.78   0.8%  page in :   71099181  disk 1:  2827023r 968
nice  :   1d 22:21:27.87  14.7%  page out:  690734737
system:      13:39:57.57   4.3%  page act:  138388345
IOwait:      18:02:18.59   5.7%  page dea:   29639529
hw irq:       0:03:39.44   0.0%  page flt: 9539791626
sw irq:       1:15:35.25   0.4%  swap in :         69
idle  :   9d 16:07:56.79  73.8%  swap out:        209
uptime:   6d 13:07:11.14         context :  542720687

irq  0: 141399308 timer          irq 14:   5074312 ide0
irq  1:     73784 i8042          irq 50:   1938076 uhci_hcd:usb1, ehci_
```

```
irq  4:          2                irq 58:         0 uhci_hcd:usb2
irq  6:          5 floppy [2]     irq 66:    872711 uhci_hcd:usb3, HDA I
irq  7:          2                irq 74:        15 uhci_hcd:usb4
irq  8:          0 rtc            irq 82: 178717720 0         PCI-MSI  e
irq  9:          0 acpi           irq169:  44352794 nvidia
irq 12:          3                irq233:   8209068 0         PCI-MSI  l
```

To see all the information, use the parameter `-a`. The parameter `-nN` produces updates of the information every *N* seconds. In this case, terminate the program by pressing Q.

By default, the cumulative values are displayed. The parameter `-d` produces the differential values. `procinfo -dn5` displays the values that have changed in the last five seconds:

11.6 Processes

11.6.1 Interprocess Communication: ipcs

The command `ipcs` produces a list of the IPC resources currently in use:

```
------ Shared Memory Segments --------
key        shmid      owner     perms      bytes      nattch     status
0x00000000 58261504   tux    600        393216     2          dest
0x00000000 58294273   tux    600        196608     2          dest
0x00000000 83886083   tux    666        43264      2
0x00000000 83951622   tux    666        192000     2
0x00000000 83984391   tux    666        282464     2
0x00000000 84738056   root      644        151552     2          dest

------ Semaphore Arrays --------
key        semid      owner     perms      nsems
0x4d038abf 0          tux    600        8

------ Message Queues --------
key        msqid      owner      perms      used-bytes   messages
```

11.6.2 Process List: ps

The command `ps` produces a list of processes. Most parameters must be written without a minus sign. Refer to `ps --help` for a brief help or to the man page for extensive help.

To list all processes with user and command line information, use `ps axu`:

```
tux@mercury:~> ps axu
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0    696   272 ?        S    12:59   0:01 init [5]
root         2  0.0  0.0      0     0 ?        SN   12:59   0:00 [ksoftirqd
root         3  0.0  0.0      0     0 ?        S<   12:59   0:00 [events
[...]
tux    4047  0.0  6.0 158548 31400 ?        Ssl  13:02   0:06 mono-best
tux    4057  0.0  0.7   9036  3684 ?        Sl   13:02   0:00 /opt/gnome
tux    4067  0.0  0.1   2204   636 ?        S    13:02   0:00 /opt/gnome
tux    4072  0.0  1.0  15996  5160 ?        Ss   13:02   0:00 gnome-scre
tux    4114  0.0  3.7 130988 19172 ?        SLl  13:06   0:04 sound-juic
tux    4818  0.0  0.3   4192  1812 pts/0    Ss   15:59   0:00 -bash
tux    4959  0.0  0.1   2324   816 pts/0    R+   16:17   0:00 ps axu
```

To check how many `sshd` processes are running, use the option `-p` together with the command `pidof`, which lists the process IDs of the given processes.

```
tux@mercury:~> ps -p $(pidof sshd)
  PID TTY      STAT   TIME COMMAND
 3524 ?        Ss     0:00 /usr/sbin/sshd -o PidFile=/var/run/sshd.init.pid
 4813 ?        Ss     0:00 sshd: tux [priv]
 4817 ?        R      0:00 sshd: tux@pts/0
```

The process list can be formatted according to your needs. The option `-L` returns a list of all keywords. Enter the following command to issue a list of all processes sorted by memory usage:

```
tux@mercury:~> ps ax --format pid,rss,cmd --sort rss
  PID   RSS CMD
    2     0 [ksoftirqd/0]
    3     0 [events/0]
    4     0 [khelper]
    5     0 [kthread]
   11     0 [kblockd/0]
   12     0 [kacpid]
  472     0 [pdflush]
  473     0 [pdflush]
[...]
 4028 17556 nautilus --no-default-window --sm-client-id default2
 4118 17800 ksnapshot
 4114 19172 sound-juicer
 4023 25144 gnome-panel --sm-client-id default1
 4047 31400 mono-best --debug /usr/lib/beagle/Best.exe --autostarted
 3973 31520 mono-beagled --debug /usr/lib/beagle/BeagleDaemon.exe --bg --aut
```

11.6.3 Process Tree: pstree

The command `pstree` produces a list of processes in the form of a tree:

```
tux@mercury:~> pstree
init-+-NetworkManagerD
     |-acpid
     |-3*[automount]
     |-cron
     |-cupsd
     |-2*[dbus-daemon]
     |-dbus-launch
     |-dcopserver
     |-dhcpcd
     |-events/0
     |-gpg-agent
     |-hald-+-hald-addon-acpi
     |      `-hald-addon-stor
     |-kded
     |-kdeinit-+-kdesu---su---kdesu_stub---yast2---y2controlcenter
     |         |-kio_file
     |         |-klauncher
     |         |-konqueror
     |         |-konsole-+-bash---su---bash
     |         |         `-bash
     |         `-kwin
     |-kdesktop---kdesktop_lock---xmatrix
     |-kdesud
     |-kdm-+-X
     |     `-kdm---startkde---kwrapper
[...]
```

The parameter `-p` adds the process ID to a given name. To have the command lines displayed as well, use the `-a` parameter:

11.6.4 Processes: top

The command `top`, which stands for "table of processes," displays a list of processes that is refreshed every two seconds. To terminate the program, press Q. The parameter `-n 1` terminates the program after a single display of the process list. The following is an example output of the command `top -n 1`:

```
tux@mercury:~> top -n 1
top - 17:06:28 up  2:10,  5 users,  load average: 0.00, 0.00, 0.00
Tasks:  85 total,   1 running,  83 sleeping,   1 stopped,   0 zombie
Cpu(s):  5.5% us,  0.8% sy,  0.8% ni, 91.9% id,  1.0% wa,  0.0% hi,  0.0% si
Mem:    515584k total,   506468k used,     9116k free,    66324k buffers
Swap:   658656k total,        0k used,   658656k free,   353328k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
    1 root      16   0   700  272  236 S  0.0  0.1   0:01.33 init
    2 root      34  19     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/0
    3 root      10  -5     0    0    0 S  0.0  0.0   0:00.27 events/0
    4 root      10  -5     0    0    0 S  0.0  0.0   0:00.01 khelper
    5 root      10  -5     0    0    0 S  0.0  0.0   0:00.00 kthread
   11 root      10  -5     0    0    0 S  0.0  0.0   0:00.05 kblockd/0
   12 root      20  -5     0    0    0 S  0.0  0.0   0:00.00 kacpid
  472 root      20   0     0    0    0 S  0.0  0.0   0:00.00 pdflush
  473 root      15   0     0    0    0 S  0.0  0.0   0:00.06 pdflush
  475 root      11  -5     0    0    0 S  0.0  0.0   0:00.00 aio/0
  474 root      15   0     0    0    0 S  0.0  0.0   0:00.07 kswapd0
  681 root      10  -5     0    0    0 S  0.0  0.0   0:00.01 kseriod
  839 root      10  -5     0    0    0 S  0.0  0.0   0:00.02 reiserfs/0
  923 root      13  -4  1712  552  344 S  0.0  0.1   0:00.67 udevd
 1343 root      10  -5     0    0    0 S  0.0  0.0   0:00.00 khubd
 1587 root      20   0     0    0    0 S  0.0  0.0   0:00.00 shpchpd_event
 1746 root      15   0     0    0    0 S  0.0  0.0   0:00.00 w1_control
 1752 root      15   0     0    0    0 S  0.0  0.0   0:00.00 w1_bus_master1
 2151 root      16   0  1464  496  416 S  0.0  0.1   0:00.00 acpid
 2165 messageb  16   0  3340 1048  792 S  0.0  0.2   0:00.64 dbus-daemon
 2166 root      15   0  1840  752  556 S  0.0  0.1   0:00.01 syslog-ng
 2171 root      16   0  1600  516  320 S  0.0  0.1   0:00.00 klogd
 2235 root      15   0  1736  800  652 S  0.0  0.2   0:00.10 resmgrd
 2289 root      16   0  4192 2852 1444 S  0.0  0.6   0:02.05 hald
 2403 root      23   0  1756  600  524 S  0.0  0.1   0:00.00 hald-addon-acpi
 2709 root      19   0  2668 1076  944 S  0.0  0.2   0:00.00 NetworkManagerD
 2714 root      16   0  1756  648  564 S  0.0  0.1   0:00.56 hald-addon-stor
```

If you press F while `top` is running, a menu opens with which to make extensive changes to the format of the output.

The parameter `-U` *`UID`* monitors only the processes associated with a particular user. Replace *`UID`* with the user ID of the user. `top -U $(id -u)` returns the UID of the user on the basis of the username and displays his processes.

11.6.5 Modify a process' niceness: nice and renice

The kernel determines which processes require more cpu time than others by the process' nice level, also called niceness. The higher the nice level of a process is, the less CPU time it will take from other processes. Nice levels range from -20 (the least "nice" level) to 19. Negative values can only set by `root`.

Adjusting the nice level is useful when running a non time-critical process that lasts long and uses large amounts of CPU time, such as compiling a kernel, on a system that also performs other tasks. Making such a process "nicer", ensures that the other tasks, for example a webserver, will have a higher priority.

Calling `nice` without any parameters prints the current niceness:

```
tux@mercury:~> nice
0
```

Running `nice` *`command`* increments the current nice level for the given command by 10. Using `nice -n` *`level command`* lets you specify a new niceness relative to the current one.

To change the niceness of a running process, use `renice` *`priority`* `-p` *`process id`*, for example:

```
renice +5 3266
```

To renice all processes owned by a specific user, use the option `-u` *`user`*. Process groups are reniced by the option `-g` *`process group id`*.

11.7 System Information

11.7.1 Memory Usage: free

The utility `free` examines RAM usage. Details of both free and used memory and swap areas are shown:

```
tux@mercury:~> free
            total       used       free     shared    buffers     cached
Mem:      2062844    2047444      15400          0     129580     921936
-/+ buffers/cache:    995928    1066916
Swap:     2104472          0    2104472
```

The options `-b`,`-k`,`-m`,`-g` show output in bytes, KB, MB, or GB, respectively. The parameter `-d delay` ensures that the display is refreshed every *`delay`* seconds. For example, `free -d 1.5` produces an update every 1.5 seconds.

11.7.2 User Accessing Files: fuser

It can be useful to determine what processes or users are currently accessing certain files. Suppose, for example, you want to unmount a file system mounted at `/mnt`. `umount` returns "device is busy." The command `fuser` can then be used to determine what processes are accessing the device:

```
tux@mercury:~> fuser -v /mnt/*

                     USER        PID ACCESS COMMAND
/mnt/notes.txt       tux    26597 f....  less
```

Following termination of the `less` process, which was running on another terminal, the file system can successfully be unmounted.

11.7.3 Kernel Ring Buffer: dmesg

The Linux kernel keeps certain messages in a ring buffer. To view these messages, enter the command `dmesg`:

```
$ dmesg
[...]
end_request: I/O error, dev fd0, sector 0
subfs: unsuccessful attempt to mount media (256)
e100: eth0: e100_watchdog: link up, 100Mbps, half-duplex
NET: Registered protocol family 17
IA-32 Microcode Update Driver: v1.14 <tigran@veritas.com>
microcode: CPU0 updated from revision 0xe to 0x2e, date = 08112004
IA-32 Microcode Update Driver v1.14 unregistered
bootsplash: status on console 0 changed to on
NET: Registered protocol family 10
Disabled Privacy Extensions on device c0326ea0(lo)
IPv6 over IPv4 tunneling driver
```

```
powernow: This module only works with AMD K7 CPUs
bootsplash: status on console 0 changed to on
```

Older events are logged in the files `/var/log/messages` and `/var/log/warn`.

11.7.4 List of Open Files: lsof

To view a list of all the files open for the process with process ID *`PID`*, use `-p`. For example, to view all the files used by the current shell, enter:

```
tux@mercury:~> lsof -p $$
COMMAND  PID   USER   FD   TYPE DEVICE    SIZE   NODE NAME
bash    5552 tux  cwd    DIR    3,3    1512 117619 /home/tux
bash    5552 tux  rtd    DIR    3,3     584      2 /
bash    5552 tux  txt    REG    3,3  498816  13047 /bin/bash
bash    5552 tux  mem    REG    0,0              0 [heap] (stat: No such
bash    5552 tux  mem    REG    3,3  217016 115687 /var/run/nscd/passwd
bash    5552 tux  mem    REG    3,3  208464  11867 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3  882134  11868 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3 1386997   8837 /lib/libc-2.3.6.so
bash    5552 tux  mem    REG    3,3   13836   8843 /lib/libdl-2.3.6.so
bash    5552 tux  mem    REG    3,3  290856  12204 /lib/libncurses.so.5.5
bash    5552 tux  mem    REG    3,3   26936  13004 /lib/libhistory.so.5.1
bash    5552 tux  mem    REG    3,3  190200  13006 /lib/libreadline.so.5.
bash    5552 tux  mem    REG    3,3      54  11842 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3    2375  11663 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3     290  11736 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3      52  11831 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3      34  11862 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3      62  11839 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3     127  11664 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3      56  11735 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3      23  11866 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3   21544   9109 /usr/lib/gconv/gconv-m
bash    5552 tux  mem    REG    3,3     366   9720 /usr/lib/locale/en_GB.
bash    5552 tux  mem    REG    3,3   97165   8828 /lib/ld-2.3.6.so
bash    5552 tux    0u   CHR  136,5              7 /dev/pts/5
bash    5552 tux    1u   CHR  136,5              7 /dev/pts/5
bash    5552 tux    2u   CHR  136,5              7 /dev/pts/5
bash    5552 tux  255u   CHR  136,5              7 /dev/pts/5
```

The special shell variable `$$`, whose value is the process ID of the shell, has been used.

The command `lsof` lists all the files currently open when used without any parameters. Because there are often thousands of open files, listing all of them is rarely useful. However, the list of all files can be combined with search functions to generate useful lists. For example, list all used character devices:

```
tux@mercury:~> lsof | grep CHR
bash      3838     tux    0u      CHR  136,0                 2 /dev/pts/0
bash      3838     tux    1u      CHR  136,0                 2 /dev/pts/0
bash      3838     tux    2u      CHR  136,0                 2 /dev/pts/0
bash      3838     tux  255u      CHR  136,0                 2 /dev/pts/0
bash      5552     tux    0u      CHR  136,5                 7 /dev/pts/5
bash      5552     tux    1u      CHR  136,5                 7 /dev/pts/5
bash      5552     tux    2u      CHR  136,5                 7 /dev/pts/5
bash      5552     tux  255u      CHR  136,5                 7 /dev/pts/5
X         5646       root  mem       CHR    1,1              1006 /dev/mem
lsof      5673     tux    0u      CHR  136,5                 7 /dev/pts/5
lsof      5673     tux    2u      CHR  136,5                 7 /dev/pts/5
grep      5674     tux    1u      CHR  136,5                 7 /dev/pts/5
grep      5674     tux    2u      CHR  136,5                 7 /dev/pts/5
```

11.7.5 Kernel and udev Event Sequence Viewer: udevadm monitor

`udevadm monitor` listens to the kernel uevents and events sent out by a udev rule and prints the device path (DEVPATH) of the event to the console. This is a sequence of events while connecting a USB memory stick:

```
UEVENT[1138806687] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2
UEVENT[1138806687] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2/4-2.2
UEVENT[1138806687] add@/class/scsi_host/host4
UEVENT[1138806687] add@/class/usb_device/usbdev4.10
UDEV  [1138806687] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2
UDEV  [1138806687] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2/4-2.2
UDEV  [1138806687] add@/class/scsi_host/host4
UDEV  [1138806687] add@/class/usb_device/usbdev4.10
UEVENT[1138806692] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2/4-2.2
UEVENT[1138806692] add@/block/sdb
UEVENT[1138806692] add@/class/scsi_generic/sg1
UEVENT[1138806692] add@/class/scsi_device/4:0:0:0
UDEV  [1138806693] add@/devices/pci0000:00/0000:00:1d.7/usb4/4-2/4-2.2/4-2.2
UDEV  [1138806693] add@/class/scsi_generic/sg1
UDEV  [1138806693] add@/class/scsi_device/4:0:0:0
UDEV  [1138806693] add@/block/sdb
UEVENT[1138806694] add@/block/sdb/sdb1
UDEV  [1138806694] add@/block/sdb/sdb1
UEVENT[1138806694] mount@/block/sdb/sdb1
UEVENT[1138806697] umount@/block/sdb/sdb1
```

11.8 User Information

11.8.1 Who Is Doing What: w

With the command `w`, find out who is logged onto the system and what each user is doing. For example:

```
tux@mercury:~> w
 14:58:43 up 1 day,  1:21,  2 users,  load average: 0.00, 0.00, 0.00
USER     TTY        LOGIN@   IDLE   JCPU   PCPU WHAT
tux      :0        12:25   ?xdm?   1:23   0.12s /bin/sh /usr/bin/startkde
root     pts/4     14:13    0.00s  0.06s  0.00s w
```

If any users of other systems have logged in remotely, the parameter `-f` shows the computers from which they have established the connection.

11.9 Time and Date

11.9.1 Time Measurement with time

Determine the time spent by commands with the `time` utility. This utility is available in two versions: as a shell built-in and as a program (`/usr/bin/time`).

```
tux@mercury:~> time find . > /dev/null

real    0m4.051s
user    0m0.042s
sys     0m0.205s
```

Updating the System and System Changes 12

You can update an existing system without completely reinstalling it. There are two types of updates: *updating individual software packages* and *updating the entire system*.

12.1 Updating the System

Software tends to “grow” from version to version. Therefore, take a look at the available partition space with `df` before updating. If you suspect you are running short of disk space, secure your data before you update and repartition your system. There is no general rule of thumb regarding how much space each partition should have. Space requirements depend on your particular partitioning profile, the software selected, and the version numbers of the system.

12.1.1 Preparations

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

Before starting your update, make note of the root partition. The command `df /` lists the device name of the root partition. In Example 12.1, “List with df -h” (page 166), the root partition to write down is `/dev/sda3` (mounted as `/`).

Example 12.1 *List with df -h*

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        74G   22G   53G  29% /
udev            252M  124K  252M   1% /dev
/dev/sda5       116G  5.8G  111G   5% /home
/dev/sda1        39G  1.6G   37G   4% /windows/C
/dev/sda2       4.6G  2.6G  2.1G  57% /windows/D
```

12.1.2 Possible Problems

If you update a default system from the previous version to this version, YaST works out necessary changes and performs them. Depending on your customizations, some steps or the entire update procedure may fail and you must resort to copying back your backup data. Here, we point out more issues to check before starting the system update.

Checking passwd and group in /etc

Before updating the system, make sure that `/etc/passwd` and `/etc/group` do not contain any syntax errors. For this purpose, start the verification utilities `pwck` and `grpck` as `root` and eliminate any reported errors.

PostgreSQL

Before updating PostgreSQL (`postgres`), dump the databases. See the manual page of `pg_dump`. This is only necessary if you actually used PostgreSQL prior to your update.

12.1.3 Updating with YaST

Following the preparation procedure outlined in Section 12.1.1, “Preparations” (page 165), you can now update your system:

1. Boot the system as for the installation, described in Section “System Start-Up for Installation” (Chapter 1, *Installation with YaST*, ↑Start-Up). In YaST, choose a language and select *Update* in the *Installation Mode* dialog. Do not select *New Installation*. Also add repositories to make sure to get all available software up-

dated whenever possible. Find more information about installation repositories in Section “Add-On Products” (Chapter 1, *Installation with YaST*, ↑Start-Up).

2 YaST determines whether there are multiple root partitions. If there is only one, continue with the next step. If there are several, select the right partition and confirm with *Next* (`/dev/sda3` was selected in the example in Section 12.1.1, “Preparations” (page 165)). YaST reads the old `fstab` on this partition to analyze and mount the file systems listed there.

3 Check the previously used repositories, if there are any. Enable all the repositories you still want to use and update third-party software from. Click the *Toggle Status* for every list item, if appropriate.

4 In case you added repositories during the update procedure as recommended above, you now can activate those you are actually interested in.

5 In the *Installation Settings* dialog, adjust the settings according to your requirements. Normally, you can leave the default settings untouched, but if you intend to enhance your system, check the packages offered in the *Software Selection* submenus or add support for additional languages.

   You also have the possibility to make backups of various system components. Selecting backups slows down the update process. Use this option if you do not have a recent system backup.

6 Confirm the update by clicking *Start Update*.

Once the basic update installation is finished, test the Internet connection as offered by the YaST dialog. Finally, YaST updates the remaining software, offers the Novell Customer Center Configuration, and displays the release notes. Click *Finish* to write the YaST configuration.

For more information about the Novell Customer Center, see Section “Registration” (Chapter 1, *Installation with YaST*, ↑Start-Up).

12.1.4 Updating Individual Packages

Regardless of your overall updated environment, you can always update individual packages. From this point on, however, it is your responsibility to ensure that your system remains consistent. Update advice can be found at `http://www.novell.com/linux/download/updates/`.

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

12.2 Software Changes from Version to Version

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

Problems and special issues of the respective versions are published online as they are identified. See the links listed below. Important updates of individual packages can be accessed at `http://www.novell.com/products/linuxprofessional/downloads/` using the YaST Online Update. For more information, see Chapter 5, *YaST Online Update* (page 79).

12.2.1 From 10.1 to 10.2

Refer to the "Bugs" article in the openSUSE wiki at `http://en.opensuse.org/Bugs`.

The Standard Kernel

The `kernel-default` package contains the standard kernel for both, uniprocessor and multiprocessor systems. The kernel comes with SMP support and runs with minimal overhead on uniprocessor systems. There is no `kernel-smp` package anymore.

Add-On Medium with Additional Languages

Include the language add-on medium in your list of installation sources, if you want better support for one of our tier 2 languages. Tier 2 languages are all but the tier 1 languages (English, French, German, Italian, Spanish, Brazilian Portuguese, simplified and traditional Chinese, Japanese, and Czech). Support for tier 1 languages is available on the standard media set.

12.2.2 From 10.2 to 10.3

Refer to the `Bugs` article in the openSUSE wiki at http://en.opensuse.org/Bugs.

Text Installation Pattern

The scope of the text installation pattern is very limited. It is not recommended to install this pattern without adding additional software. Add packages from other patterns. The purpose of this pattern is to have a minimal bootable system running on real hardware. It provides a multiuser system with local login, network setup, and the default filesystems. No service is enabled by default and the only YaST modules installed are those needed during installation.

Adding Extra Software Repositories During Installation

After setting up the update configuration at the end of the installation, YaST offers to add the following three software repositories as additional installation sources:

- The “oss” repository contains the complete FTP distribution including more packages than available on the CDs.

- The “non-oss” repository contains software under a propietary or non-open source license.

- The “debug” repository contains debuginfo packages used for debugging programs and libraries and getting backtraces. If an error occurs, this additional information helps you write good bug reports.

The source RPMs for “oss” are available at http://download.opensuse.org/distribution/10.3/src-oss, the source RPMs for “non-oss” are available at http://download.opensuse.org/distribution/10.3/src-non-oss.

Localization Support

The 1-CD installation media (GNOME or KDE) come with language support for American English only.

Support for all the other languages is available separately. If you are interested in additional languages, add an extra online repository during installation offering these translations. The “oss” repository, as mentioned above in Section “Adding Extra Software Repositories During Installation” (page 169), is such a repository.

AppArmor 2.1

Find more detailed information about new features at http://en.opensuse.org/AppArmor/Changes_AppArmor_2_1.

The syntax now distinguishes directories from files. There are additional minor syntax bug fixes.

The reporting of change_hat related events and information has changed. The log messages and profile state (as available via /proc/<pid>/attr/current) are reported as `/profile//hat`.

A new change_profile policy specification has been added. change_profile is similar to change_hat, but allows changing to any profile, including hats. The profiles you can change to must be specified, that is the only restriction. To change to a hat via change_profile, the hat name hsa to be specified by separating the profile and hat_name with `//` .

GAIM Renamed to Pidgin

The GAIM instant messenger has been renamed to Pidgin.

New Location for KDE and GNOME

GNOME 2 is installed under the `/usr` file system hierarchy since openSUSE 10.3 and KDE 4 now follows. KDE 3 will stay in `/opt` for compatibility reasons.

Before starting the update, make sure that there is enough disk space under `/usr` (approx. 2.5GB for both desktops is required). If you are short on space under `/usr`, resize or rearrange your partitions.

Berkeley DB Change Affects OpenLDAP Server

There is a format change in Berkeley DB's on-disk log files between Berkely DB 4.3 and 4.4. This change prevents an installed OpenLDAP server from starting after the system update.

To avoid this issue, export existing LDAP Databases using the `slapcat` utility *before* starting the system update and re-import the data using `slapadd` after the update. On an already updated machine get the LDAP Server running as follows:

1. Stop the LDAP Server.

2. Remove all files starting with `_db.` from the database directory.

3. Start the LDAP server again.

libata for IDE Devices

libata uses `/dev/sda` for the first hard disk instead of `/dev/hda`. Disks with more than 15 partitions are not handled automatically right now. You can disable libata support by booting with the following kernel parameter:

```
hwprobe=-modules.pata
```

Now you see all the partitions over 15 again and can access them for installation.

Changes in Setting up Encrypted Partitions

The back-end technology of `boot.crypto` has been changed from `cryptoloop` to `dm-crypt`.

Any old `/etc/cryptotab` will work unmodified on openSUSE 10.3 (modulo partition renaming issues from `hdX` to `sdX` due to libata changes—see Section "libata for IDE Devices" (page 171)). Additionally, `/etc/crypttab` (note the missing `o`) is now supported, which also includes support for LUKS volumes. In contrast to previous releases, `boot.crypto` is no longer enabled by default. YaST enables it if you create an encrypted volume with YaST. You can also manually enable it with the following command:

```
chkconfig boot.crypto on
```

It is still possible to use cryptoloop via `losetup` and `mount`. Since we dropped the crude loop-AES patch from the util-linux package, some parameters for `losetup` (such as `itercountk` and `pseed`) no longer exist. If any of these settings are used in `/etc/fstab`, the device cannot be mounted directly any more. Migrate these settings to `/etc/crypttab` where `boot.crypto` contains the necessary compatability code.

Enabling Quota Support

You now can configure quota for user accounts from within YaST. To enable quota support activate the *Enable Quota Support* check box in the fstab options when partitioning in the first stage of the installation. Thus, ensure that the `/etc/init.d/boot.quota` script is executed at boot time. In the second stage, the advanced options for user accounts provide the quota module where quota rules can be set.

If you enable quota support in the partitioner in the running system after the installation, either reboot the system or manually remount the partitions concerned and execute the following command as `root`:

```
/etc/init.d/boot.quota restart
```

Zeroconf

The Zeroconf service—also known as Bonjour, Multicast DNS, mDNS, or DNS-SD—is now provided by the Avahi stack instead of mDNSResponder. However, the mDNSResponder and howl compatibility libraries are still available.

To enable mDNS for all network interfaces, use the *Zeroconf/Bonjour Multicast DNS* SuSEfirewall rule.

Older Intel Graphics Chips

Older Intel graphics chips are supported by two drivers: i810 and intel. Due to the high demand for features like native mode setting (no longer VESA BIOS based) and RANDR 1.2 support, the intel driver is the default on openSUSE 10.3.

When updating to openSUSE 10.3, the i810 driver is not exchanged with the intel driver. Use the `sax2 -r` command to switch to the intel driver.

The intel driver is still not as stable as i810. Use the `sax2 -r -m 0=i810` command to switch back to i810, if you encounter problems that did not occur previously with the i810 driver. In those cases, consider to open a bug report against the intel driver.

Intel Wireless Link WiFi Drivers

Two drivers are available now: the traditional `ipw3945` driver, which is installed by default, and the new `iwlwifi` driver as an alternative offer. Note the following caveats:

- `ipw3945` works on hidden networks. It does not survive suspend/resume cycles.

- `iwlwifi` does not work on hidden networks. It supports suspend/resume cycles.

You can change the default using YaST. Click *Software* > *Software Management* and remove the `ipw3945d` package. The alternative `iwlwifi` driver is now selected automatically for installation.

Tools to Write Optical Disc Media (CD-ROM and DVD)

The cdrecord package has been dropped from the distribution. The new wodim, genisoimage, and icedax packages from the cdrkit project can be used to record data or audio CDs on a CD recorder that conforms with the Orange Book standard. The following binaries have been renamed:

- cdrecord to wodim
- readcd to readom
- mkisofs to genisoimage
- cdda2wav to icedax

If your application relies on the old names, install the cdrkit-cdrtools-compat package. However, it would be appropriate to have native support for wodim in all front-end applications, because it offers some improvements:

- The preferred way of specifying a device is `dev=/dev/cdrecorder`, `dev=/dev/hdc`, `dev=/dev/sr0`, etc.
- Available devices can be listed with `wodim -devices`.
- Suid root is not needed.

If you maintain such a front-end or script, consider adding native wodim support.

Use `growisofs` for writing DVDs. The graphical front-ends handle this transparently.

KDE 4 Applications Path

If you did not install the KDE desktop during the initial openSUSE 10.3 installation, install the KDE Base System and KDE 4 Base System patterns later. The KDE 4 application path is used as default. If you launch a KDE application such as Konqueror, the KDE 4 version of Konqueror loads instead of the KDE 3 version.

Playing MP3 Files in Kaffeine

When you open an MP3 file in Kaffeine, you will see an error message telling you that the software required to play this file is not installed. openSUSE then offers to search for a suitable codec which you can install with YaST. You can also switch the engine from Xine to Gstreamer by clicking *Settings* > *Player Engine* to get MP3 support.

12.2.3 From 10.3 to 11.0

Refer to the `Bugs` article in the openSUSE wiki at http://en.opensuse.org/Bugs.

New On-Disk Format of sysstat

The new features of the sysstat package comming with 11.0 require a changed on-disk format of the data files. After the update of the sysstat package old collected data cannot be used any longer.

Part IV. System

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

13

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

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

13.1 Runtime Support

IMPORTANT: Conflicts between Application Versions

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

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

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

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

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

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

13.2 Software Development

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

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

13.3 Software Compilation on Biarch Platforms

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

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

The following example refers to an x86_64 system with x86 as the second architecture.

1. Use the 32-bit compiler:

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

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

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

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

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

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

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

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

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

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

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

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

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

13.4 Kernel Specifications

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

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

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

TIP

Some applications require separate kernel-loadable modules. If you intend to use such a 32-bit application in a 64-bit system environment, contact the provider of this application and Novell to make sure that the 64-bit version of the kernel-loadable module and the 32-bit compiled version of the kernel API are available for this module.

Booting and Configuring a Linux System

14

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

14.1 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 the computer has been turned on, the BIOS initializes the screen and keyboard and tests the main memory. Up to this stage, the machine does not access any mass storage media. Subsequently, the information about the current date, time, and the most important peripherals are loaded from the CMOS values. When the first hard disk and its geometry are recognized, the system control passes from the BIOS to the boot loader.

2. **Boot Loader** The first physical 512-byte data sector of the first hard disk is loaded into the main memory and the *boot loader* that resides at the beginning of this sector takes over. The commands executed by the boot loader determine the remaining part of the boot process. Therefore, the first 512 bytes on the first hard disk are referred to as the *Master Boot Record* (MBR). The boot loader then passes control to the actual operating system, in this case, the Linux kernel. More

information about GRUB, the Linux boot loader, can be found in Chapter 15, *The Boot Loader GRUB* (page 199).

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 14.1.1, "initramfs" (page 184).

4. **init on initramfs** This program performs all actions needed to mount the proper root file system, like providing kernel functionality for the needed file system and device drivers for mass storage controllers with udev. After the root file system has been found, it is checked for errors and mounted. If this has been successful, the initramfs is cleaned and the init program on the root file system is executed. For more information about init, refer to Section 14.1.2, "init on initramfs" (page 185). Find more information about udev in Chapter 17, *Dynamic Kernel Device Management with udev* (page 235).

5. **init** init handles the actual booting of the system through several different levels providing different functionality. init is described in Section 14.2, "The init Process" (page 187).

14.1.1 initramfs

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

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

If you need to change hardware (e.g. hard disks) in an installed system and this hardware requires different drivers to be present in the kernel at boot time, you must update initramfs. This is done in the same way as with its predecessor, initrd—by calling `mkinitrd`. Calling `mkinitrd` without any argument creates an initramfs. Calling `mkinitrd -R` creates an initrd. In openSUSE®, the modules to load are specified by the variable `INITRD_MODULES` in `/etc/sysconfig/kernel`. After installation, this variable is automatically set to the correct value. The modules are loaded in exactly the order in which they appear in `INITRD_MODULES`. This is only important if you rely on the correct setting of the device files `/dev/sd?`. However, in current systems you also may use the device files below `/dev/disk/` that are sorted in several subdirectories, named `by-id`, `by-path` and `by-uuid`, and always represent the same disk. This is also possible at install time by specifying the respective mount option.

IMPORTANT: Updating initramfs or initrd

The boot loader loads initramfs or initrd in the same way as the kernel. It is not necessary to reinstall GRUB after updating initramfs or initrd, because GRUB searches the directory for the right file when booting.

14.1.2 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 2, *Advanced Disk Setup* (page 41).

Managing Network Configuration

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

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

Finding the Installation Medium

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

Initiating Hardware Recognition and Loading Appropriate Kernel Modules

As mentioned in Section 14.1.1, "initramfs" (page 184), the boot process starts with a minimum set of drivers that can be used with most hardware configurations. init starts an initial hardware scanning process that determines the set of drivers suitable for your hardware configuration. The names of the modules needed for the boot process are written to `INITRD_MODULES` in `/etc/sysconfig/kernel`. These names are used to generate a custom initramfs that is needed to boot the system. If the modules are not needed for boot but for coldplug, the modules are written to `/etc/sysconfig/hardware/hwconfig-*`. All devices that are described with configuration files in this directory are initialized in the boot process.

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

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

14.2 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 14.2.1, "Runlevels" (page 187)). 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 14.2.2, "Init Scripts" (page 190)).

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

14.2.1 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 14.1, "Available Runlevels" (page 188). 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 14.1 *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	*User Defined*, 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 openSUSE standard installations. Users are prompted for login with a graphical interface or the default user is logged in automatically. If the default runlevel is `3`, the X Window System must be configured properly, as described in Chapter 10, *The X Window System* (page 131), before the runlevel can be switched to `5`. If this is done, check whether the system works in the desired way by entering `telinit 5`. If everything turns out as expected, you can use YaST to set the default runlevel to `5`.

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

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

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

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

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

14.2.2 Init Scripts

There are two types of scripts in `/etc/init.d`:

Scripts Executed Directly by init
: This is the case only during the boot process or if an immediate system shutdown is initiated (power failure or a user pressing Ctrl + Alt + Del). The execution of these scripts is defined in `/etc/inittab`.

Scripts Executed Indirectly by init
: These are run when changing the runlevel and always call the master script `/etc/init.d/rc`, which guarantees the correct order of the relevant scripts.

All scripts are located in `/etc/init.d`. Scripts that are run at boot time are called through symbolic links from `/etc/init.d/boot.d`. Scripts for changing the runlevel are called through symbolic links from one of the subdirectories (`/etc/init.d/rc0.d` to `/etc/init.d/rc6.d`). This is just for clarity reasons and avoids duplicate scripts if they are used in several runlevels. Because every script can be executed as both a start and a stop script, these scripts must understand the parameters `start` and `stop`. The scripts also understand the `restart`, `reload`, `force-reload`, and `status` options. These different options are explained in Table 14.2, "Possible init Script Options" (page 191). Scripts that are run directly by init do not have these links. They are run independently from the runlevel when needed.

Table 14.2 *Possible init Script Options*

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

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

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

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

`boot`
: Executed while starting the system directly using init. It is independent of the chosen runlevel and is only executed once. Here, the `/proc` and `/dev/pts` file systems are mounted and blogd (boot logging daemon) is activated. If the system is booted for the first time after an update or an installation, the initial system configuration is started.

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

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

`boot.local`
: Here, enter additional commands to execute at boot before changing into a runlevel. It can be compared to `AUTOEXEC.BAT` on DOS systems.

`halt`
: This script is only executed while changing into runlevel 0 or 6. Here, it is executed either as `halt` or as `reboot`. Whether the system shuts down or reboots depends on how `halt` is called. If special commands are needed during the shutdown, add these to the `halt.local` script.

`rc`
: This script calls the appropriate stop scripts of the current runlevel and the start scripts of the newly selected runlevel. Like the `/etc/init.d/boot` script, this script is called from `/etc/inittab` with the desired runlevel as parameter.

You can create your own scripts and easily integrate them into the scheme described above. For instructions about formatting, naming, and organizing custom scripts, refer to the specifications of the LSB and to the man pages of `init`, `init.d`, `chkconfig`, and `insserv`. Additionally consult the man pages of `startproc` and `killproc`.

WARNING: Faulty init Scripts May Halt Your System

Faulty init scripts may hang your machine. Edit such scripts with great care and, if possible, subject them to heavy testing in the multiuser environment. Find some useful information about init scripts in Section 14.2.1, "Runlevels" (page 187).

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 14.1, "A Minimal INIT INFO Block" (page 193).

***Example 14.1** A Minimal INIT INFO Block*

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

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

To create the links from the runlevel directories (`/etc/init.d/rc?.d/`) to the corresponding scripts in `/etc/init.d/`, enter the command `insserv` *`new-script-name`*. The insserv program evaluates the `INIT INFO` header to create the necessary links for start and stop scripts in the runlevel directories (`/etc/init.d/rc?.d/`). The program also takes care of the correct start and stop order for each runlevel by including the necessary numbers in the names of these links. If you prefer a graphical tool to create such links, use the runlevel editor provided by YaST, as described in Section 14.2.3, "Configuring System Services (Runlevel) with YaST" (page 194).

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.

14.2.3 Configuring System Services (Runlevel) with YaST

After starting this YaST module with *YaST* > *System* > *System Services (Runlevel)*, it displays an overview listing all the available services and the current status of each service (disabled or enabled). Decide whether to use the module in *Simple Mode* or in *Expert Mode*. The default *Simple Mode* should be sufficient for most purposes. The left column shows the name of the service, the center column indicates its current status, and the right column gives a short description. For the selected service, a more detailed description is provided in the lower part of the window. To enable a service, select it in the table then select *Enable*. The same steps apply to disable a service.

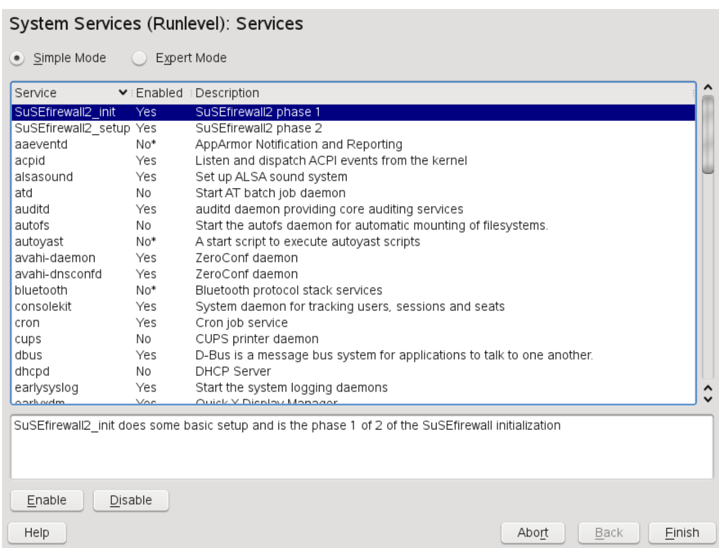
For detailed control over the runlevels in which a service is started or stopped or to change the default runlevel, first select *Expert Mode*. The current default runlevel or "initdefault" (the runlevel into which the system boots by default) is displayed at the top. Normally, the default runlevel of a openSUSE system is runlevel 5 (full multiuser mode with network and X). A suitable alternative might be runlevel 3 (full multiuser mode with network).

This YaST dialog allows the selection of one of the runlevels (as listed in Table 14.1, “Available Runlevels” (page 188)) 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 14.1 *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.

14.3 System Configuration via /etc/sysconfig

The main configuration of openSUSE is controlled by the configuration files in `/etc/sysconfig`. The individual files in `/etc/sysconfig` are only read by the scripts to which they are relevant. This ensures that network settings, for example, only need to be parsed by network-related scripts.

There are two ways to edit the system configuration. Either use the YaST sysconfig Editor or edit the configuration files manually.

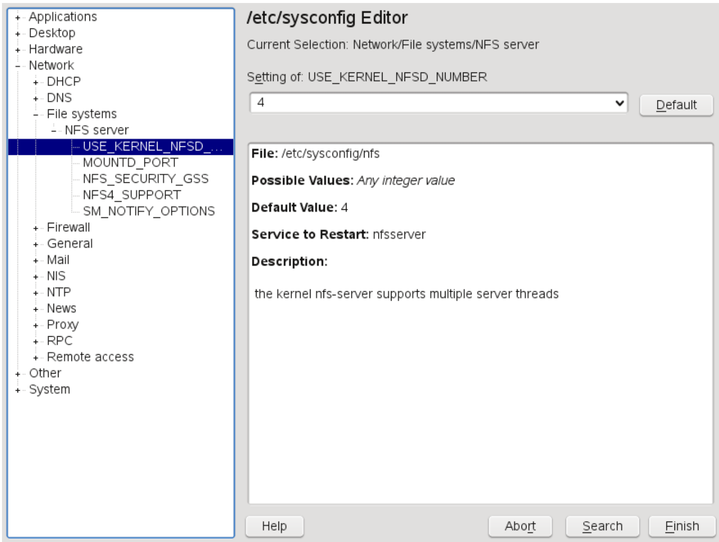
14.3.1 Changing the System Configuration Using the YaST sysconfig Editor

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

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

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

Figure 14.2 *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.

14.3.2 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.

15 The Boot Loader GRUB

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

This chapter focuses on boot management and the configuration of the boot loader GRUB. The boot procedure as a whole is outlined in Chapter 14, *Booting and Configuring a Linux System* (page 183). A boot loader represents the interface between the machine (BIOS) and the operating system (openSUSE). The configuration of the boot loader directly impacts the start of the operating system.

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

Master Boot Record
: The structure of the MBR is defined by an operating system–independent convention. The first 446 bytes are reserved for the program code. They typically hold part of a boot loader program or an operating system selector. The next 64 bytes provide space for a partition table of up to four entries. The partition table contains information about the partitioning of the hard disk and the file system types. The operating system needs this table for handling the hard disk. With conventional generic code in the MBR, exactly one partition must be marked *active*. The last two bytes of the MBR must contain a static “magic number” (AA55). An MBR containing a different value is regarded as invalid by some BIOSes, so is not considered for booting.

Boot Sectors

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

15.1 Booting with GRUB

GRUB (Grand Unified Bootloader) comprises two stages. Stage 1 consists of 512 bytes and its only task is to load the second stage of the boot loader. Subsequently, stage 2 is loaded. This stage contains the main part of the boot loader.

In some configurations, an intermediate stage 1.5 can be used, which locates and loads stage 2 from an appropriate file system. If possible, this method is chosen by default on installation or when initially setting up GRUB with YaST.

Stage 2 is able to access many file systems. Currently, Ext2, Ext3, ReiserFS, Minix, and the DOS FAT file system used by Windows are supported. To a certain extent, XFS, and UFS and FFS used by BSD systems are also supported. Since version 0.95, GRUB is also able to boot from a CD or DVD containing an ISO 9660 standard file system pursuant to the “El Torito” specification. Even before the system is booted, GRUB can access file systems of supported BIOS disk devices (floppy disks or hard disks, CD drives, and DVD drives detected by the BIOS). Therefore, changes to the GRUB configuration file (`menu.lst`) do not require a new installation of the boot manager. When the system is booted, GRUB reloads the menu file with the valid paths and partition data of the kernel or the initial RAM disk (`initrd`) and locates these files.

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

`/boot/grub/menu.lst`
: This file contains all information about partitions or operating systems that can be booted with GRUB. Without this information, the GRUB command line prompts the user for how to proceed (see Section "Editing Menu Entries during the Boot Procedure" (page 206) for details).

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

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

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

In GRUB, all boot parameters can be changed prior to booting. For example, errors made when editing the menu file can be corrected in this way. Boot commands can also be entered interactively at a kind of input prompt (see Section "Editing Menu Entries during the Boot Procedure" (page 206)). GRUB offers the possibility of determining the location of the kernel and the `initrd` prior to booting. In this way, you can even boot an installed operating system for which no entry exists in the boot loader configuration.

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

15.1.1 The GRUB Boot Menu

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

Every time the system is booted, GRUB loads the menu file from the file system. For this reason, GRUB does not need to be reinstalled after every change to the file. Use the YaST boot loader to modify the GRUB configuration as described in Section 15.2, "Configuring the Boot Loader with YaST" (page 209).

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 203). 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 204).

Naming Conventions for Hard Disks and Partitions

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

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

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

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

Unfortunately, it is often not possible to map the Linux device names to BIOS device names exactly. It generates this mapping with the help of an algorithm and saves it to the file `device.map`, which can be edited if necessary. Information about the file `device.map` is available in Section 15.1.2, "The File device.map" (page 207).

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 openSUSE. The kernel (`vmlinuz`) is located in the first logical partition (the boot partition) of the first hard disk. Kernel parameters, such as the root partition and VGA mode, are appended here. The root partition is specified according to the Linux naming convention (`/dev/sda7/`), because this information is read by the kernel and has nothing to do with GRUB. The `initrd` is also located in the first logical partition of the first hard disk.

- 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 206).

Editing Menu Entries during the Boot Procedure

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

IMPORTANT: Keyboard Layout during the Boot Procedure

The US keyboard layout is the only one available when booting. See Figure “US Keyboard Layout” (↑Start-Up) for a figure.

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

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

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

```
title linux
   root(hd0,0)
   kernel /vmlinuz root=/dev/sda3 additional parameter
   initrd /initrd
```

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

15.1.2 The File device.map

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

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

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

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

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

15.1.3 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.

15.1.4 Setting a Boot Password

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

IMPORTANT: Boot Password and Splash Screen

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

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

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

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

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

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

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

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

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

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

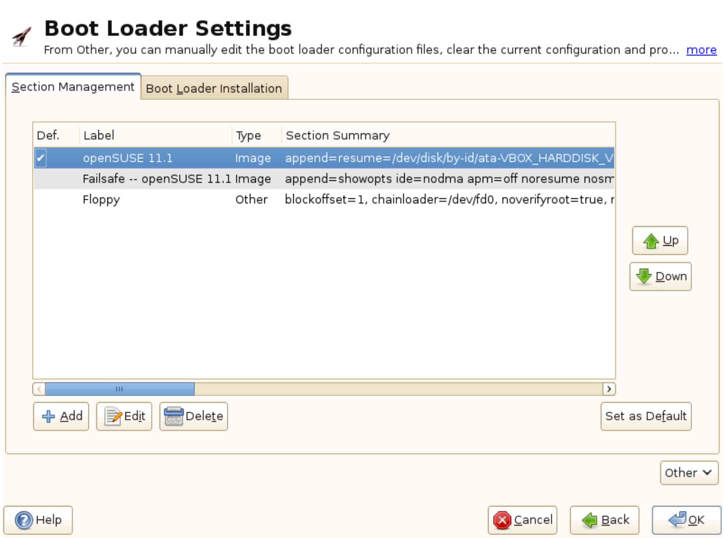
```
Error 32: Must be authenticated
```

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

15.2 Configuring the Boot Loader with YaST

The easiest way to configure the boot loader in your openSUSE system is to use the YaST module. In the YaST Control Center, select *System* > *Boot Loader*. As in Figure 15.1, "Boot Loader Settings" (page 209), this shows the current boot loader configuration of your system and allows you to make changes.

Figure 15.1 *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 15.1, "Booting with GRUB" (page 200) 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 15.1, "Booting with GRUB" (page 200) 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*.

15.2.1 Adjusting the Default Boot Entry

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

Procedure 15.1 *Setting the Default System*

1. Open the *Section Management* tab.

2. Select the desired entry from the list.

3. Click *Set as Default*.

4. Click *Finish* to activate these changes.

15.2.2 Modifying the Boot Loader Location

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

Procedure 15.2 *Changing the Boot Loader Location*

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

Boot from Boot Partition
: The boot sector of the `/boot` partition.

Boot from Extended Partition
: This installs the boot loader in the extended partition container.

Boot from Master Boot Record
: This installs the boot loader in the MBR of the first disk (according to the boot sequence preset in the BIOS).

Boot from Root Partition
: This installs the boot loader in the boot sector of the `/` partition.

Custom Boot Partition
: Use this option to specify the location of the boot loader manually.

2 Click *Finish* to apply your changes.

15.2.3 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 15.3 *Changing the Boot Loader Time-Out*

1 Open the *Boot Loader Installation* tab.

2 Click *Boot Loader Options*.

3 Change the value of *Time-Out in Seconds* by typing in a new value, clicking the appropriate arrow key with your mouse, or by using the arrow keys on the keyboard.

4 Click *OK*.

5 Click *Finish* to save the changes.

15.2.4 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 15.4 *Setting a Boot Loader Password*

1 Open the *Boot Loader Installation* tab.

2 Click *Boot Loader Options*.

3 Set your password in *Password for the Menu Interface*.

4 Click *OK*.

5 Click *Finish* to save the changes.

15.2.5 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 15.1.2, “The File device.map” (page 207)). To do so, proceed as follows:

Procedure 15.5 *Setting the Disk Order*

1 Open the *Boot Loader Installation* tab.

2 Click *Boot Loader Installation Details*.

3 If more than one disk is listed, select a disk and click *Up* or *Down* to reorder the displayed disks.

4 Click *OK* to save the changes.

5 Click *Finish* to save the changes.

15.2.6 Configuring Advanced Options

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

Set Active Flag in Partition Table for Boot Partition
: Activates the partition that contains the boot loader. Some legacy operating systems, such as Windows 98, can only boot from an active partition.

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

Write Generic Boot Code to MBR
: Replaces the current MBR with generic, operating system independent code.

Hide Boot Menu
: Hides the boot menu and boots the default entry.

Use Trusted GRUB
: Starts the Trusted GRUB which supports trusted computing functionality.

Graphical Menu File
: Path to the graphics file used when displaying the boot screen.

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

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

15.2.7 Changing Boot Loader Type

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

Procedure 15.6 *Changing the Boot Loader Type*

1. Select the *Boot Loader Installation* tab.

2. For *Boot Loader*, select *LILO*.

3. In the dialog box that opens, select one of the following actions:

   Propose New Configuration
   : Have YaST propose a new configuration.

   Convert Current Configuration
   : Have YaST convert the current configuration. When converting the configuration, some settings may be lost.

   Start New Configuration from Scratch
   : Write a custom configuration. This action is not available during the installation of openSUSE.

   Read Configuration Saved on Disk
   : Load your own `/etc/lilo.conf`. This action is not available during the installation of openSUSE.

4. Click *OK* to save the changes

5. Click *Finish* in the main dialog to apply the changes.

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

NOTE: Custom Boot Loader

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

15.3 Uninstalling the Linux Boot Loader

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

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

15.4 Creating Boot CDs

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

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

Procedure 15.7 *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 `(hd`*x*`,`*y*`)`, in the pathnames with `(cd)`, the device name of the CD-ROM drive. You may also need to adjust the paths to the message file, the kernel, and the initrd—they need to point to `/boot/message`, `/boot/vmlinuz` and `/boot/initrd`, respectively. After having made the adjustments, `menu.lst` should look similar to the following example:

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

title Linux
   root (cd)
   kernel /boot/vmlinuz root=/dev/sda5 vga=794 resume=/dev/sda1 \
   splash=verbose showopts
   initrd /boot/initrd
```

Use `splash=silent` instead of `splash=verbose` to prevent the boot messages from appearing during the boot procedure.

5 Create the ISO image with the following command:

```
genisoimage -R -b boot/grub/stage2_eltorito -no-emul-boot \
-boot-load-size 4 -boot-info-table -iso-level 2 -input-charset utf-8 \
-o grub.iso /tmp/iso
```

6 Write the resulting file `grub.iso` to a CD using your preferred utility. Do not burn the ISO image as data file, but use the option for burning a CD image in your burning utility.

15.5 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 15, *The Boot Loader GRUB* (page 199) provides more information about this. However, if you prefer the text mode, which was the default in earlier versions, set `vga=normal`.

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

TIP

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

15.6 Troubleshooting

This section lists some of the problems frequently encountered when booting with GRUB and a short description of possible solutions. Some of the problems are covered in articles in the Support Database at `http://en.opensuse.org/SDB:SDB`. Use the search dialog to search for keywords like *GRUB*, *boot*, and *boot loader*.

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

GRUB Reports GRUB Geom Error
: GRUB checks the geometry of connected hard disks when the system is booted. Sometimes, the BIOS returns inconsistent information and GRUB reports a GRUB Geom Error. If this is the case, update the BIOS.

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

System Containing Several Hard Disks Does Not Boot
: During the installation, YaST may have incorrectly determined the boot sequence of the hard disks. For example, GRUB may regard the IDE disk as `hd0` and the SCSI disk as `hd1`, although the boot sequence in the BIOS is reversed (SCSI *before* IDE).

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

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

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

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

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

15.7 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 "SDB:GRUB" in the Support Database at `http://www.opensuse.org/` to get information about special issues.

16 Special System Features

This chapter starts with information about various software packages, the virtual consoles, and the keyboard layout. We talk about software components like `bash`, `cron`, and `logrotate`, because they were changed or enhanced during the last release cycles. Even if they are small or considered of minor importance, users may want to change their default behavior, because these components are often closely coupled with the system. The chapter is finished by a section about language and country-specific settings (I18N and L10N).

16.1 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.

16.1.1 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.

16.1.2 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 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 16.1, "Entry in /etc/crontab" (page 222), `root` is entered. Package-specific tables, located in `/etc/cron.d`, have the same format. See the `cron` man page (`man cron`).

***Example 16.1** Entry in /etc/crontab*

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

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

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

To run the hourly, daily, or other periodic maintenance scripts at custom times, remove the time stamp files regularly using /etc/crontab entries (see Example 16.2, "/etc/crontab: Remove Time Stamp Files" (page 223), which removes the hourly one before every full hour, the daily one once a day at 2:14 a.m., etc.).

Example 16.2 */etc/crontab: Remove Time Stamp Files*

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

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

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

16.1.3 Log Files: Package logrotate

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

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

***Example 16.3** 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.

16.1.4 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.

16.1.5 The ulimit Command

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

`ulimit` can be used with various options. To limit memory usage, use the options listed in Table 16.1, "ulimit: Setting Resources for the User" (page 225).

***Table 16.1** ulimit: Setting Resources for the User*

`-m`	The maximum resident set size
`-v`	The maximum amount of virtual memory available to the shell
`-s`	The maximum size of the stack
`-c`	The maximum size of core files created
`-a`	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 16.4 *ulimit: Settings in ~/.bashrc*

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

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

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

IMPORTANT

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

16.1.6 The free Command

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

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

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

16.1.7 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.

16.1.8 Selecting Man-Pages Using the man Command

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

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

16.1.9 Settings for GNU Emacs

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

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

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

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

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

The components of Emacs are divided into several packages:

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

16.2 Virtual Consoles

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

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

16.3 Keyboard Mapping

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

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

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

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

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

TIP: For More Information

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

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

16.4 Language and Country-Specific Settings

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

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

`RC_LC_MESSAGES`, `RC_LC_CTYPE`, `RC_LC_COLLATE`, `RC_LC_TIME`, `RC_LC_NUMERIC`, `RC_LC_MONETARY`
: These variables are passed to the shell without the `RC_` prefix and represent the listed categories. The shell profiles concerned are listed below. The current setting can be shown with the command `locale`.

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

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

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

The variables can be set with the YaST sysconfig editor (see Section 14.3.1, “Changing the System Configuration Using the YaST sysconfig Editor” (page 196)). 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>]]
```

16.4.1 Some Examples

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

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

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

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

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

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

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

profile. /etc/SuSEconfig/csh.cshrc is sourced by /etc/csh.cshrc. This makes the settings available systemwide.

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

16.4.2 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
```

16.4.3 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.

16.4.4 For More Information

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

Dynamic Kernel Device Management with udev

17

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

17.1 The /dev Directory

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

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

17.2 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 17.6, "Influencing Kernel Device Event Handling with udev Rules" (page 239).

Every received event is matched against the set of provides 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.

17.3 Drivers, Kernel Modules, and Devices

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

```
MODALIAS=usb:v046DpC03Ed2000dc00dsc00dp00ic03isc01ip02
```

Every device driver carries a list of known aliases for devices it can handle. The list is contained in the kernel module file itself. The program depmod reads the ID lists and creates the file `modules.alias` in the kernel's `/lib/modules` directory for all currently available modules. With this infrastructure, module loading is as easy as calling `modprobe` for every event that carries a `MODALIAS` key. If `modprobe $MODALIAS` is called, it matches the device alias composed for the device with the aliases provided by the modules. If a matching entry is found, that module is loaded. All this is triggered by udev and happens automatically.

17.4 Booting and Initial Device Setup

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

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

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

17.5 Monitoring the Running udev Daemon

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

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

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

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

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

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

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

17.6 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 17.1 *Example udev Rules*

```
# console
KERNEL=="console", MODE="0600", OPTIONS="last_rule"

# serial devices
KERNEL=="ttyUSB*", ATTRS{product}=="[Pp]alm*Handheld*", SYMLINK+="pilot"

# printer
SUBSYSTEM=="usb", KERNEL=="lp*", NAME="usb/%k", SYMLINK+="usb%k", GROUP="lp"

# kernel firmware loader
SUBSYSTEM=="firmware", ACTION=="add", RUN+="firmware.sh"
```

The `console` rule consists of three keys: one match key (`KERNEL`), and two assign keys (`MODE`, `OPTIONS`). The `KERNEL` match rule searches the device list for any items of the type `console`. Only exact matches are valid and trigger this rule to be executed. The `MODE` key assigns special permissions to the device node, in this case, read and write permissions to the owner of this device only. The `OPTIONS` key makes this rule the last rule to be applied to any device of this type. Any later rule matching this particular device type does not have any effect.

The `serial devices` rule is not available in `50-udev-default.rules` anymore, but it is still worth a look. It consists of two match keys (`KERNEL` and `ATTRS`) and one assign key (`SYMLINK`). The `KERNEL` key searches for all devices of the `ttyUSB` type. Using the `*` wild card, this key matches several of these devices. The second match key, `ATTRS`, checks whether the `product` attribute file in `sysfs` for any `ttyUSB` device contains a certain string. The assign key (`SYMLINK`) triggers the addition of a symbolic link to this device under `/dev/pilot`. The operator used in this key (+=) tells udev to additionally perform this action, even if previous or later rules add other symbolic links. As this rule contains two match keys, it is only applied if both conditions are met.

The `printer` rule deals with USB printers and contains two match keys which must both apply to get the entire rule applied (`SUBSYSTEM` and `KERNEL`). Three assign keys deal with the naming for this device type (`NAME`), the creation of symbolic device links (`SYMLINK`), and the group membership for this device type (`GROUP`). Using the `*` wild card in the `KERNEL` key makes it match several `lp` printer devices. Substitutions are used in both, the `NAME` and the `SYMLINK` keys to extend these strings by the internal device name. For example, the symlink to the first `lp` USB printer would read `/dev/usblp0`.

The `kernel firmware loader` rule makes udev load additional firmware by an external helper script during runtime. The `SUBSYSTEM` match key searches for the `firmware` subsystem. The `ACTION` key checks whether any device belonging to the `firmware` subsystem has been added. The `RUN+=` key triggers the execution of the `firmware.sh` script to locate the firmware that is to be loaded.

Some general characteristics are common to all rules:

- Each rule consists of one or more key value pairs separated by a comma.
- A key's operation is determined by the operator. udev rules support several different operators.
- Each given value must be enclosed by quotation marks.
- Each line of the rules file represents one rule. If a rule is longer than just one line, use `\` to join the different lines just as you would do in shell syntax.
- udev rules support a shell-style pattern that matches the `*`, `?`, and `[]` patterns.

- udev rules support substitutions.

17.6.1 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.

17.6.2 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.

17.6.3 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.

17.6.4 Using udev Assign Keys

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

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

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

`OWNER, GROUP, MODE`
: The permissions for the new device node. Values specified here overwrite anything that has been compiled in.

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

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

`RUN`
: Tell udev to add a program to the list of programs to be executed for this device. Mind to restrict this to very short tasks to avoid blocking further events for this device.

`LABEL`
: Add a label where a `GOTO` can jump to.

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

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

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

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

`OPTIONS`
: The `OPTION` key may have several possible values:

- `last_rule` tells udev to ignore all later rules.
- `ignore_device` tells udev to ignore this event completely.
- `ignore_remove` tells udev to ignore all later remove events for the device.

- `all_partitions` tells udev to create device nodes for all available partitions on a block device.

17.7 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
```

17.8 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.

17.9 For More Information

For more information about the udev infrastructure, refer to the following man pages:

udev
: General information about udev, keys, rules, and other important configuration issues.

udevadm
: udevadm can be used to control the runtime behavior of udev, request kernel events, manage the event queue, and provide simple debugging mechanisms.

udevd
: Information about the udev event managing daemon.

18 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.

18.1 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, each employing different features and characteristics. If you need further information about other shells, search for *shell* in YaST.

18.1.1 Knowing The Bash Configuration Files

A shell can be invoked:

1. as an interactive login shell. This is used when logging in to a machine, invoking Bash with the `--login` option or when logging in to a remote machine with SSH.

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

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

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

***Table 18.1** Bash Configuration Files for Login Shells*

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

***Table 18.2** Bash Configuration Files for Non-Login Shells*

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

Additionally, Bash uses some more files:

***Table 18.3** Special Files for Bash*

File	Description
~/.bash_history	contains a list of all commands you have been typing
~/.bash_logout	used when logging out

18.1.2 The Directory Structure

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

***Table 18.4** 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 an account on the system. Only `root`'s home directory is not located in `/home` but in `/root`.
`/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 `root`.
`/sbin`	Essential system binaries.

Directory	Contents
`/srv`	Data for services provided by the system.
`/tmp`	Temporary files.
`/usr`	Secondary hierarchy with read-only data.
`/var`	Variable data such as log files.
`/windows`	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 openSUSE.

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

`/dev`
: Holds device files that represent hardware components.

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

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

guration data for their desktop in `.kde` or `.kde4` respectively, GNOME users find it in `.gconf`. For information about hidden files, refer to Section “Key Features” (Chapter 7, *Basic Concepts*, ↑Start-Up).

NOTE: Home Directory in a Network Environment

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

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

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

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

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

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

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

`/srv`
: Holds data for services provided by the system, such as FTP and HTTP.

`/tmp`
: This directory is used by programs that require temporary storage of files.

`/usr`
: `/usr` has nothing to do with users, but is the acronym for UNIX system resources. The data in `/usr` is static, read-only data that can be shared among various hosts compliant to the Filesystem Hierarchy Standard (FHS). This directory contains all application programs and establishes a secondary hierarchy in the file system. KDE4 and GNOME are also located here. `/usr` holds a number of subdirectories, such as `/usr/bin`, `/usr/sbin`, `/usr/local`, and `/usr/share/doc`.

`/usr/bin`
: Contains generally accessible programs.

`/usr/sbin`
: Contains programs reserved for the system administrator, such as repair functions.

`/usr/local`
: In this directory the system administrator can install local, distribution-independent extensions.

`/usr/share/doc`
: Holds various documentation files and the release notes for your system. In the `manual` subdirectory find an online version of this manual. If more than one language is installed, this directory may contain versions of the manuals for different languages.

  Under `packages` find the documentation included in the software packages installed on your system. For every package, a subdirectory `/usr/share/doc/packages/`*`packagename`* is created that often holds README files for the package and sometimes examples, configuration files or additional scripts.

  If HOWTOs are installed on your system `/usr/share/doc` also holds the `howto` subdirectory in which to find additional documentation on many tasks 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

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

`/windows`
: Only available if you have both Microsoft Windows and Linux installed on your system. Contains the Windows data available on the Windows partition of your system. Whether you can edit the data in this directory depends on the file system your Windows partition uses. If it is FAT32, you can open and edit the files in this directory. For an NTFS file system, however, you can only read your Windows files from Linux, but not modify them. Learn more in Section “Accessing Files on Different OS on the Same Computer” (Chapter 11, *Copying and Sharing Files*, ↑Start-Up).

18.2 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 18.1 *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 respective text.

Before you can run this script you need some prerequisites:

1. Every script should contain a Shebang line (this is already the case with our example above.) If a script does not have this line, you have to call the interpreter yourself.

2. You can save the script wherever you want. However, it is a good idea to save it in a directory where the shell searches for it. The search path in a shell is determined by the environment variable `PATH`. For example, save it in the directory `~/bin/` under the name `hello.sh`.

3. The script needs executable permissions. Set the permissions with the following command:

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

If you have fullfilled all of the above prerequisites, you can execute the script with either `~/bin/hello.sh` or `hello.sh`. The first call uses an absolute path whereas the second one searches for the command in each directory given by the `PATH` environment variable.

18.3 Redirecting Command Events

Each command can use three channels, either for input or output:

- **Standard Output** This is the default output channel. Whenever a command prints something, it uses the standard output channel.

- **Standard Input** If a command needs input from users or other commands, it uses this channel.

- **Standard Error** Commands use this channel for error reporting.

To redirect these channels, there are the following possibilities:

`Command > File`
: Saves the output of the command into a file, an existing file will be deleted. For example, the `ls` command writes its output into the file `listing.txt`:

  ```
  ls > listing.txt
  ```

`Command >> File`
: Appends the output of the command to a file. For example, the `ls` command appends its output to the file `listing.txt`:

  ```
  ls >> listing.txt
  ```

`Command < File`
: Reads the file as input for the given command. For example, the `read` command reads in the content of the file into a variable:

```
read a < foo
```

`Command1 | Command2`
: Redirects the output of the left command as input for the right command.

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

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

18.4 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`.

18.5 Using Variables in Bash

A shell variable can be global or local. Global variables, or environment variables, can be accessed in all shells. In contrast, local variables are visible in the current shell only.

To view all environment variables, use the `printenv` command. If you need a special variable, insert the name of your variable as an argument:

```
printenv PATH
```

A variable can also be viewed with `echo`:

```
echo $PATH
```

This prints the `PATH` variable. To set a local variable, use a variable name followed by the equal sign, followed by the value:

```
PROJECT="SLED"
```

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

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

To remove a variable, use `unset`:

```
unset NAME
```

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

Table 18.5 *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

18.5.1 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" "" ""
```

18.5.2 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
```

18.6 Grouping And Combining Commands

Shells allow0.0 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
```

18.7 Working with Common Flow Constructs

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

18.7.1 The if Control Command

The `if` 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
```

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

18.7.2 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
```

18.8 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 V. Services

19 Basic Networking

Linux offers the necessary networking tools and features for integration into all types of network structures. The customary Linux protocol, TCP/IP, has various services and special features, which are discussed here. Network access using a network card, modem, or other device can be configured with YaST. Manual configuration is also possible. In this chapter only the fundamental mechanisms and the relevant network configuration files are covered.

Linux and other Unix operating systems use the TCP/IP protocol. It is not a single network protocol, but a family of network protocols that offer various services. The protocols listed in Table 19.1, "Several Protocols in the TCP/IP Protocol Family" (page 266) are provided for the purpose of exchanging data between two machines via TCP/IP. Networks combined by TCP/IP, comprising a worldwide network are also referred to as "the Internet."

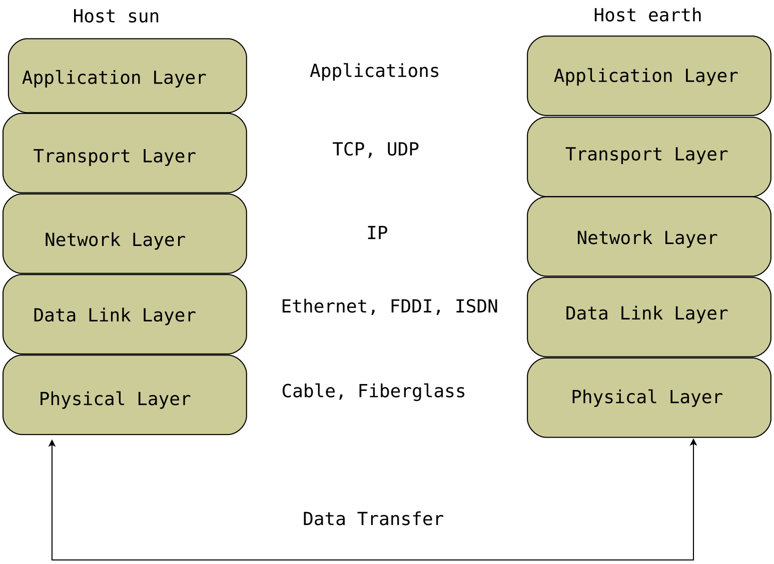
RFC stands for *Request for Comments*. RFCs are documents that describe various Internet protocols and implementation procedures for the operating system and its applications. The RFC documents describe the setup of Internet protocols. To expand your knowledge about any of the protocols, refer to the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

Table 19.1 Several Protocols in the TCP/IP Protocol Family

Protocol	Description
TCP	Transmission Control Protocol: a connection-oriented secure protocol. The data to transmit is first sent by the application as a stream of data and converted into the appropriate format by the operating system. The data arrives at the respective application on the destination host in the original data stream format it was initially sent. TCP determines whether any data has been lost during the transmission or the order of the data got mixed up. TCP is implemented wherever the data sequence matters.
UDP	User Datagram Protocol: a connectionless, insecure protocol. The data to transmit is sent in the form of packets generated by the application. The order in which the data arrives at the recipient is not guaranteed and data loss is possible. UDP is suitable for record-oriented applications. It features a smaller latency period than TCP.
ICMP	Internet Control Message Protocol: Essentially, this is not a protocol for the end user, but a special control protocol that issues error reports and can control the behavior of machines participating in TCP/IP data transfer. In addition, it provides a special echo mode that can be viewed using the program ping.
IGMP	Internet Group Management Protocol: This protocol controls machine behavior when implementing IP multicast.

As shown in Figure 19.1, "Simplified Layer Model for TCP/IP" (page 267), data exchange takes place in different layers. The actual network layer is the insecure data transfer via IP (Internet protocol). On top of IP, TCP (transmission control protocol) guarantees, to a certain extent, security of the data transfer. The IP layer is supported by the underlying hardware-dependent protocol, such as ethernet.

Figure 19.1 *Simplified Layer Model for TCP/IP*

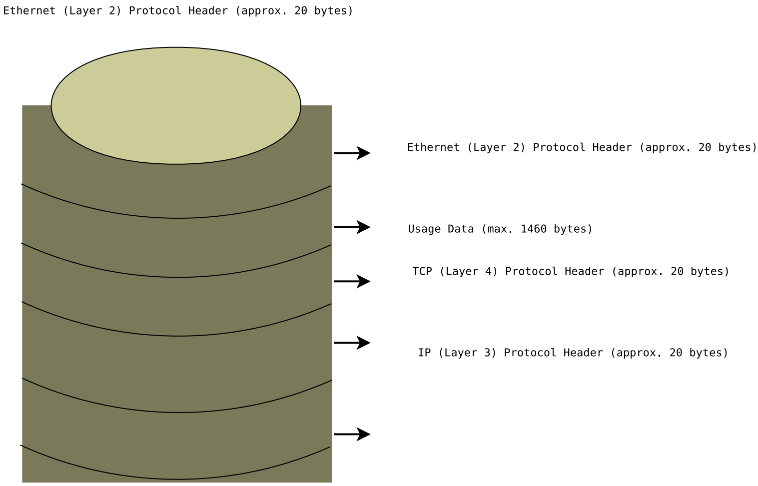


The diagram provides one or two examples for each layer. The layers are ordered according to *abstraction levels*. The lowest layer is very close to the hardware. The uppermost layer, however, is almost a complete abstraction from the hardware. Every layer has its own special function. The special functions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

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

For the layers to serve their designated functions, additional information regarding each layer must be saved in the data packet. This takes place in the *header* of the packet. Every layer attaches a small block of data, called the protocol header, to the front of each emerging packet. A sample TCP/IP data packet traveling over an ethernet cable is illustrated in Figure 19.2, "TCP/IP Ethernet Packet" (page 268). The proof sum is located at the end of the packet, not at the beginning. This simplifies things for the network hardware.

***Figure 19.2** TCP/IP Ethernet Packet*



When an application sends data over the network, the data passes through each layer, all implemented in the Linux kernel except the physical layer. Each layer is responsible for preparing the data so it can be passed to the next layer. The lowest layer is ultimately responsible for sending the data. The entire procedure is reversed when data is received. Like the layers of an onion, in each layer the protocol headers are removed from the transported data. Finally, the transport layer is responsible for making the data available for use by the applications at the destination. In this manner, one layer only communicates with the layer directly above or below it. For applications, it is irrelevant whether data is transmitted via a 100 Mbit/s FDDI network or via a 56-Kbit/s modem line. Likewise, it is irrelevant for the data line which kind of data is transmitted, as long as packets are in the correct format.

19.1 IP Addresses and Routing

The discussion in this section is limited to IPv4 networks. For information about IPv6 protocol, the successor to IPv4, refer to Section 19.2, "IPv6—The Next Generation Internet" (page 272).

19.1.1 IP Addresses

Every computer on the Internet has a unique 32-bit address. These 32 bits (or 4 bytes) are normally written as illustrated in the second row in Example 19.1, "Writing IP Addresses" (page 269).

Example 19.1 *Writing IP Addresses*

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

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

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

19.1.2 Netmasks and Routing

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

To understand how the netmask works, look at Example 19.2, "Linking IP Addresses to the Netmask" (page 270). The netmask consists of 32 bits that identify how much of an IP address belongs to the network. All those bits that are `1` mark the corresponding bit in the IP address as belonging to the network. All bits that are `0` mark bits inside the subnetwork. This means that the more bits are `1`, the smaller the subnetwork is. Because the netmask always consists of several successive `1` bits, it is also possible to just count the number of bits in the netmask. In Example 19.2, "Linking IP Addresses to the Netmask" (page 270) the first net with 24 bits could also be written as `192.168.0.0/24`.

***Example 19.2** Linking IP Addresses to the Netmask*

```
IP address (192.168.0.20):  11000000 10101000 00000000 00010100
Netmask   (255.255.255.0):  11111111 11111111 11111111 00000000
---------------------------------------------------------------
Result of the link:         11000000 10101000 00000000 00000000
In the decimal system:           192.     168.       0.       0

IP address (213.95.15.200): 11010101 10111111 00001111 11001000
Netmask    (255.255.255.0): 11111111 11111111 11111111 00000000
---------------------------------------------------------------
Result of the link:         11010101 10111111 00001111 00000000
In the decimal system:           213.      95.      15.       0
```

To give another example: all machines connected with the same ethernet cable are usually located in the same subnetwork and are directly accessible. Even when the subnet is physically divided by switches or bridges, these hosts can still be reached directly.

IP addresses outside the local subnet can only be reached if a gateway is configured for the target network. In the most common case, there is only one gateway that handles all traffic that is external. However, it is also possible to configure several gateways for different subnets.

If a gateway has been configured, all external IP packets are sent to the appropriate gateway. This gateway then attempts to forward the packets in the same manner—from host to host—until it reaches the destination host or the packet's TTL (time to live) expires.

Table 19.2 Specific Addresses

Address Type	Description
Base Network Address	This is the netmask AND any address in the network, as shown in Example 19.2, "Linking IP Addresses to the Netmask" (page 270) under `Result`. This address cannot be assigned to any hosts.
Broadcast Address	This basically says, "Access all hosts in this subnetwork." To generate this, the netmask is inverted in binary form and linked to the base network address with a logical OR. The above example therefore results in 192.168.0.255. This address cannot be assigned to any hosts.
Local Host	The address `127.0.0.1` is assigned to the "loopback device" on each host. A connection can be set up to your own machine with this address.

Because IP addresses must be unique all over the world, you cannot just select random addresses. There are three address domains to use if you want to set up a private IP-based network. These cannot get any connection from the rest of the Internet, because they cannot be transmitted over the Internet. These address domains are specified in RFC 1597 and listed in Table 19.3, "Private IP Address Domains" (page 271).

Table 19.3 Private IP Address Domains

Network/Netmask	Domain
`10.0.0.0/255.0.0.0`	`10.x.x.x`
`172.16.0.0/255.240.0.0`	`172.16.x.x – 172.31.x.x`
`192.168.0.0/255.255.0.0`	`192.168.x.x`

19.2 IPv6—The Next Generation Internet

Due to the emergence of the WWW (World Wide Web), the Internet has experienced explosive growth with an increasing number of computers communicating via TCP/IP in the past fifteen years. Since Tim Berners-Lee at CERN (`http://public.web.cern.ch`) invented the WWW in 1990, the number of Internet hosts has grown from a few thousand to about a hundred million.

As mentioned, an IPv4 address consists of only 32 bits. Also, quite a few IP addresses are lost—they cannot be used due to the way in which networks are organized. The number of addresses available in your subnet is two to the power of the number of bits, minus two. A subnetwork has, for example, 2, 6, or 14 addresses available. To connect 128 hosts to the Internet, for example, you need a subnetwork with 256 IP addresses, from which only 254 are usable, because two IP addresses are needed for the structure of the subnetwork itself: the broadcast and the base network address.

Under the current IPv4 protocol, DHCP or NAT (network address translation) are the typical mechanisms used to circumvent the potential address shortage. Combined with the convention to keep private and public address spaces separate, these methods can certainly mitigate the shortage. The problem with them lies in their configuration, which is a chore to set up and a burden to maintain. To set up a host in an IPv4 network, you need a number of address items, such as the host's own IP address, the subnetmask, the gateway address, and maybe a name server address. All these items need to be known and cannot be derived from somewhere else.

With IPv6, both the address shortage and the complicated configuration should be a thing of the past. The following sections tell more about the improvements and benefits brought by IPv6 and about the transition from the old protocol to the new one.

19.2.1 Advantages

The most important and most visible improvement brought by the new protocol is the enormous expansion of the available address space. An IPv6 address is made up of 128 bit values instead of the traditional 32 bits. This provides for as many as several quadrillion IP addresses.

However, IPv6 addresses are not only different from their predecessors with regard to their length. They also have a different internal structure that may contain more specific information about the systems and the networks to which they belong. More details about this are found in Section 19.2.2, "Address Types and Structure" (page 274).

The following is a list of some other advantages of the new protocol:

Autoconfiguration
: IPv6 makes the network "plug and play" capable, which means that a newly set up system integrates into the (local) network without any manual configuration. The new host uses its automatic configuration mechanism to derive its own address from the information made available by the neighboring routers, relying on a protocol called the *neighbor discovery* (ND) protocol. This method does not require any intervention on the administrator's part and there is no need to maintain a central server for address allocation—an additional advantage over IPv4, where automatic address allocation requires a DHCP server.

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

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

Backward Compatibility
: Realistically, it would be impossible to switch the entire Internet from IPv4 to IPv6 at one time. Therefore, it is crucial that both protocols are able to coexist not only on the Internet, but also on one system. This is ensured by compatible addresses (IPv4 addresses can easily be translated into IPv6 addresses) and through the use of a number of tunnels. See Section 19.2.3, "Coexistence of IPv4 and IPv6" (page 278). Also, systems can rely on a *dual stack IP* technique to support both protocols at the same time, meaning that they have two network stacks that are

completely separate, such that there is no interference between the two protocol versions.

Custom Tailored Services through Multicasting
: With IPv4, some services, such as SMB, need to broadcast their packets to all hosts in the local network. IPv6 allows a much more fine-grained approach by enabling servers to address hosts through *multicasting*—by addressing a number of hosts as parts of a group (which is different from addressing all hosts through *broadcasting* or each host individually through *unicasting*). Which hosts are addressed as a group may depend on the concrete application. There are some predefined groups to address all name servers (the *all name servers multicast group*), for example, or all routers (the *all routers multicast group*).

19.2.2 Address Types and Structure

As mentioned, the current IP protocol is lacking in two important aspects: there is an increasing shortage of IP addresses and configuring the network and maintaining the routing tables is becoming a more complex and burdensome task. IPv6 solves the first problem by expanding the address space to 128 bits. The second one is countered by introducing a hierarchical address structure, combined with sophisticated techniques to allocate network addresses, as well as *multihoming* (the ability to assign several addresses to one device, giving access to several networks).

When dealing with IPv6, it is useful to know about three different types of addresses:

Unicast
: Addresses of this type are associated with exactly one network interface. Packets with such an address are delivered to only one destination. Accordingly, unicast addresses are used to transfer packets to individual hosts on the local network or the Internet.

Multicast
: Addresses of this type relate to a group of network interfaces. Packets with such an address are delivered to all destinations that belong to the group. Multicast addresses are mainly used by certain network services to communicate with certain groups of hosts in a well-directed manner.

Anycast

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

An IPv6 address is made up of eight four-digit fields, each representing 16 bits, written in hexadecimal notation. They are also separated by colons (`:`). Any leading zero bytes within a given field may be dropped, but zeros within the field or at its end may not. Another convention is that more than four consecutive zero bytes may be collapsed into a double colon. However, only one such `::` is allowed per address. This kind of shorthand notation is shown in Example 19.3, “Sample IPv6 Address” (page 275), where all three lines represent the same address.

Example 19.3 *Sample IPv6 Address*

```
fe80 : 0000 : 0000 : 0000 : 0000 : 10 : 1000 : 1a4
fe80 :    0 :    0 :    0 :    0 : 10 : 1000 : 1a4
fe80 :                           : 10 : 1000 : 1a4
```

Each part of an IPv6 address has a defined function. The first bytes form the prefix and specify the type of address. The center part is the network portion of the address, but it may be unused. The end of the address forms the host part. With IPv6, the netmask is defined by indicating the length of the prefix after a slash at the end of the address. An address, as shown in Example 19.4, “IPv6 Address Specifying the Prefix Length” (page 275), contains the information that the first 64 bits form the network part of the address and the last 64 form its host part. In other words, the `64` means that the netmask is filled with 64 1-bit values from the left. Just like with IPv4, the IP address is combined with AND with the values from the netmask to determine whether the host is located in the same subnetwork or in another one.

Example 19.4 *IPv6 Address Specifying the Prefix Length*

```
fe80::10:1000:1a4/64
```

IPv6 knows about several predefined types of prefixes. Some of these are shown in Table 19.4, “Various IPv6 Prefixes” (page 276).

Table 19.4 Various IPv6 Prefixes

Prefix (hex)	Definition
`00`	IPv4 addresses and IPv4 over IPv6 compatibility addresses. These are used to maintain compatibility with IPv4. Their use still requires a router able to translate IPv6 packets into IPv4 packets. Several special addresses, such as the one for the loopback device, have this prefix as well.
`2` or `3` as the first digit	Aggregatable global unicast addresses. As is the case with IPv4, an interface can be assigned to form part of a certain subnetwork. Currently, there are the following address spaces: `2001::/16` (production quality address space) and `2002::/16` (6to4 address space).
`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 19.2.3, "Coexistence of IPv4 and IPv6" (page 278)) to allow IPv4 and IPv6 hosts to communicate with others operating in a pure IPv4 environment.

IPv4 Addresses Mapped to IPv6
: This type of address specifies a pure IPv4 address in IPv6 notation.

Local Addresses
: There are two address types for local use:

    link-local
    : This type of address can only be used in the local subnetwork. Packets with a source or target address of this type should not be routed to the Internet or other subnetworks. These addresses contain a special prefix (`fe80::/10`) and the interface ID of the network card, with the middle part consisting of zero bytes. Addresses of this type are used during automatic configuration to communicate with other hosts belonging to the same subnetwork.

site-local
: Packets with this type of address may be routed to other subnetworks, but not to the wider Internet—they must remain inside the organization's own network. Such addresses are used for intranets and are an equivalent of the private address space defined by IPv4. They contain a special prefix (`fec0::/10`), the interface ID, and a 16 bit field specifying the subnetwork ID. Again, the rest is filled with zero bytes.

As a completely new feature introduced with IPv6, each network interface normally gets several IP addresses, with the advantage that several networks can be accessed through the same interface. One of these networks can be configured completely automatically using the MAC and a known prefix with the result that all hosts on the local network can be reached as soon as IPv6 is enabled (using the link-local address). With the MAC forming part of it, any IP address used in the world is unique. The only variable parts of the address are those specifying the *site topology* and the *public topology*, depending on the actual network in which the host is currently operating.

For a host to go back and forth between different networks, it needs at least two addresses. One of them, the *home address*, not only contains the interface ID but also an identifier of the home network to which it normally belongs (and the corresponding prefix). The home address is a static address and, as such, it does not normally change. Still, all packets destined to the mobile host can be delivered to it, regardless of whether it operates in the home network or somewhere outside. This is made possible by the completely new features introduced with IPv6, such as *stateless autoconfiguration* and *neighbor discovery*. In addition to its home address, a mobile host gets one or more additional addresses that belong to the foreign networks where it is roaming. These are called *care-of* addresses. The home network has a facility that forwards any packets destined to the host when it is roaming outside. In an IPv6 environment, this task is performed by the *home agent*, which takes all packets destined to the home address and relays them through a tunnel. On the other hand, those packets destined to the care-of address are directly transferred to the mobile host without any special detours.

19.2.3 Coexistence of IPv4 and IPv6

The migration of all hosts connected to the Internet from IPv4 to IPv6 is a gradual process. Both protocols will coexist for some time to come. The coexistence on one system is guaranteed where there is a *dual stack* implementation of both protocols. That still leaves the question of how an IPv6 enabled host should communicate with an IPv4 host and how IPv6 packets should be transported by the current networks, which are predominantly IPv4 based. The best solutions offer tunneling and compatibility addresses (see Section 19.2.2, “Address Types and Structure” (page 274)).

IPv6 hosts that are more or less isolated in the (worldwide) IPv4 network can communicate through tunnels: IPv6 packets are encapsulated as IPv4 packets to move them across an IPv4 network. Such a connection between two IPv4 hosts is called a *tunnel*. To achieve this, packets must include the IPv6 destination address (or the corresponding prefix) as well as the IPv4 address of the remote host at the receiving end of the tunnel. A basic tunnel can be configured manually according to an agreement between the hosts' administrators. This is also called *static tunneling*.

However, the configuration and maintenance of static tunnels is often too labor-intensive to use them for daily communication needs. Therefore, IPv6 provides for three different methods of *dynamic tunneling*:

6over4
: IPv6 packets are automatically encapsulated as IPv4 packets and sent over an IPv4 network capable of multicasting. IPv6 is tricked into seeing the whole network (Internet) as a huge local area network (LAN). This makes it possible to determine the receiving end of the IPv4 tunnel automatically. However, this method does not scale very well and is also hampered by the fact that IP multicasting is far from widespread on the Internet. Therefore, it only provides a solution for smaller corporate or institutional networks where multicasting can be enabled. The specifications for this method are laid down in RFC 2529.

6to4
: With this method, IPv4 addresses are automatically generated from IPv6 addresses, enabling isolated IPv6 hosts to communicate over an IPv4 network. However, a number of problems have been reported regarding the communication between those isolated IPv6 hosts and the Internet. The method is described in RFC 3056.

IPv6 Tunnel Broker
: This method relies on special servers that provide dedicated tunnels for IPv6 hosts. It is described in RFC 3053.

19.2.4 Configuring IPv6

To configure IPv6, you normally do not need to make any changes on the individual workstations. IPv6 is enabled by default. You can disable it during installation in the network configuration step described in Section “Network Configuration” (Chapter 1, *Installation with YaST*, ↑Start-Up). To disable or enable IPv6 on an installed system, use the YaST *Network Settings* module. On the *Global Options* tab, check or uncheck the *Enable IPv6* option as necessary. To enable IPv6 manually, enter `modprobe ipv6` as `root`.

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

19.2.5 For More Information

The above overview does not cover the topic of IPv6 comprehensively. For a more in-depth look at the new protocol, refer to the following online documentation and books:

`http://www.ipv6.org/`
: The starting point for everything about IPv6.

`http://www.ipv6day.org`
: All information needed to start your own IPv6 network.

`http://www.ipv6-to-standard.org/`
: The list of IPv6-enabled products.

`http://www.bieringer.de/linux/IPv6/`
: Here, find the Linux IPv6-HOWTO and many links related to the topic.

RFC 2640
: The fundamental RFC about IPv6.

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

19.3 Name Resolution

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

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

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

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

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

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

For your machine to resolve an IP address, it must know about at least one name server and its IP address. Easily specify such a name server with the help of YaST. If you have a modem dial-up connection, you may not need to configure a name server manually at all. The dial-up protocol provides the name server address as the connection is made. The configuration of name server access with openSUSE® is described in Section "Configuring Hostname and DNS" (page 291). Setting up your own name server is described in Chapter 21, *The Domain Name System* (page 325).

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 send 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`. Also read the `host.conf` manual page.

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

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

19.4 Configuring a Network Connection with YaST

There are many supported networking types on Linux. Most of them use different device names and the configuration files are spread over several locations in the file system. For a detailed overview of the aspects of manual network configuration, see Section 19.6, "Configuring a Network Connection Manually" (page 303).

During installation on a laptop, where NetworkManager is active by default, YaST configures all interfaces that have been detected. If NetworkManager is not active, only the first interface with link up (with a network cable connected) is automatically configured. Additional hardware can be configured any time on the installed system. The following sections describe the network configuration for all types of network connections supported by openSUSE.

19.4.1 Configuring the Network Card with YaST

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

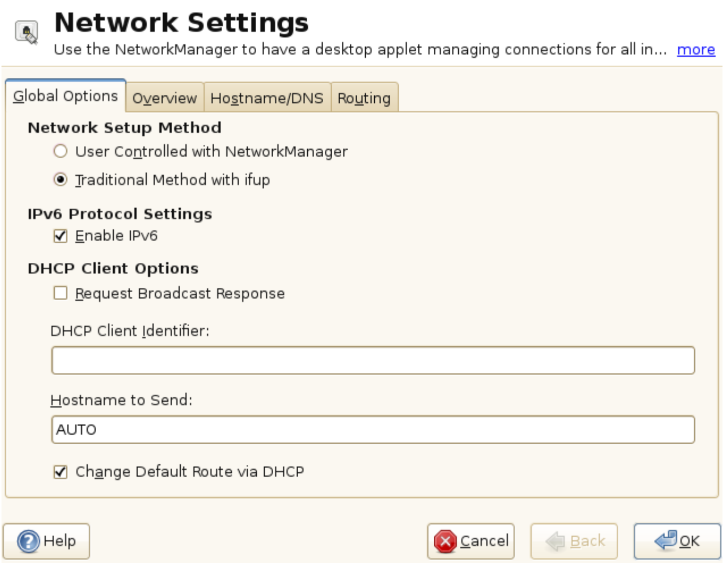
The *Global Options* tab allows to set general networking options such as the use of NetworkManager, IPv6 and general DHCP options. For more information, see Section "Configuring Global Networking Options" (page 284).

The *Overview* tab contains information about installed network interfaces and configurations. Any properly detected network card is listed with its name. You can manually configure new cards, remove or change their configuration in this dialog. If you want to manually configure a card that was not automatically detected, see Section "Configuring an Undetected Network Card" (page 290). If you want to change the configuration of an already configured card, see Section "Changing the Configuration of a Network Card" (page 285).

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 291).

The *Routing* tab is used for the configuration of routing. See Section "Configuring Routing" (page 293) for more information.

Figure 19.3 Configuring Network Settings



Configuring Global Networking Options

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

In the *Network Setup Method* choose the way network connections are managed. If you want a NetworkManager desktop applet to manage connections for all interfaces, choose *User Controlled with NetworkManager*. This option is well suited for switching between multiple wired and wireless networks. If you do not run a desktop environment (GNOME or KDE), or if your computer is a Xen server, virtual system, or provides network services such as DHCP or DNS in your network, use the *Traditional Method with ifup*. If NetworkManager is used, `nm-applet` should be used to configure network options and the *Overview*, *Hostname/DNS*, and *Routing* tabs of the *Network Settings* module are disabled. For more information on NetworkManager, see Chapter 10, *Using NetworkManager* (↑Start-Up).

In the *IPv6 Protocol Settings* choose whether you want to use the IPv6 protocol. It is possible to use IPv6 together with IPv4. By default, IPv6 is activated. However, in networks not using IPv6 protocol, response times can be faster with IPv6 protocol disabled. If you want to disable IPv6, uncheck the *Enable IPv6* option. This disables autoload of the kernel module for IPv6. This will be applied after reboot.

In the *DHCP Client Options* configure options for the DHCP client. 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 yo 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 28.1.2, “Configuration with YaST” (page 447).

Configuring IP Addresses

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

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

If possible, the first network card with link that is available during the installation is automatically configured to use automatic address setup via DHCP. In case of laptop computers where NetworkManager is active by default, all network cards are configured.

DHCP should also be used if you are using a DSL line but with no static IP assigned by the ISP (Internet Service Provider). If you decide to use DHCP, configure the details in *DHCP Client Options* in the *Global Options* tab of the *Network Settings* dialog of the YaST network card configuration module. 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 291). To configure a gateway, proceed as described in Section "Configuring Routing" (page 293).

Configuring Aliases

One network device can have multiple IP addresses, called aliases. To set an alias for your network card, proceed as follows:

1 Select a card from the list of detected cards in the *Overview* tab of the YaST network card configuration module and click *Edit*.

2 In the *Address* > *Additional Addresses* tab, click *Add*.

3 Enter *Alias Name*, *IP Address*, and *Netmask*. Do not include the interface name in the alias name.

4 Click *OK*.

5 Click *Next*.

6 To activate the configuration, click *OK*.

Changing the Device Name and Udev Rules

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

1 Select a card from the list of detected cards in the *Overview* tab of the YaST *Network Settings* module and click *Edit*.

2 Go to the *Hardware* tab. The current device name is shown in *Udev Rules*. Click *Change*.

3 Select whether udev should identify the card by its *MAC Address* or *Bus ID*. The current MAC address and bus ID of the card are shown in the dialog.

4 To change the device name, check the *Change Device Name* option and edit the name.

5 Click *OK* and *Next*.

6 To activate the configuration, click *OK*.

Changing Network Card Kernel Driver

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

1. Select a card from the list of detected cards in the *Overview* tab of the YaST Network Settings module and click *Edit*.

2. Go to the *Hardware* tab.

3. Select the kernel driver to be used in *Module Name*. Enter any options for the selected driver in *Options* in the form `option=value` . If more options are used, they should be space-separated.

4. Click *OK* and *Next*.

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

Activating the Network Device

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

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

2. In the *General* tab, select the desired entry from *Device Activation*.

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

3. Click *Next*.

4. To activate the configuration, click *OK*.

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

Setting Up Maximum Transfer Unit Size

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

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

2. In the *General* tab, select the desired entry from the *Set MTU* list.

3. Click *Next*.

4. To activate the configuration, click *OK*.

Configuring the Firewall

Without having to enter the detailed firewall setup as described in Section “Configuring the Firewall with YaST” (Chapter 14, *Masquerading and Firewalls*, ↑Security Guide), you can determine the basic firewall setup for your device as part of the device setup. Proceed as follows:

1. Open the YaST *Network Devices > Network Settings* module. In the *Overview* tab, select a card from the list of detected cards and click *Edit*.

2. Enter the *General* tab of the *Network Settings* dialog.

3. Determine the firewall zone to which your interface should be assigned. The following options are available:

Firewall Disabled
: This option is available only if the firewall is disabled and the firewall does not run at all. Only use this option, if your machine is part of a greater network that is protected by an outer firewall.

Automatically Assign Zone
: This option is available only if the firewall is enabled. The firewall is running and the interface is automatically assigned to a firewall zone. The zone which contains the keyword `any` or the external zone will be used for such an interface.

Internal Zone (Unprotected)
: The firewall is running, but does not enforce any rules to protect this interface. Use this option, if your machine is part of a greater network that is protected by an outer firewall. It is also useful for the interfaces connected to the internal network, when the machine has more network interfaces.

Demilitarized Zone
: A demilitarized zone is an additional line of defense in front of an internal network and the (hostile) Internet. Hosts assigned to this zone can be reached from the internal network and from the Internet, but cannot access the internal network.

External Zone
: The firewall is running on this interface and fully protects it against other—presumably hostile— network traffic. This is the default option.

4. Click *Next*.

5. Activate the configuration by clicking *OK*.

Configuring an Undetected Network Card

Your card may not be detected correctly. In this case, the card is not included in the list of detected cards. If you are sure that your system includes a driver for your card, you can configure it manually. You can also configure special network device types, such as bridge, bond, TUN, or TAP. To configure an undetected network card, or a special device proceed as follows:

1 In the *Network Devices* > *Network Settings* > *Overview* dialog in YaST click *Add*.

2 In the *Hardware* dialog, set the *Device Type* of the interface from the available options and *Configuration Name*. If the network card is a PCMCIA or USB device, activate the respective check box and exit this dialog with *Next*. Otherwise, you can define the kernel *Module Name* to be used for the card and its *Options*, if necessary.

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 285).

5 If you selected *Wireless* as the device type of the interface, configure the wireless connection in the next dialog.

6 Click *Next*.

7 To activate the new network configuration, click *OK*.

Configuring Hostname and DNS

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

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

1 Go to the *Network Settings* > *Hostname/DNS* tab in the *Network Devices* module in YaST.

2 Enter the *Hostname* and, if needed, the *Domain Name*. The domain is especially important if the machine is a mail server. Note that the hostname is global and applies to all set network interfaces.

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

If you are using DHCP to get an IP address, your hostname will be written to `/etc/hosts` by default and be resolvable as a `127.0.0.2` IP address. To disable this uncheck *Write Hostname to /etc/hosts* but note, that your hostname will not be resolvable without an active network.

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

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

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

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

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

`STATIC_FALLBACK`
: The static settings are used only when no dynamic configuration is avalaible.

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

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

5 To activate the configuration, click *OK*.

Configuring Routing

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

1 In YaST go to *Network Settings* > *Routing*.

2 Enter the IP address of the *Default Gateway*. The default gateway matches every possible destination, but if any other entry exists that matches the required address, use this instead of the default route.

3 More entries can be entered in the *Routing Table*. Enter the *Destination* network IP address, *Gateway* IP address and the *Netmask*. Select the *Device* through which the traffic to the defined network will be routed (the minus sign stands for any device). To omit any of these values, use the minus sign `-`. To enter a default gateway into the table, use `default` in the *Destination* field.

NOTE

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

---

4 If the system is a router, enable the *IP Forwarding* option in the *Network Settings*.

5 To activate the configuration, click *OK*.

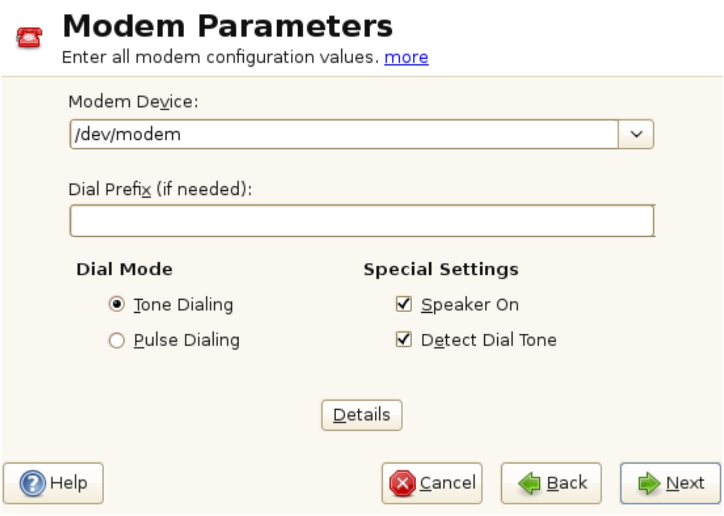
19.4.2 Modem

In the YaST Control Center, access the modem configuration under *Network Devices* > *Modem*. If your modem was not automatically detected, go to the *Modem Devices* tab and open the dialog for manual configuration by clicking *Add*. Enter the interface to which the modem is connected under *Modem Device*.

TIP: CDMA and GPRS Modems

Configure supported CDMA and GPRS modems with the YaST *Modem* module just as you would configure regular modems.

Figure 19.4 *Modem Configuration*



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.

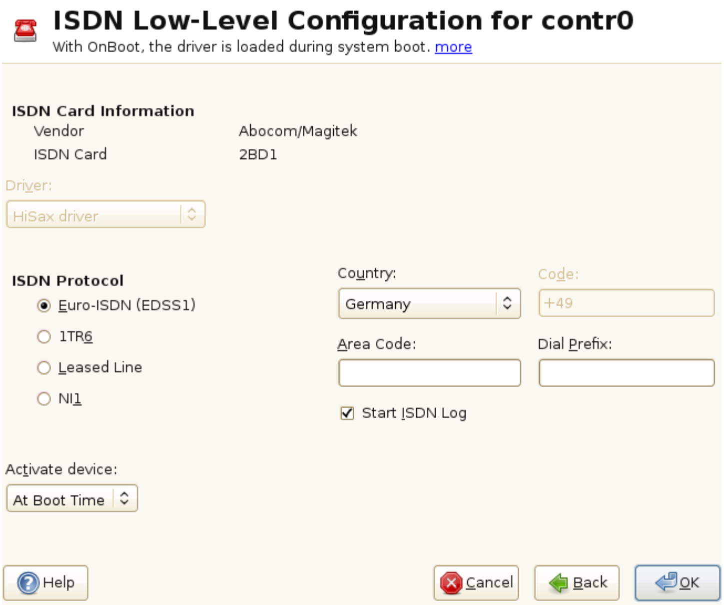
IP Details
: This opens the address configuration dialog. If your ISP does not assign a dynamic IP address to your host, disable *Dynamic IP Address* then enter your host's local IP address and the remote IP address. Ask your ISP for this information. Leave *Default Route* enabled and close the dialog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

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

19.4.3 ISDN

Use this module to configure one or several ISDN cards for your system. If YaST did not detect your ISDN card, click on *Add* in the *ISDN Devices* tab and manually select your card. Multiple interfaces are possible, but several ISPs can be configured for one interface. In the subsequent dialogs, set the ISDN options necessary for the proper functioning of the card.

Figure 19.5 ISDN Configuration

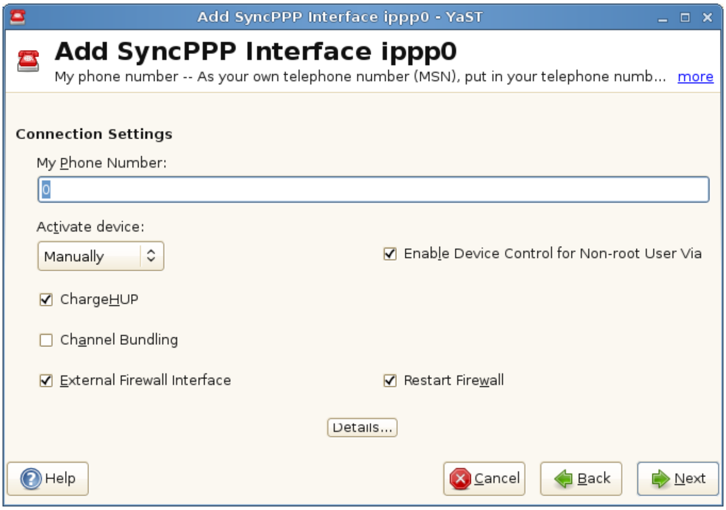


In the next dialog, shown in Figure 19.5, "ISDN Configuration" (page 297), select the protocol to use. The default is *Euro-ISDN (EDSS1)*, but for older or larger exchanges, select *1TR6*. If you are in the US, select *NI1*. Select your country in the relevant field. The corresponding country code then appears in the field next to it. Finally, provide your *Area Code* and the *Dial Prefix* if necessary. If you do not want to log all your ISDN traffic, uncheck the *Start ISDN Log* option.

Activate Device defines how the ISDN interface should be started: *At Boot Time* causes the ISDN driver to be initialized each time the system boots. *Manually* requires you to load the ISDN driver as `root` with the command `rcisdn start`. *On Hotplug*, used for PCMCIA or USB devices, loads the driver after the device is plugged in. When finished with these settings, select *OK*.

In the next dialog, specify the interface type for your ISDN card and add ISPs to an existing interface. Interfaces may be either the `SyncPPP` or the `RawIP` type, but most ISPs operate in the `SyncPPP` mode, which is described below.

Figure 19.6 ISDN Interface Configuration



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

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

ISDN Card Connected to a Private Branch Exchange
: Again, the configuration may vary depending on the equipment installed:

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

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

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

dial number. For the configuration under Linux, it should be sufficient to enter the last digit of the EAZ. As a last resort, try each of the digits from 1 to 9.

For the connection to be terminated just before the next charge unit is due, enable *ChargeHUP*. However, remember that may not work with every ISP. You can also enable channel bundling (multilink PPP) by selecting the corresponding option. Finally, you can enable firewall for your link by selecting *External Firewall Interface* and *Restart Firewall*. To enable the normal user without administrator permissions to activate or deactivate the interface, select the *Enable Device Control for Non-root User via KInternet*.

Details opens a dialog in which to implement more complex connection schemes, which are not relevant for normal home users. Leave the *Details* dialog by selecting *OK*.

In the next dialog, make IP address settings. If you have not been given a static IP by your provider, select *Dynamic IP Address*. Otherwise, use the fields provided to enter your host's local IP address and the remote IP address according to the specifications of your ISP. If the interface should be the default route to the Internet, select *Default Route*. Each host can only have one interface configured as the default route. Leave this dialog by selecting *Next*.

The following dialog allows you to set your country and select an ISP. The ISPs included in the list are call-by-call providers only. If your ISP is not in the list, select *New*. This opens the *Provider Parameters* dialog in which to enter all the details for your ISP. When entering the phone number, do not include any blanks or commas among the digits. Finally, enter your login and the password as provided by the ISP. When finished, select *Next*.

To use *Dial on Demand* on a stand-alone workstation, also specify the name server (DNS server). Most ISPs support dynamic DNS, which means the IP address of a name server is sent by the ISP each time you connect. For a single workstation, however, you still need to provide a placeholder address like `192.168.22.99`. If your ISP does not support dynamic DNS, specify the name server IP addresses of the ISP. If desired, specify a time-out for the connection—the period of network inactivity (in seconds) after which the connection should be automatically terminated. Confirm your settings with *Next*. YaST displays a summary of the configured interfaces. To activate these settings, select *OK*.

19.4.4 Cable Modem

In some countries it is quite common to access the Internet through the TV cable network. The TV cable subscriber usually gets a modem that is connected to the TV cable outlet on one side and to a computer network card on the other (using a 10Base-TG twisted pair cable). The cable modem then provides a dedicated Internet connection with a fixed IP address.

Depending on the instructions provided by your ISP, when configuring the network card either select *Dynamic Address* or *Statically Assigned IP Address*. Most providers today use DHCP. A static IP address often comes as part of a special business account.

For further information about the configuration of cable modems, read the Support Database article on the topic, which is available online at `http://en.opensuse.org/SDB:Setting_Up_an_Internet_Connection_via_Cable_Modem_with_SuSE_Linux_8.0_or_Higher`.

19.4.5 DSL

To configure your DSL device, select the *DSL* module from the YaST *Network Devices* section. This YaST module consists of several dialogs in which to set the parameters of DSL links based on one of the following protocols:

- PPP over Ethernet (PPPoE)
- PPP over ATM (PPPoATM)
- CAPI for ADSL (Fritz Cards)
- Point-to-Point Tunneling Protocol (PPTP)—Austria

In the *DSL Devices* tab of the *DSL Configuration Overview* dialog, you will find a list of installed DSL devices. To change the configuration of a DSL device, select it in the list and click *Edit*. If you click *Add*, you can manually configure a new DSL device.

The configuration of a DSL connection based on PPPoE or PPTP requires that the corresponding network card has already been set up in the correct way. If you have not done so yet, first configure the card by selecting *Configure Network Cards* (see Section 19.4.1, “Configuring the Network Card with YaST” (page 283)). In the case of a

DSL link, addresses may be assigned automatically but not via DHCP, which is why you should not enable the option *Dynamic Address*. Instead, enter a static dummy address for the interface, such as `192.168.22.1`. In *Subnet Mask*, enter `255.255.255.0`. If you are configuring a stand-alone workstation, leave *Default Gateway* empty.

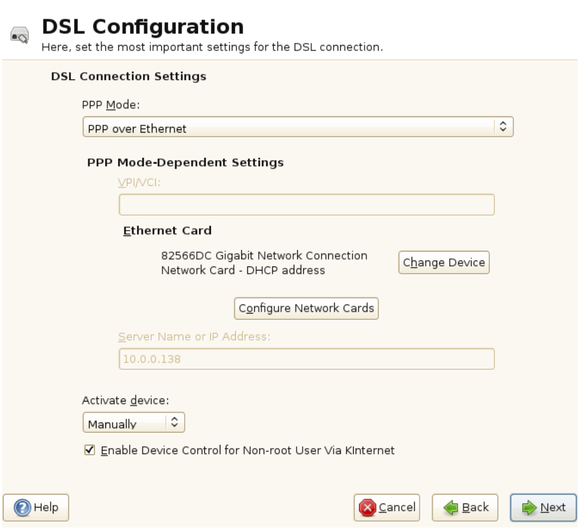
TIP

Values in *IP Address* and *Subnet Mask* are only placeholders. They are only needed to initialize the network card and do not represent the DSL link as such.

In the first DSL configuration dialog (see Figure 19.7, "DSL Configuration" (page 301)), select the *PPP Mode* and the *Ethernet Card* to which the DSL modem is connected (in most cases, this is `eth0`). Then use *Activate Device* to specify whether the DSL link should be established during the boot process. Click *Enable Device Control for Non-root User via KInternet* to authorize the normal user without root permissions to activate or deactivate the interface with KInternet.

In the next dialog select your country and choose from a number of ISPs operating in it. The details of any subsequent dialogs of the DSL configuration depend on the options set so far, which is why they are only briefly mentioned in the following paragraphs. For details on the available options, read the detailed help available from the dialogs.

***Figure 19.7** DSL Configuration*



To use *Dial on Demand* on a stand-alone workstation, also specify the name server (DNS server). Most ISPs support dynamic DNS—the IP address of a name server is sent by the ISP each time you connect. For a single workstation, however, provide a placeholder address like `192.168.22.99`. If your ISP does not support dynamic DNS, enter the name server IP address provided by your ISP.

Idle Time-Out (seconds) defines a period of network inactivity after which to terminate the connection automatically. A reasonable time-out value is between 60 and 300 seconds. If *Dial on Demand* is disabled, it may be useful to set the time-out to zero to prevent automatic hang-up.

The configuration of T-DSL is very similar to the DSL setup. Just select *T-Online* as your provider and YaST opens the T-DSL configuration dialog. In this dialog, provide some additional information required for T-DSL—the line ID, the T-Online number, the user code, and your password. All of these should be included in the information you received after subscribing to T-DSL.

19.5 NetworkManager

NetworkManager is the ideal solution for a mobile workstation. With NetworkManager, you do not need to worry about configuring network interfaces and switching between networks when you are moving. NetworkManager can automatically connect to known WLAN networks. If you have two or more connection possibilities, it can connect to the faster one.

However, NetworkManager is not a suitable solution for all cases, so you can still choose between the traditional method for managing network connections (ifup) and NetworkManager. If you want to manage your network connection with NetworkManager, enable NetworkManager in the YaST Network Settings module as described in Section "Enabling NetworkManager" (Chapter 10, *Using NetworkManager*, ↑Start-Up) 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 10, *Using NetworkManager* (↑Start-Up).

Some differences between ifup and NetworkManager include:

`root` Privileges
: If you use NetworkManager for network setup, you can easily switch, stop, or start your network connection at any time from within your desktop environment using an applet. NetworkManager also makes it possible to change and configure wireless card connections without requiring `root` privileges. For this reason, NetworkManager is the ideal solution for a mobile workstation.

  Traditional configuration with ifup also provides some ways to switch, stop, or start the connection with or without user intervention, like user-managed devices, but it always requires `root` privileges to change or configure a network device. This is often a problem for mobile computing, where it is not possible to preconfigure all connection possibilities.

Types of Network Connections
: Both, traditional configuration and NetworkManager can handle network connections with a wireless network (with WEP, WPA-PSK, and WPA-Enterprise access), dial-up, and wired networks using DHCP and static configuration. They also support connection through VPN.

  NetworkManager tries to keep your computer connected at all times using the best connection available. If the network cable is accidentally disconnected, it tries to reconnect. It can find the network with the best signal strength from the list of your wireless connections and automatically use it to connect. To get the same functionality with ifup, a great deal of configuration effort is required.

19.6 Configuring a Network Connection Manually

Manual configuration of the network software should always be the last alternative. Using YaST is recommended. However, this background information about the network configuration can also assist your work with YaST.

When the kernel detects a network card and creates a corresponding network interface, it assigns the device a name depending on the order of device discovery, or order of the loading of the kernel modules. The default kernel device names are only predictable in very simple or tightly controlled hardware environments. Systems which allow adding or removing hardware during runtime, or support automatic configuration of devices cannot expect stable network device names assigned by the kernel across reboots.

However, all system configuration tools rely on persistent interface names. The problem is solved by udev. The udev persistent net generator (`/etc/udev/rules.d/75-persistent-net-generator.rules`) generates a rule matching the hardware (using its hardware address by default) and assigns a persistently unique interface for the hardware. The udev database of network interfaces is stored in the file `/etc/udev/rules.d/70-persistent-net.rules`. Every line in the file describes one network interface and specifies its persistent name. System administrators can change the assigned names by editing the `NAME=""` entries. The persistent rules can also be modified using YaST.

Table 19.5, "Manual Network Configuration Scripts" (page 304) summarizes the most important scripts involved in the network configuration.

***Table 19.5** Manual Network Configuration Scripts*

Command	Function
`if{up,down,status}`	The `if*` scripts start, stop network interfaces, or return the status of the specified interface. More information is available in the manual page of `ifup`.
`rcnetwork`	The `rcnetwork` script can be used to start, stop, or restart all network interfaces or just a specified one. Use `rcnetwork stop` to stop, `rcnetwork start` to start, and `rcnetwork restart` 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 `rcnetwork restart eth0`. The `rcnetwork status` command displays the state of the interfaces, their IP addresses, and whether a DHCP client is running. With `rcnetwork stop-all-dhcp-clients` and `rcnetwork restart-all-dhcp-clients` you can stop or restart DHCP clients running on network interfaces.

More information about udev and persistent device names is available in Chapter 17, *Dynamic Kernel Device Management with udev* (page 235).

19.6.1 Configuration Files

This section provides an overview of the network configuration files and explains their purpose and the format used.

/etc/sysconfig/network/ifcfg-*

These files contain the configurations for network interfaces. They include information such as the start mode and the IP address. Possible parameters are described in the manual page of `ifup`. Additionally, all variables from the files `dhcp`, `wireless`, and `config` can be used in the `ifcfg-*` files if a general setting should be used for only one interface.

/etc/sysconfig/network/{config, dhcp, wireless}

The file `config` contains general settings for the behavior of `ifup`, `ifdown`, and `ifstatus`. `dhcp` contains settings for DHCP and `wireless` for wireless LAN cards. The variables in all three configuration files are commented. Some of the variables from `/etc/sysconfig/network/config` can also be used in `ifcfg-*` files, where they are treated with higher priority. The `/etc/sysconfig/network/ifcfg.template` file lists variables that can be specified in a per interface scope. However, most of the `/etc/sysconfig/network/config` variables are global and cannot be overriden in ifcfg-files. For example `NETWORKMANAGER` or `NETCONFIG_*` variables are global.

/etc/sysconfig/network/{routes,ifroute-*}

The static routing of TCP/IP packets is determined here. All the static routes required by the various system tasks can be entered in the `/etc/sysconfig/network/routes` file: routes to a host, routes to a host via a gateway, and routes to a network. For each interface that needs individual routing, define an additional configuration file: `/etc/sysconfig/network/ifroute-*`. Replace `*` with the name of the interface. The entries in the routing configuration files look like this:

```
# Destination      Dummy/Gateway     Netmask            Device
#
127.0.0.0          0.0.0.0           255.255.255.0      lo
204.127.235.0      0.0.0.0           255.255.255.0      eth0
default            204.127.235.41    0.0.0.0            eth0
207.68.156.51      207.68.145.45     255.255.255.255    eth1
192.168.0.0        207.68.156.51     255.255.0.0        eth1
```

The route's destination is in the first column. This column may contain the IP address of a network or host or, in the case of *reachable* name servers, the fully qualified network or hostname.

The second column contains the default gateway or a gateway through which a host or network can be accessed. The third column contains the netmask for networks or hosts behind a gateway. For example, the mask is `255.255.255.255` for a host behind a gateway.

The fourth column is only relevant for networks connected to the local host such as loopback, Ethernet, ISDN, PPP, and dummy device. The device name must be entered here.

An (optional) fifth column can be used to specify the type of a route. Columns that are not needed should contain a minus sign `-` to ensure that the parser correctly interprets the command. For details, refer to the `routes(5)` man page.

/etc/resolv.conf

The domain to which the host belongs is specified in this file (keyword `search`). Also listed is the status of the name server address to access (keyword `nameserver`). Multiple domain names can be specified in the file. When resolving a name that is not fully qualified, an attempt is made to generate one by attaching the individual `search` entries. Multiple name servers can be specified in multiple lines, each beginning with `nameserver`. Comments are preceded with # signs. Example 19.5, "/etc/resolv.conf" (page 307) shows what `/etc/resolv.conf` could look like.

However, the `/etc/resolv.conf` should not be edited by hand. Instead, it is generated by the `netconfig` script. To define static DNS configuration without using YaST, edit the appropriate variables manually in the `/etc/sysconfig/network/config` file: `NETCONFIG_DNS_STATIC_SEARCHLIST` (list of DNS domain names used for hostname lookup), `NETCONFIG_DNS_STATIC_SERVERS` (list of name server IP addresses to use for hostname lookup), `NETCONFIG_DNS_FORWARDER` (defines the name of the DNS forwarder that has to be configured). To disable DNS configuration using netconfig, set `NETCONFIG_DNS_POLICY=''`. For more information about `netconfig`, see `man 8 netconfig`.

Example 19.5 */etc/resolv.conf*

```
# Our domain
search example.com
#
# We use dns.example.com (192.168.1.116) as nameserver
nameserver 192.168.1.116
```

/sbin/netconfig

`netconfig` is a modular tool to manage additional network configuration settings. It merges statically defined settings with settings provided by autoconfiguration mechanisms as dhcp or ppp according to a predefined policy. The required changes are applied to the system by calling the netconfig modules that are responsible for modifying a configuration file and restarting a service or a similar action.

`netconfig` recognizes three main actions. The `netconfig modify` and `netconfig remove` commands are used by daemons such as dhcp or ppp to provide or remove settings to netconfig. Only the `netconfig update` command is available for the user:

`modify`
: The `netconfig modify` command modifies the current interface and service specific dynamic settings and updates the network configuration. Netconfig reads settings from standard input or from a file specified with the `--lease-file` *`filename`* option and internally stores them until a system reboot or the next modify or remove action. Already existing settings for the same interface and service combination are overwritten. The interface is specified by the `-i` *`interface_name`* parameter. The service is specified by the `-s` *`service_name`* parameter.

`remove`
: The `netconfig remove` command removes the dynamic settings provided by a modificatory action for the specified interface and service combination and updates the network configuration. The interface is specified by the `-i` *`interface_name`* parameter. The service is specified by the `-s` *`service_name`* parameter.

`update`
: The `netconfig update` command updates the network configuration using current settings. This is useful when the policy or the static configuration changed.

The netconfig policy and the static configuration settings are defined either manually or using YaST in the `/etc/sysconfig/network/config` file. The dynamic configuration settings provided by autoconfiguration tools as dhcp or ppp are delivered directly by these tools with the `netconfig modify` and `netconfig remove` actions. NetworkManager also uses `netconfig modify` and `netconfig remove` actions. When NetworkManager is enabled, netconfig (in policy mode `auto`) uses only NetworkManager settings, ignoring settings from any other interfaces configured using the traditional ifup method. If NetworkManager does not provide any setting, static settings are used as a fallback. A mixed usage of NetworkManager and the traditional ifup method is not supported.

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

/etc/hosts

In this file, shown in Example 19.6, "/etc/hosts" (page 308), IP addresses are assigned to hostnames. If no name server is implemented, all hosts to which an IP connection will be set up must be listed here. For each host, enter a line consisting of the IP address, the fully qualified hostname, and the hostname into the file. The IP address must be at the beginning of the line and the entries separated by blanks and tabs. Comments are always preceded by the # sign.

Example 19.6 */etc/hosts*

```
127.0.0.1 localhost
192.168.2.100 jupiter.example.com jupiter
192.168.2.101 venus.example.com venus
```

/etc/networks

Here, network names are converted to network addresses. The format is similar to that of the `hosts` file, except the network names precede the addresses. See Example 19.7, "/etc/networks" (page 309).

***Example 19.7** /etc/networks*

```
loopback      127.0.0.0
localnet      192.168.0.0
```

/etc/host.conf

Name resolution—the translation of host and network names via the *resolver* library—is controlled by this file. This file is only used for programs linked to libc4 or libc5. For current glibc programs, refer to the settings in `/etc/nsswitch.conf`. A parameter must always stand alone in its own line. Comments are preceded by a # sign. Table 19.6, "Parameters for /etc/host.conf" (page 309) shows the parameters available. A sample `/etc/host.conf` is shown in Example 19.8, "/etc/host.conf" (page 310).

***Table 19.6** Parameters for /etc/host.conf*

order *hosts*, *bind*	Specifies in which order the services are accessed for the name resolution. Available arguments are (separated by blank spaces or commas): *hosts*: searches the `/etc/hosts` file *bind*: accesses a name server *nis*: uses NIS
multi *on*/*off*	Defines if a host entered in `/etc/hosts` can have multiple IP addresses.
nospoof *on* spoofalert *on*/*off*	These parameters influence the name server *spoofing* but do not exert any influence on the network configuration.

trim *domainname*	The specified domain name is separated from the hostname after hostname resolution (as long as the hostname includes the domain name). This option is useful if only names from the local domain are in the `/etc/hosts` file, but should still be recognized with the attached domain names.

Example 19.8 */etc/host.conf*

```
# We have named running
order hosts bind
# Allow multiple address
multi on
```

/etc/nsswitch.conf

The introduction of the GNU C Library 2.0 was accompanied by the introduction of the *Name Service Switch* (NSS). Refer to the `nsswitch.conf(5)` man page and *The GNU C Library Reference Manual* for details.

The order for queries is defined in the file `/etc/nsswitch.conf`. A sample `nsswitch.conf` is shown in Example 19.9, "/etc/nsswitch.conf" (page 310). Comments are introduced by # signs. In this example, the entry under the `hosts` database means that a request is sent to `/etc/hosts` (`files`) via DNS.

Example 19.9 */etc/nsswitch.conf*

```
passwd:     compat
group:      compat

hosts:      files dns
networks:   files dns

services:   db files
protocols:  db files

netgroup:   files
automount:  files nis
```

The "databases" available over NSS are listed in Table 19.7, "Databases Available via /etc/nsswitch.conf" (page 311). In addition, `automount`, `bootparams`, `netmasks`, and `publickey` are expected in the near future. The configuration options for NSS databases are listed in Table 19.8, "Configuration Options for NSS Databases" (page 311).

Table 19.7 *Databases Available via /etc/nsswitch.conf*

`aliases`	Mail aliases implemented by `sendmail`; see `man 5 aliases`.
`ethers`	Ethernet addresses.
`group`	For user groups used by `getgrent`. See also the man page for `group`.
`hosts`	For hostnames and IP addresses, used by `gethostbyname` and similar functions.
`netgroup`	Valid host and user lists in the network for the purpose of controlling access permissions; see the `netgroup(5)` man page.
`networks`	Network names and addresses, used by `getnetent`.
`passwd`	User passwords, used by `getpwent`; see the `passwd(5)` man page.
`protocols`	Network protocols, used by `getprotoent`; see the `protocols(5)` man page.
`rpc`	Remote procedure call names and addresses, used by `getrpcbyname` and similar functions.
`services`	Network services, used by `getservent`.
`shadow`	Shadow passwords of users, used by `getspnam`; see the `shadow(5)` man page.

Table 19.8 *Configuration Options for NSS "Databases"*

`files`	directly access files, for example, `/etc/aliases`
`db`	access via a database

`nis`, `nisplus`	NIS, see also Chapter 3, *Using NIS* (↑Security Guide)
`dns`	can only be used as an extension for `hosts` and `networks`
`compat`	can only be used as an extension for `passwd`, `shadow`, and `group`

/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 may only contain one line in which the hostname is set.

19.6.2 Testing the Configuration

Before you write your configuration to the configuration files, you can test it. To set up a test configuration, use the `ip` command. To test the connection, use the `ping` command. Older configuration tools, `ifconfig` and `route`, are also available.

The commands `ip`, `ifconfig`, and `route` change the network configuration directly without saving it in the configuration file. Unless you enter your configuration in the correct configuration files, the changed network configuration is lost on reboot.

Configuring a Network Interface with ip

`ip` is a tool to show and configure routing, network devices, policy routing, and tunnels. It was designed as a replacement for the older tools ifconfig and route.

`ip` is a very complex tool. Its common syntax is `ip` *`options object command`*. You can work with the following objects:

link
: This object represents a network device.

address
: This object represents the IP address of device.

neighbour
: This object represents a ARP or NDISC cache entry.

route
: This object represents the routing table entry.

rule
: This object represents a rule in the routing policy database.

maddress
: This object represents a multicast address.

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 19.10, "Output of the Command ping" (page 315), you can see an example of the `ping` output. The second-to-last line contains information about number of transmitted packets, packet loss, and total time of `ping` running.

As the destination, you can use a hostname or IP address, for example, `ping example.com` or `ping 192.168.3.100`. The program sends packets until you press Ctrl + C.

If you only need to check the functionality of the connection, you can limit the number of the packets with the `-c` option. For example to limit ping to three packets, enter `ping -c 3 example.com`.

Example 19.10 *Output of the Command ping*

```
ping -c 3 example.com
PING example.com (192.168.3.100) 56(84) bytes of data.
64 bytes from example.com (192.168.3.100): icmp_seq=1 ttl=49 time=188 ms
64 bytes from example.com (192.168.3.100): icmp_seq=2 ttl=49 time=184 ms
64 bytes from example.com (192.168.3.100): icmp_seq=3 ttl=49 time=183 ms
--- example.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 183.417/185.447/188.259/2.052 ms
```

The default interval between two packets is one second. To change the interval, ping provides option `-i`. For example to increase ping interval to ten seconds, enter `ping -i 10 example.com`.

In a system with multiple network devices, it is sometimes useful to send the ping through a specific interface address. To do so, use the `-I` option with the name of the selected device, for example, `ping -I wlan1 example.com`.

For more options and information about using ping, enter `ping -h` or see the `ping (8)` man page.

Configuring the Network with ifconfig

`ifconfig` is a traditional network configuration tool. In contrast to ip, you can use it only for interface configuration. If you want to configure routing, use `route`.

NOTE: ifconfig and ip

The program ifconfig is obsolete. Use ip instead.

Without arguments, ifconfig displays the status of the currently active interfaces. As you can see in Example 19.11, “Output of the ifconfig Command” (page 316), ifconfig has very well-arranged and detailed output. The output also contains information about the MAC address of your device, the value of `HWaddr`, in the first line.

Example 19.11 *Output of the ifconfig Command*

```
eth0      Link encap:Ethernet  HWaddr 00:08:74:98:ED:51
          inet6 addr: fe80::208:74ff:fe98:ed51/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:634735 errors:0 dropped:0 overruns:4 frame:0
          TX packets:154779 errors:0 dropped:0 overruns:0 carrier:1
          collisions:0 txqueuelen:1000
          RX bytes:162531992 (155.0 Mb)  TX bytes:49575995 (47.2 Mb)
          Interrupt:11 Base address:0xec80

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8559 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8559 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:533234 (520.7 Kb)  TX bytes:533234 (520.7 Kb)

wlan1     Link encap:Ethernet  HWaddr 00:0E:2E:52:3B:1D
          inet addr:192.168.2.4  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::20e:2eff:fe52:3b1d/64 Scope:Link
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:50828 errors:0 dropped:0 overruns:0 frame:0
          TX packets:43770 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:45978185 (43.8 Mb)  TX bytes:7526693 (7.1 MB)
```

For more options and information about using ifconfig, enter `ifconfig -h` or see the `ifconfig (8)` man page.

Configuring Routing with route

`route` is a program for manipulating the IP routing table. You can use it to view your routing configuration and add or remove of routes.

NOTE: route and ip

The program route is obsolete. Use ip instead.

---

route is especially useful if you need quick and comprehensible information about your routing configuration to determine problems with routing. To view your current routing configuration, enter `route -n` as `root`.

Example 19.12 *Output of the route -n Command*

```
route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.20.0.0       *               255.255.248.0   U         0 0          0 eth0
link-local      *               255.255.0.0     U         0 0          0 eth0
loopback        *               255.0.0.0       U         0 0          0 lo
default         styx.exam.com   0.0.0.0         UG        0 0          0 eth0
```

For more options and information about using route, enter `route -h` or see the `route (8)` man page.

19.6.3 Start-Up Scripts

Apart from the configuration files described above, there are also various scripts that load the network programs while the machine is booting. These are started as soon as the system is switched to one of the *multiuser runlevels*. Some of these scripts are described in Table 19.9, "Some Start-Up Scripts for Network Programs" (page 317).

Table 19.9 *Some Start-Up Scripts for Network Programs*

`/etc/init.d/network`	This script handles the configuration of the network interfaces. If the `network` service was not started, no network interfaces are implemented.
`/etc/init.d/xinetd`	Starts xinetd. xinetd can be used to make server services available on the system. For example, it can start vsftpd whenever an FTP connection is initiated.
`/etc/init.d/portmap`	Starts the portmapper needed for the RPC server, such as an NFS server.
`/etc/init.d/nfsserver`	Starts the NFS server.

`/etc/init.d/postfix`	Controls the postfix process.
`/etc/init.d/ypserv`	Starts the NIS server.
`/etc/init.d/ypbind`	Starts the NIS client.

19.7 smpppd as Dial-up Assistant

Some home users do not have a dedicated line connecting them to the Internet. Instead, they use dial-up connections. Depending on the dial-up method (ISDN or DSL), the connection is controlled by ipppd or pppd. Basically, all that needs to be done to go online is to start these programs correctly.

If you have a flat-rate connection that does not generate any additional costs for the dial-up connection, simply start the respective daemon. Control the dial-up connection with a KDE applet or a command-line interface. If the Internet gateway is not the host you are using, you might want to control the dial-up connection by way of a network host.

This is where smpppd is involved. It provides a uniform interface for auxiliary programs and acts in two directions. First, it programs the required pppd or ipppd and controls its dial-up properties. Second, it makes various providers available to the user programs and transmits information about the current status of the connection. As smpppd can also be controlled by way of the network, it is suitable for controlling dial-up connections to the Internet from a workstation in a private subnetwork.

19.7.1 Configuring smpppd

The connections provided by smpppd are automatically configured by YaST. The actual dial-up programs KInternet and cinternet are also preconfigured. Manual settings are only required to configure additional features of smpppd, such as remote control.

The configuration file of smpppd is `/etc/smpppd.conf`. By default, it does not enable remote control. The most important options of this configuration file are:

open-inet-socket = *yes|no*
: To control smpppd via the network, this option must be set to `yes`. The port on which smpppd listens is `3185`. If this parameter is set to `yes`, the parameters `bind-address`, `host-range`, and `password` should also be set accordingly.

bind-address = *ip address*
: If a host has several IP addresses, use this parameter to determine at which IP address smpppd should accept connections. The default is to listen at all addresses.

host-range = *min ipmax ip*
: The parameter `host-range` defines a network range. Hosts whose IP addresses are within this range are granted access to smpppd. All hosts not within this range are denied access.

password = *password*
: By assigning a password, limit the clients to authorized hosts. As this is a plaintext password, you should not overrate the security it provides. If no password is assigned, all clients are permitted to access smpppd.

slp-register = *yes|no*
: With this parameter, the smpppd service can be announced in the network via SLP.

More information about smpppd is available in the `smpppd(8)` and `smpppd.conf(5)` man pages.

19.7.2 Configuring KInternet and cinternet for Remote Use

KInternet and cinternet can be used to control a local or remote smpppd. cinternet is the command-line counterpart of the graphical KInternet. To prepare these utilities for use with a remote smpppd, edit the configuration file `/etc/smpppd-c.conf` manually or using KInternet. This file only uses four options:

sites = *list of sites*
: Here, tell the front-ends where to search for smpppd. The front-ends test the options in the order specified here. The `local` option orders the establishment of a connection to the local smpppd. The `gateway` option points to an smpppd on the gateway. The `config-file` indicates, that the connection should be established to the smpppd specified in the `server` and `port` options in the `/etc/smpppd-c.conf` file. `slp` orders the front-ends to connect to an smpppd found via SLP.

server = *server*
: Here, specify the host on which smpppd runs.

port = *port*
: Here, specify the port on which smpppd runs.

password = *password*
: Insert the password selected for smpppd.

If smpppd is active, you can now try to access it, for example, with `cinternet --verbose --interface-list`. If you experience difficulties at this point, refer to the `smpppd-c.conf(5)` and `cinternet(8)` man pages.

SLP Services in the Network 20

The *service location protocol* (SLP) was developed to simplify the configuration of networked clients within a local network. To configure a network client, including all required services, the administrator traditionally needs detailed knowledge of the servers available in the network. SLP makes the availability of selected services known to all clients in the local network. Applications that support SLP can use the information distributed and be configured automatically.

openSUSE® supports installation using installation sources provided with SLP and contains many system services with integrated support for SLP. YaST and Konqueror both have appropriate front-ends for SLP. You can use SLP to provide networked clients with central functions, such as an installation server, file server, or print server on your system.

IMPORTANT: SLP Support in openSUSE

Services that offer SLP support include cupsd, rsyncd, ypserv, openldap2, ksysguardd, saned, kdm, vnc, login, smpppd, rpasswd, postfix, and sshd (via fish).

20.1 Installation

Only an SLP client and slptools are installed by default. If you want to provide services via SLP, install the package `openslp-server`. To install the package, start YaST and select *Software* > *Software Management*. Now choose *Filter* > *Patterns* and click *Misc. Server*. Select `openslp-server`. Confirm the installation of the required packages to finish the installation process.

20.2 Activating SLP

slpd must run on your system to offer services with SLP. If the machine should only operate as client, and does not offer services, it is not necessary to run slpd. Like most system services in openSUSE, the slpd daemon is controlled by means of a separate `init` script. After the installation, the daemon is inactive by default. To activate it temporarily, run `rcslpd start` as `root` or `rcslpd stop` to stop it. Perform a restart or status check with `restart` or `status`. If slpd should be always active after booting, enable slpd in YaST *System* > *System Services (Runlevel)* or run the `insserv slpd` command as `root`. This includes slpd in the set of services to be started at boot time.

20.3 SLP Front-Ends in openSUSE

To find services provided via SLP in your network, use an SLP front-end. openSUSE 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 annouce 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.

20.4 Installation over SLP

If you offer an installation server with openSUSE installation media within your network, this can be registered with SLP. For details, see Section 1.2, "Setting Up the Server

Holding the Installation Sources” (page 12). 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.

20.5 Providing Services via SLP

Many applications in openSUSE have integrated SLP support through the use of the `libslp` library. If a service has not been compiled with SLP support, use one of the following methods to make it available via SLP:

Static Registration with `/etc/slp.reg.d`
: Create a separate registration file for each new service. The following is an example of a file for registering a scanner service:

```
## Register a saned service on this system
## en means english language
## 65535 disables the timeout, so the service registration does
## not need refreshes
service:scanner.sane://$HOSTNAME:6566,en,65535
watch-port-tcp=6566
description=SANE scanner daemon
```

The most important line in this file is the *service URL*, which begins with `service:`. This contains the service type (`scanner.sane`) and the address under which the service is available on the server. *`$HOSTNAME`* is automatically replaced with the full hostname. The name of the TCP port on which the relevant service can be found follows, separated by a colon. Then enter the language in which the service should appear and the duration of registration in seconds. These should be separated from the service URL by commas. Set the value for the duration of registration between `0` and `65535`. `0` prevents registration. `65535` removes all restrictions.

The registration file also contains the two variables `watch-port-tcp` and `description`. `watch-port-tcp` links the SLP service announcement to whether the relevant service is active by having slpd check the status of the service. The second variable contains a more precise description of the service that is displayed in suitable browsers.

Static Registration with `/etc/slp.reg`
: The only difference between this method and the procedure with `/etc/slp.reg.d` is that all services are grouped within a central file.

Dynamic Registration with slptool
: If a service should be registerd 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.

20.6 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 openSUSE details, the RFCs, and two introductory HTML documents. Programmers who want to use the SLP functions find more information in the *Programmers Guide* that is included in the `openslp-devel` package.

21 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 19.3, "Name Resolution" (page 281). 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

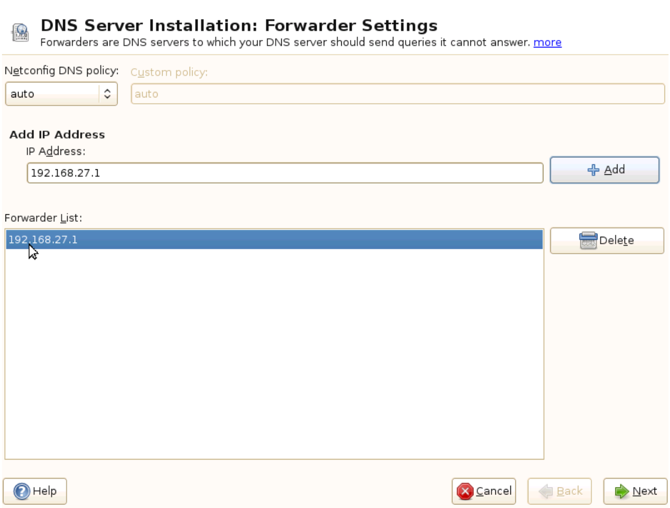
You can use the DNS module of YaST to configure a DNS server for your local network. When starting the module for the first time, a wizard starts, prompting you to make just a few basic decisions concerning administration of the server. Completing this initial setup produces a very basic server configuration that should be functioning in its essential aspects. The expert mode can be used to deal with more advanced configuration tasks.

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.

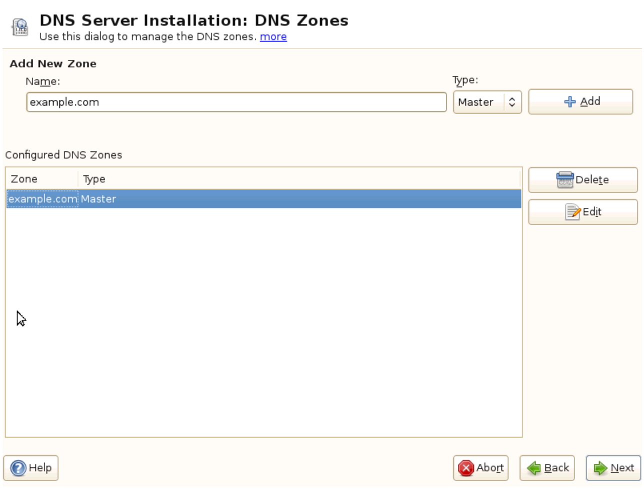
1 When starting the module for the first time, the *Forwarder Settings* dialog, shown in Figure 21.1, "DNS Server Installation: Forwarder Settings" (page 327), opens. The *Netconfig DNS Policy* decides which devices should provide forwarders or whether you want to supply your own *Forwarder List*. For more information about netconfig, see `man 8 netconfig`.

Figure 21.1 *DNS Server Installation: Forwarder Settings*



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 341). 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 328). 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 (*On* or *Off*). You can also activate LDAP support. See Figure 21.3, "DNS Server Installation: Finish Wizard" (page 329).

***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 when the system boots (during 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`. server

Basic Options

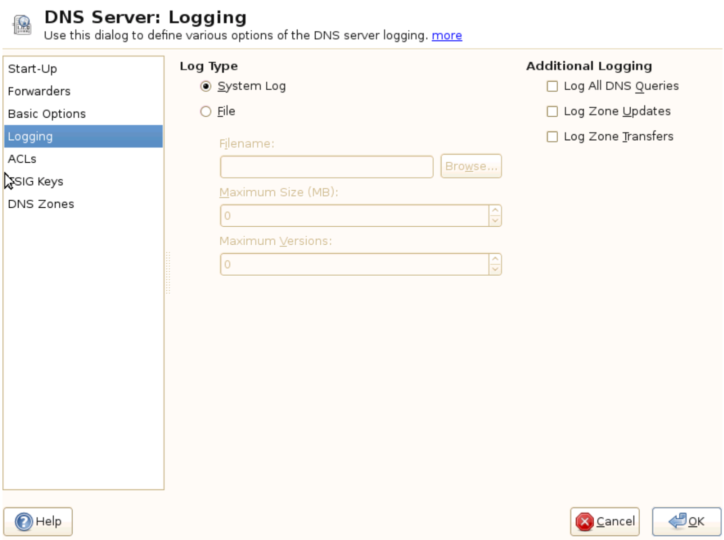
In this section, set basic server options. From the *Option* menu, select the desired item then specify the value in the corresponding entry field. Include the new entry by selecting *Add*.

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 versions of log files 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 331).

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 345).

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 fetch 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`.

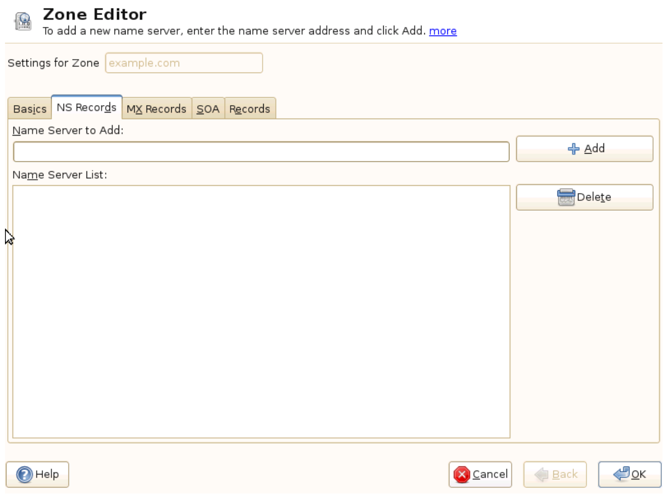
Editing a Master Zone

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*.

In the basic dialog, select whether to enable zone transfers. Use the listed ACLs to define who can download zones.

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 21.5, “DNS Server: Zone Editor (NS Records)” (page 333).

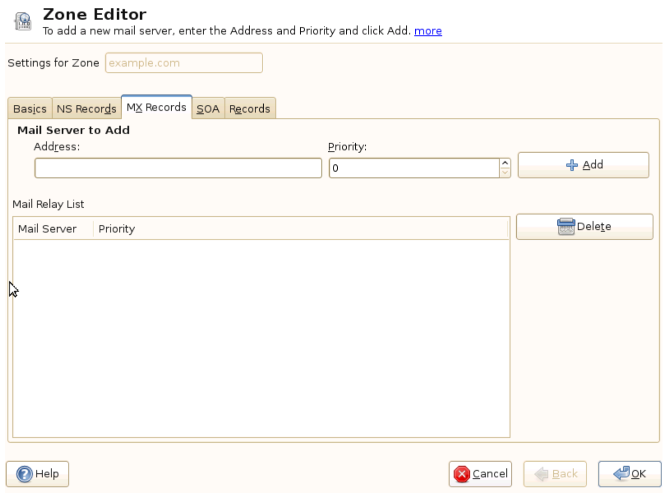
***Figure 21.5** DNS Server: Zone Editor (NS Records)*



Zone Editor (MX Records)

To add a mail server for the current zone to the existing list, enter the corresponding address and priority value. After doing so, confirm by selecting *Add*. See Figure 21.6, "DNS Server: Zone Editor (MX Records)" (page 333).

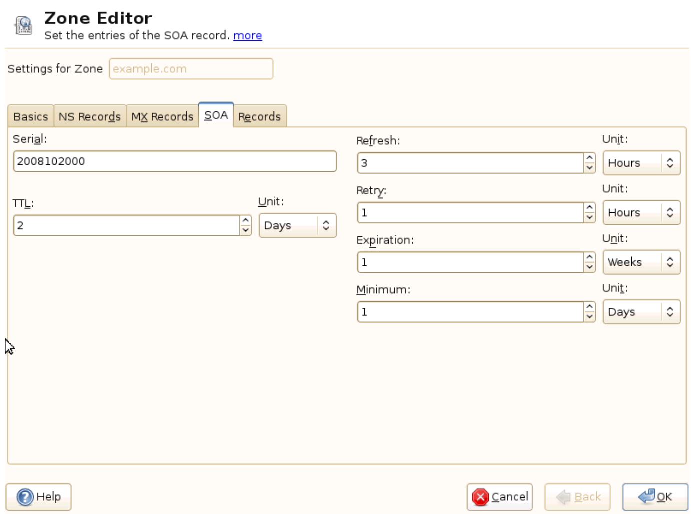
***Figure 21.6** DNS Server: Zone Editor (MX Records)*



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 341).

***Figure 21.7** DNS Server: Zone Editor (SOA)*



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 an openSUSE® 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_DYNAMICALLY` in the file `/etc/sysconfig/network/config` 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, 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 `rcnamed 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 336) 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 `rcnamed reload`. Achieve the same by stopping and restarting the name server with `rcnamed restart`. Stop the server at any time by entering `rcnamed stop`.

21.5 The Configuration File /etc/named.conf

All the settings for the BIND name server itself are stored in the file `/etc/named.conf`. However, the zone data for the domains to handle, consisting of the 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 337).

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 a wild card address.

query-source address * port 53;
: This entry is necessary if a firewall is blocking outgoing DNS requests. This tells BIND to post requests externally from port 53 and not from any of the high ports above 1024.

query-source-v6 address * port 53;
: Tells BIND which port to use for IPv6 queries.

allow-query { 127.0.0.1; *`net`*; };
: Defines the networks from which clients can post DNS requests. Replace *`net`* with address information like `192.168.2.0/24`. 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 339) 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 339). 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 340).

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 fetched from another name server. To differentiate master and slave files, use the directory `slave` for the slave files.

masters { *`server-ip-address`*; };
: This entry is only needed for slave zones. It specifies from which name server the zone file should be transferred.

allow-update {! *; };
: This option controls external write access, which would allow clients to make a DNS entry—something not normally desirable for security reasons. Without this entry, zone updates are not allowed at all. The above entry achieves the same because `! *` effectively bans any such activity.

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 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 dot 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 341).

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, it is a week.

Line 7:

The last entry in the SOA record specifies the `negative caching TTL`—the time for which results of unresolved DNS queries from other servers may be cached.

Line 9:

The `IN NS` specifies the name server responsible for this domain. `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`, because it has two network cards. Wherever the host address is a traditional one (IPv4), the record is marked with `A`. If the address is an IPv6 address, the entry is marked with `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 344).

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`, 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 345).

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 file `/etc/named.conf` 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 openSUSE.

22 DHCP

The purpose of the *dynamic host configuration protocol* (DHCP) is to assign network settings centrally from a server rather than configuring them locally on each and every workstation. A host configured to use DHCP does not have control over its own static address. It is enabled to configure itself completely and automatically according to directions from the server. If you use the NetworkManager on the client side, you do not need to configure the client at all. This is useful if you have changing environments and only one interface active at a time. Never use NetworkManager on a machine that runs a DHCP server.

One way to configure a DHCP server is to identify each client using the hardware address of its network card (which should be fixed in most cases), then supply that client with identical settings each time it connects to the server. DHCP can also be configured to assign addresses to each interested client dynamically from an address pool set up for that purpose. In the latter case, the DHCP server tries to assign the same address to the client each time it receives a request, even over longer periods. This works only if the network does not have more clients than addresses.

DHCP makes life easier for system administrators. Any changes, even bigger ones, related to addresses and the network configuration in general can be implemented centrally by editing the server's configuration file. This is much more convenient than reconfiguring numerous workstations. Also it is 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 192.168.2.100 192.168.2.200;.

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 openSUSE, 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, setup 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.

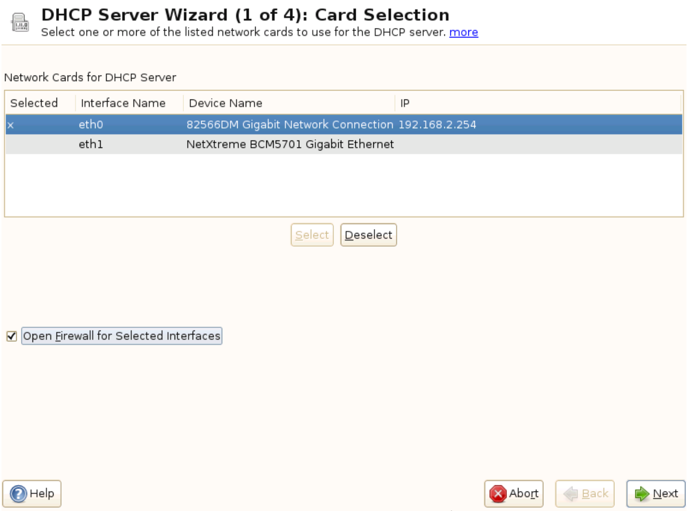
22.1.1 Initial Configuration (Wizard)

When the module is started for the first time, a wizard starts, prompting you to make a few basic decision concerning server administration. Completing this initial setup produces a very basic server configuration that should function in 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 *Add*. After this, select *Open Firewall for Selected*

Interfaces to open the firewall for this interface. See Figure 22.1, "DHCP Server: Card Selection" (page 351).

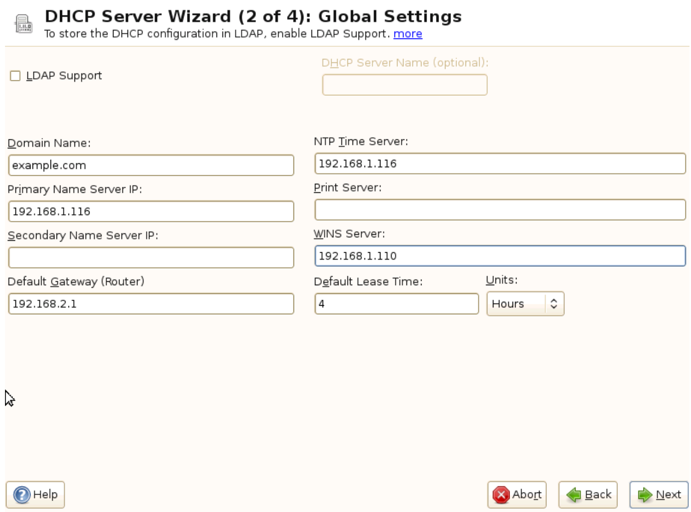
Figure 22.1 *DHCP Server: Card Selection*



Global Settings

Use the check box to determine whether your DHCP settings should be automatically stored by an LDAP server. In the entry fields, provide the network specifics for all clients the DHCP server should manage. These specifics are the domain name, address of a time server, addresses of the primary and secondary name server, addresses of a print and a WINS server (for a mixed network with both Windows and Linux clients), gateway address, and lease time. See Figure 22.2, "DHCP Server: Global Settings" (page 352).

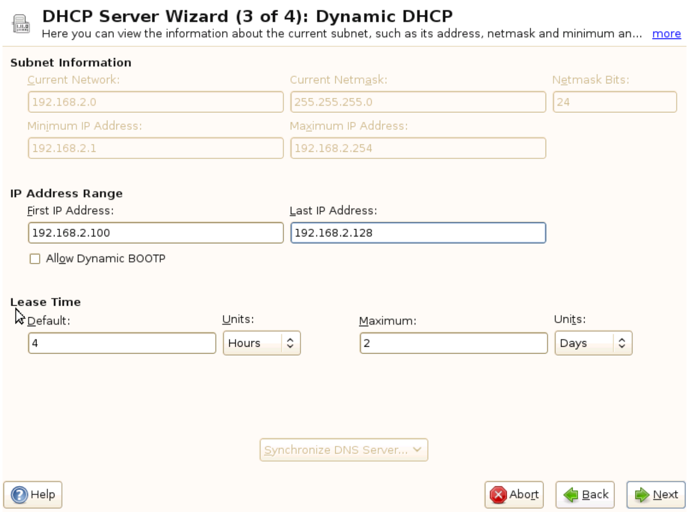
Figure 22.2 *DHCP Server: Global Settings*



Dynamic DHCP

In this step, configure how dynamic IP addresses should be assigned to clients. To do so, specify an IP range from which the server can assign addresses to DHCP clients. All these addresses must be covered by the same netmask. Also specify the lease time during which a client may keep its IP address without needing to request an extension of the lease. Optionally, specify the maximum lease time—the period during which the server reserves an IP address for a particular client. See Figure 22.3, “DHCP Server: Dynamic DHCP” (page 353).

Figure 22.3 *DHCP Server: Dynamic DHCP*



Finishing the Configuration and Setting the Start Mode

After the third part of the configuration wizard, a last dialog is shown in which 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 354).

***Figure 22.4** DHCP Server: Start-Up*

22.2 DHCP Software Packages

Both a DHCP server and DHCP clients are available for openSUSE. The DHCP server available is dhcpd (published by the Internet Systems Consortium). On the client side, choose between two different DHCP client programs: `dhcp-client` (also from ISC) and the DHCP client daemon in the `dhcpcd` package.

openSUSE installs dhcpcd by default. The program is very easy to handle and is launched automatically on each system boot to watch for a DHCP server. It does not need a configuration file to do its job and works out of the box in most standard setups. For more complex situations, use the ISC dhcp-client, which is controlled by means of the configuration file `/etc/dhclient.conf`.

22.3 The DHCP Server dhcpd

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 355).

***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 dhcpd 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. The section also includes a statement of the maximum period for which a machine may keep an IP address assigned by the DHCP server without applying for renewal (`max-lease-time`).

In the second part, some basic network parameters are defined on a global level:

- The line `option domain-name` defines the default domain of your network.

- With the entry `option domain-name-servers`, specify up to three values for the DNS servers used to resolve IP addresses into 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 325).

- 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 355), clients may be given any address between `192.168.2.10` and `192.168.2.20` as well as `192.168.2.100` and `192.168.2.200`.

After editing these few lines, you should be able to activate the DHCP daemon with the command `rcdhcpd start`. It will be ready for use immediately. Use the command `rcdhcpd check-syntax` to perform a brief syntax check. If you encounter any unexpected problems with your configuration—the server aborts with an error or does not return `done` on start—you should be able to find out what has gone wrong by looking for information either in the main system log `/var/log/messages` or on console 10 (Ctrl + Alt + F10).

On a default openSUSE system, the DHCP daemon is started in a chroot environment for security reasons. The configuration files must be copied to the chroot environment so the daemon can find them. Normally, there is no need to worry about this because the command `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 357), are added to the configuration file of Example 22.1, “The Configuration File /etc/dhcpd.conf” (page 355), 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 openSUSE Version

To improve security, the openSUSE version of the ISC's DHCP server comes with the non-root/chroot patch by Ari Edelkind applied. This enables dhcpd to run with the user ID `nobody` and run in a chroot environment (`/var/lib/dhcp`). To make this possible, the configuration file `dhcpd.conf` must be located in `/var/lib/dhcp/etc`. The init script automatically copies the file to this directory when starting.

Control the server's behavior regarding this feature by means of entries in the file `/etc/sysconfig/dhcpd`. To run dhcpd without the chroot environment, set the variable `DHCPD_RUN_CHROOTED` in `/etc/sysconfig/dhcpd` to “no”.

To enable dhcpd to resolve hostnames even from within the chroot environment, some other configuration files must be copied as well:

- `/etc/localtime`
- `/etc/host.conf`
- `/etc/hosts`
- `/etc/resolv.conf`

These files are copied to `/var/lib/dhcp/etc/` when starting the init script. Take these copies into account for any changes that they require if they are dynamically modified by scripts like `/etc/ppp/ip-up`. However, there should be no need to worry about this if the configuration file only specifies IP addresses (instead of hostnames).

If your configuration includes additional files that should be copied into the chroot environment, set these under the variable `DHCPD_CONF_INCLUDE_FILES` in the file `/etc/sysconfig/dhcpd`. To ensure that the DHCP logging facility keeps working even after a restart of the syslog-ng daemon, there is an additional entry `SYSLOGD_ADDITIONAL_SOCKET_DHCP` in the file `/etc/sysconfig/syslog`.

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.

Time Synchronization with NTP 23

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.

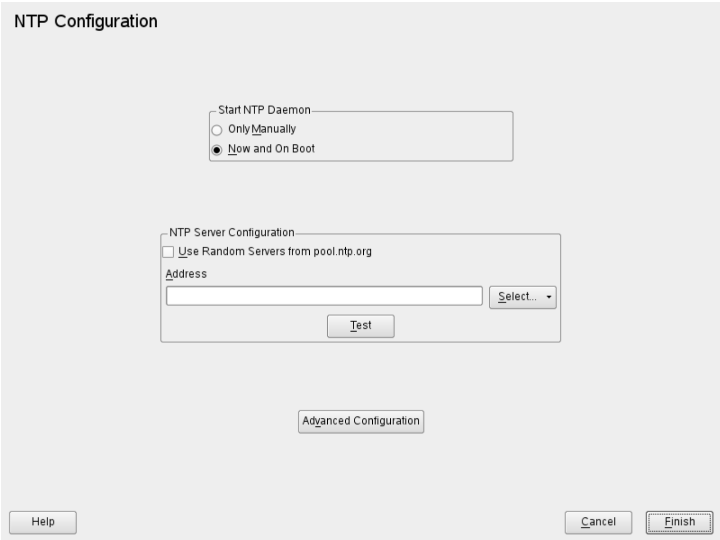
23.1 Configuring an NTP Client with YaST

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.

23.1.1 Quick NTP Client Configuration

The quick NTP client configuration (*Network Services* > *NTP Configuration*) consists of two dialogs. Set the start mode of ntpd and the server to query in the first dialog. To start ntpd automatically when the system is booted, click *Now and On Boot*. Then specify the *NTP Server Configuration*. Either of `0.opensuse.pool.ntp.org`, `1.opensuse.pool.ntp.org`, `2.opensuse.pool.ntp.org`, or `3.opensuse.pool.ntp.org` is pre-selected. Click *Use Random Servers from pool.ntp.org* if you do not want to use the pre-selected time server. Alternatively, click *Select* to access a second dialog in which to select a suitable time server for your network.

Figure 23.1 *YaST: NTP Configuration*

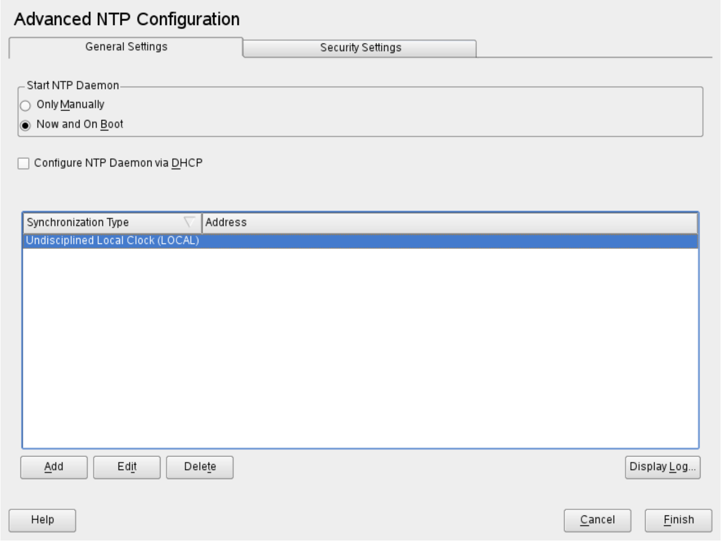


In the pull-down *Select* list, determine whether to implement time synchronization using a time server from your local network (*Local NTP Server*) or an Internet-based time server that takes care of your time zone (*Public NTP Server*). For a local time server, click *Lookup* to start an SLP query for available time servers in your network. Select the most suitable time server from the list of search results and exit the dialog with *OK*. For a public time server, select your country (time zone) and a suitable server from the list under *Public NTP Server* then exit the dialog with *OK*. In the main dialog, test the availability of the selected server with *Test* and quit the dialog with *Finish*.

23.1.2 Advanced NTP Client Configuration

The advanced configuration of an NTP client can be accessed under *Advanced Configuration* from the main dialog of the *NTP Configuration* module, shown in Figure 23.1, "YaST: NTP Configuration" (page 360), after selecting the start-up mode as described in the quick configuration.

Figure 23.2 *Advanced NTP Configuration: General Settings*



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 (as described in Section 23.1.1, "Quick NTP Client Configuration" (page 360)). 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.

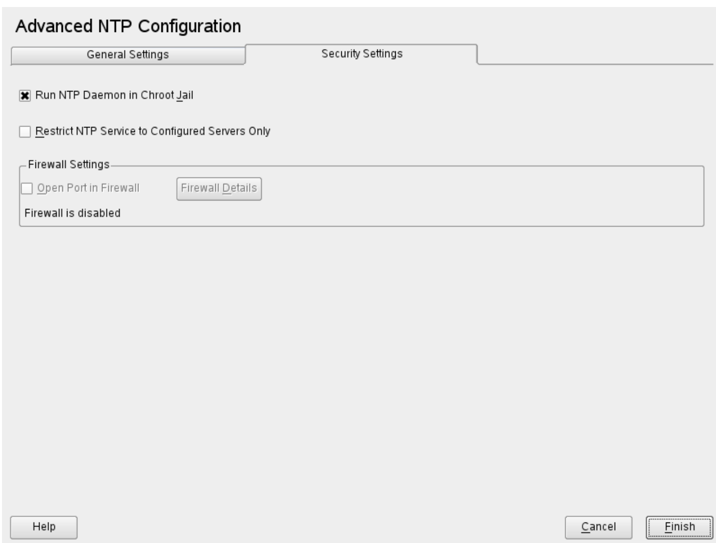
Peer
: A peer is a machine to which a symmetric relationship is established: it acts both as a time server and as a client. To use a peer in the same network instead of a server, enter the address of the system. The rest of the dialog is identical to the *Server* dialog.

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 23.3 *Advanced NTP Configuration: Security Settings*



In the *Security Settings* tab, determine whether ntpd should be started in a chroot jail. By default, *Run NTP Daemon in Chroot Jail* is activated. This increases the security in the event of an attack over ntpd, because it prevents the attacker from compromising the entire system.

Restrict NTP Service to Configured Servers Only increases the security of your system by disallowing remote computers to view and modify NTP settings of your computer and to use the trap facility for remote event logging. Once enabled, these restrictions apply to all remote computers, unless you override the access control options for individual computers in the list of time sources in the *General Settings* tab. For all other remote computers, only querying for local time is allowed.

Enable *Open Port in Firewall* if SuSEfirewall2 is active, which it is by default. If you leave the port closed, it is not possible to establish a connection to the time server.

23.2 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`.

23.3 Setting Up a Local Reference Clock

The software package ntp contains drivers for connecting local reference clocks. A list of supported clocks is available in the `ntp-doc` package in the file `/usr/share/doc/packages/ntp-doc/refclock.html`. Every driver is associated with a number. In ntp, the actual configuration takes place by means of pseudo IP addresses. The clocks are entered in the file `/etc/ntp.conf` as though they existed in the network. For this purpose, they are assigned special IP addresses in the form `127.127.`*`t`*`.`*`u`*. Here, *`t`* stands for the type of the clock and determines which driver is used and *`u`* for the unit, which determines the interface used.

Normally, the individual drivers have special parameters that describe configuration details. The file `/usr/share/doc/packages/ntp-doc/drivers/driver`*`NN`*`.html` (where *`NN`* is the number of the driver) provides information about the particular type of clock. For example, the "type 8" clock (radio clock over serial interface) requires an additional mode that specifies the clock more precisely. The Conrad DCF77 receiver module, for example, has mode 5. To use this clock as a preferred reference, specify the keyword `prefer`. The complete `server` line for a Conrad DCF77 receiver module would be:

```
server 127.127.8.0 mode 5 prefer
```

Other clocks follow the same pattern. Following the installation of the `ntp-doc` package, the documentation for ntp is available in the directory `/usr/share/doc/packages/ntp-doc`. The file `/usr/share/doc/packages/ntp-doc/refclock.html` provides links to the driver pages describing the driver parameters.

24 Sharing File Systems with NFS

Distributing and sharing file systems over a network is a common task in corporate environments. `NFS` is a proven system that also works together with the yellow pages protocol `NIS`. For a more secure protocol that works together with `LDAP` and may also be kerberized, check `NFSv4`.

NFS works with NIS to make a network transparent to the user. With NFS, it is possible to distribute arbitrary file systems over the network. With an appropriate setup, users always find themselves in the same environment regardless of the terminal they currently use.

Like NIS, NFS is a client/server system. However, a machine can be both—it can supply file systems over the network (export) and mount file systems from other hosts (import).

IMPORTANT: Need for DNS

In principle, all exports can be made using IP addresses only. To avoid time-outs, you should have a working DNS system. This is necessary at least for logging purposes, because the mountd daemon does reverse lookups.

24.1 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.

24.2 Importing File Systems with YaST

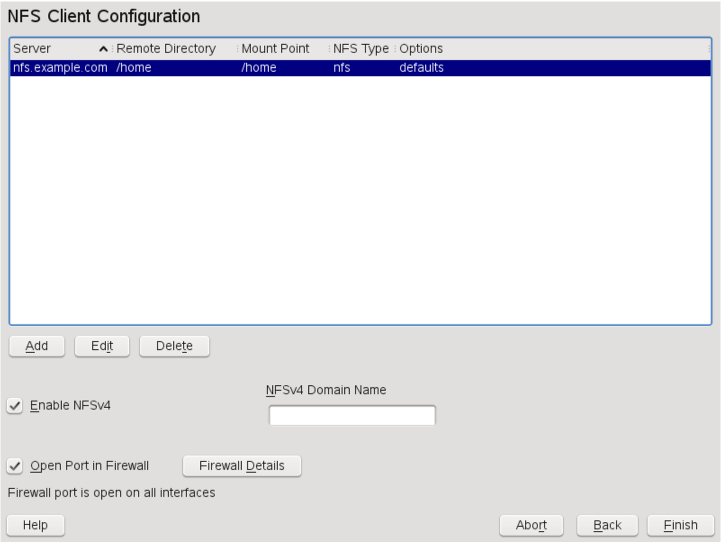
Users authorized to do so can mount NFS directories from an NFS server into their own file trees. This can be achieved using the YaST module *NFS Client*. Click on *Add* and enter the hostname of the NFS server, the directory to import, and the mount point at which to mount this directory locally. The changes will take effect after *Finish* is clicked in the first dialog.

In the *NFS Settings* tab, click *Open Port in Firewall* to open the firewall to allow access to the service from remote computers. The firewall status is displayed next to the check box. When using NFSv4, make sure that the checkbox *Enable NFSv4* is enabled, and that the *NFSv4 Domain Name* contains the same value as used by the NFSv4 server. The default domain is `localdomain`.

Click *Finish* to saves your changes. See Figure 24.1, "NFS Client Configuration with YaST" (page 369).

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 24.1 NFS Client Configuration with YaST



24.3 Importing File Systems Manually

File systems can also be imported manually from an NFS server. The prerequisite for this is a running RPC port mapper, which can be started by entering `rcrpcbind start` as `root`. Once this prerequisite is met, remote exported file systems can be mounted in the file system just like local hard disks using the `mount` command in the following manner:

```
mount host:remote-path local-path
```

If user directories from the machine nfs.example.com, for example, should be imported, use the following command:

```
mount nfs.example.com:/home /home
```

24.3.1 Using the Automount Service

As well as the regular local device mounts, the autofs daemon can be used to mount remote file systems automatically too. To do this, add the following entry in the your `/etc/auto.master` file:

```
/nfsmounts /etc/auto.nfs
```

Now the `/nfsmounts` directory acts as a root for all the NFS mounts on the client if the `auto.nfs` file is completed appropriately. The name `auto.nfs` is chosen for sake of convenience—you can choose any name. In the selected file (create it if it does not exist), add entries for all the NFS mounts as in the following example:

```
localdata -fstype=nfs server1:/data
nfs4mount -fstype=nfs4 server2:/
```

Activate the settings with `rcautofs start`. For this example, `/nfsmounts/localdata`, the `/data` directory of server1, is then mounted with NFS and `/nfsmounts/nfs4mount` from server2 is mounted with NFSv4.

If the `/etc/auto.master` file is edited while the service autofs is running, the automounter must be restarted for the changes to take effect. Do this with `rcautofs restart`.

24.3.2 Manually Editing /etc/fstab

A typical NFSv3 mount entry in `/etc/fstab` looks like this:

```
nfs.example.com:/data /local/path nfs rw,noauto 0 0
```

NFSv4 mounts may also be added to the `/etc/fstab` file manually. For these mounts, use `nfs4` instead of `nfs` in the third column and make sure that the remote file system is given as `/` after the *`nfs.example.com:`* in the first column. A sample line for an NFSv4 mount in `/etc/fstab` looks like this:

```
nfs.example.com:/ /local/pathv4 nfs4 rw,noauto 0 0
```

The `noauto` option prevents the file system from being mounted automatically at start up. If you want to mount the respective file system manually, it is possible to shorten the command for mounting and it is only needed to provide the mount point as in:

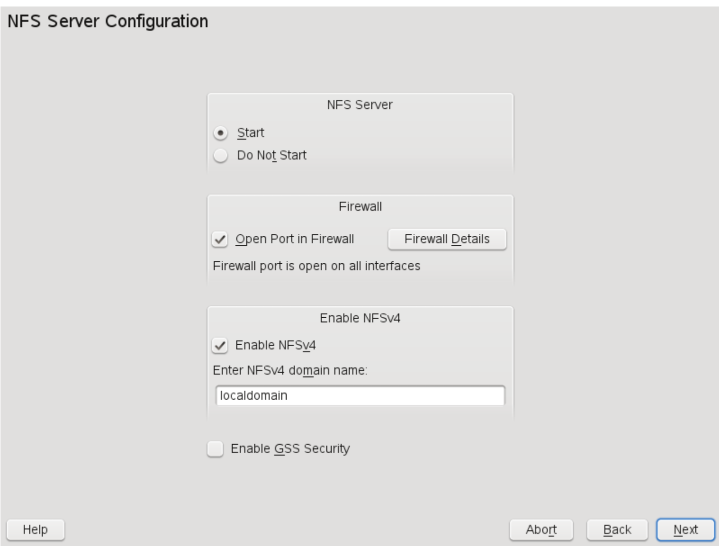
```
mount /local/path
```

Note, that if you do not enter the `noauto` option, the initialization scripts of the system will handle the mount of those file systems at start up.

24.4 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*. A dialog like the one in Figure 24.2, "NFS Server Configuration Tool" (page 371) opens.

Figure 24.2 *NFS Server Configuration Tool*

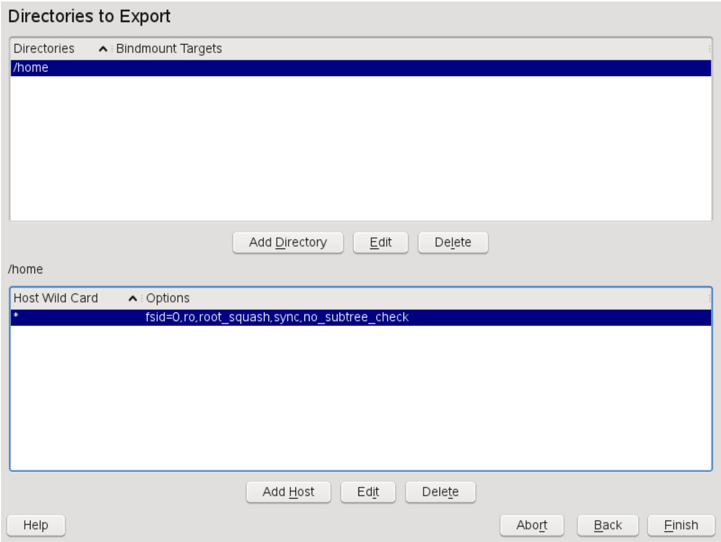


Then activate *Start NFS Server* and enter the *NFSv4 Domain Name*.

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 both, the server and the clients are 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 24.3, "Configuring an NFS Server with YaST" (page 372). The figure shows the scenario where NFSv4 is enabled in the previous dialog. `Bindmount Targets` is shown in the right pane. For more details, refer to the help shown on the left pane. 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 24.3 *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.

24.4.1 Exporting for NFSv4 Clients

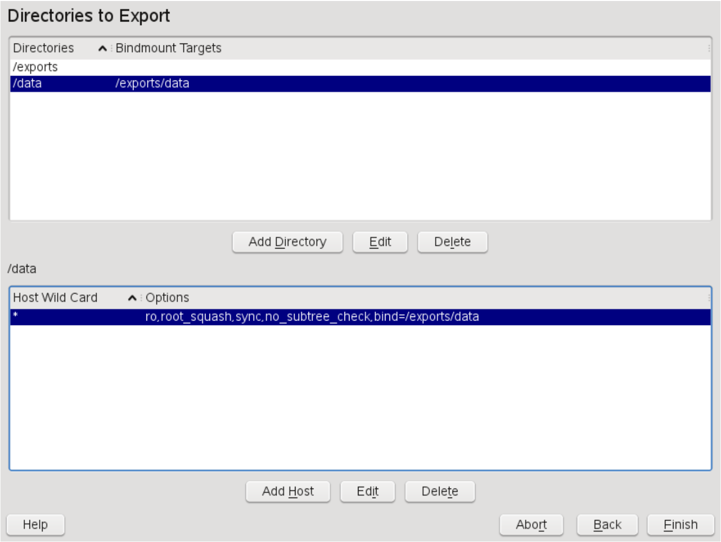
Activate *Enable NFSv4* to support NFSv4 clients. Clients with NFSv3 can still access the server's exported directories if they are exported appropriately. This is explained in detail in Section 24.4.3, “Coexisting v3 and v4 Exports” (page 375).

After activating NFSv4, enter an appropriate domain name. Make sure the name is the same as the one in the `/etc/idmapd.conf` file of any NFSv4 client that accesses this particular server. This parameter is for the idmapd 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 Section 24.7, "For More Information" (page 379).

Click *Next*. The dialog that follows has two sections. The upper half consists of two columns named *Directories* and *Bind Mount Targets*. *Directories* is a directly editable column that lists the directories to export.

For a fixed set of clients, there are two types of directories that can be exported—directories that act as pseudo root file systems and those that are bound to some subdirectory of the pseudo file system. This pseudo file system acts as a base point under which all file systems exported for the same client set take their place. For a client or set of clients, only one directory on the server can be configured as pseudo root for export. For this client, export multiple directories by binding them to some existing subdirectory in the pseudo root.

Figure 24.4 *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 should 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 directly editable column, instead summarizing directories and their nature. After the information is complete, click *Finish* to complete the configuration or *Start* to restart the service.

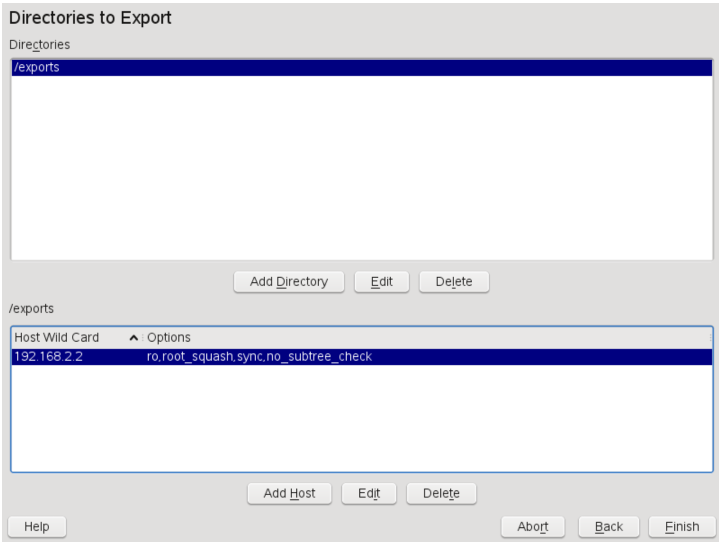
24.4.2 NFSv3 and NFSv2 Exports

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 24.5, "Exporting Directories with NFSv2 and v3" (page 375). Find a more thorough explanation of these options in `man exports`. Click *Finish* to complete the configuration.

Figure 24.5 Exporting Directories with NFSv2 and v3



24.4.3 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 24.3, "Configuring an NFS Server with YaST" (page 372). 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.

24.5 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.

24.5.1 Exporting File Systems with NFSv4

NFSv4 is the latest version of NFS protocol available on openSUSE. 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 normally below this 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 below the `/export`, but we want to export it anyway, 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, that should just mount `servername:/` rather than `servername:/export`. It is not necessary to also mount `servername:/data` and that 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. This 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.

For proper function, 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
```

Do not change these parameters unless you know exactly what you are doing. 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 should start at boot time, run the command `chkconfig nfsserver on`.

24.5.2 Exporting File Systems with NFSv2 and NFSv3

This is specific to NFSv3 and NFSv2 exports. Refer to Section 24.4.1, "Exporting for NFSv4 Clients" (page 372) 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`.

24.6 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. This means that they access the same KDC (Key Distribution Center) server and share their `krb5.keytab` file (the default location on any machine is `/etc/krb5.keytab`).

2. Start the gssd service on the client with `rcgssd start`.

3. Start the svcgssd service on the server with `rcsvcgssd start`.

For further information about configuring kerberized NFS, refer to the links in Section 24.7, "For More Information" (page 379).

24.7 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 any questions on NFSv4, refer to the Linux NFSv4 Frequently Asked Questions [`http://www.citi.umich.edu/projects/nfsv4/linux/faq/`] FAQ.

25 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.

25.1 Terminology

The following are some terms used in Samba documentation and in the YaST module.

SMB protocol
: Samba uses the SMB (server message block) protocol that is based on the NetBIOS services. Microsoft released the protocol so other software manufacturers could establish connections to a Microsoft domain network. With Samba, the SMB protocol works on top of the TCP/IP protocol, so the TCP/IP protocol must be installed on all clients.

CIFS protocol
: CIFS (common Internet file system) protocol is another protocol supported by Samba. CIFS defines a standard remote file system access protocol for use over the network, enabling groups of users to work together and share documents across the network.

NetBIOS
: NetBIOS is a software interface (API) designed for communication between machines providing a name service. It enables machines connected to the network to

reserve names for themselves. After reservation, these machines can be addressed by name. There is no central process that checks names. Any machine on the network can reserve as many names as it wants as long as the names are not already in use. The NetBIOS interface can be implemented for different network architectures. An implementation that works relatively closely with network hardware is called NetBEUI, but this is often referred to as NetBIOS. Network protocols implemented with NetBIOS are IPX from Novell (NetBIOS via TCP/IP) and TCP/IP.

The NetBIOS names sent via TCP/IP have nothing in common with the names used in `/etc/hosts` or those defined by DNS. NetBIOS uses its own, completely independent naming convention. However, it is recommended to use names that correspond to DNS hostnames to make administration easier or use DNS natively. This is the default used by Samba.

Samba server
: Samba server provides SMB/CIFS services and NetBIOS over IP naming services to clients. For Linux, there are three daemons for Samba server: smnd for SMB/CIFS services, nmbd for naming services, and winbind for authentication.

Samba client
: Samba client is a system that uses Samba services from a Samba server over the SMB protocol. All common operating systems, such as Mac OS X, Windows, and OS/2, support the SMB protocol. The TCP/IP protocol must be installed on all computers. Samba provides a client for the different UNIX flavors. For Linux, there is a kernel module for SMB that allows the integration of SMB resources on the Linux system level. You do not need run any daemon for Samba client.

Shares
: SMB servers provide resources to the clients by means of shares. Shares are printers and directories with their subdirectories on the server. It is exported by means of a name and can be accessed by its name. The share name can be set to any name—it does not have to be the name of the export directory. A printer is also assigned a name. Clients can access the printer by its name.

DC
: A domain controller (DC) is a server that handles accounts in domain. For data replication, additional domain controllers are available in one domain.

25.2 Installing a Samba Server

To install a Samba server, start YaST and select *Software* > *Software Management*. Choose *Filter* > *Patterns* and select *File Server*. Confirm the installation of the required packages to finish the installation process.

25.3 Starting and Stopping Samba

You can start or stop the Samba server automatically during boot or manually. Starting and stopping policy is a part of the YaST Samba server configuration described in Section 25.4.1, "Configuring a Samba Server with YaST" (page 383).

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.

25.4 Configuring a Samba Server

A Samba server in openSUSE® can be configured in two different ways: with YaST or manually. Manual configuration offers a higher level of detail, but lacks the convenience of the YaST GUI.

25.4.1 Configuring a Samba Server with YaST

To configure a Samba server, start YaST and select *Network Services* > *Samba Server*.

Initial Samba Configuration

When starting the module for the first time, the *Samba Installation* dialog starts, prompting you to make just a few basic decisions concerning administration of the server then at the end of the configuration prompts for the password of Samba root. 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*, and *Identity* 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 383). 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 25.3, "Starting and Stopping Samba" (page 383).

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 user authentication, click *Advanced Settings*.

25.4.2 Web Administration with SWAT

An alternative tool for Samba server administration is SWAT (Samba Web Administration Tool). It provides a simple Web interface with which to configure the Samba server. To use SWAT, open `http://localhost:901` in a Web browser and log in as user `root`. If you do not have a special Samba root account, use the system `root` account.

NOTE: Activating SWAT

After Samba server installation, SWAT is not activated. To activate it, open *Network Services* > *Network Services (xinetd)* in YaST, enable the network services configuration, select *swat* from the table, and click *Toggle Status (On or Off)*.

25.4.3 Configuring the Server Manually

If you intend to use Samba as a server, install `samba`. The main configuration file of Samba is `/etc/samba/smb.conf`. This file can be divided into two logical parts. The `[global]` section contains the central and global settings. The `[share]` sections contain the individual file and printer shares. By means of this approach, details regarding the shares can be set differently or globally in the `[global]` section, which enhances the structural transparency of the configuration file.

The global Section

The following parameters of the `[global]` section need some adjustment to match the requirements of your network setup so other machines can access your Samba server via SMB in a Windows environment.

workgroup = TUX-NET
: This line assigns the Samba server to a workgroup. Replace `TUX-NET` with an appropriate workgroup of your networking environment. Your Samba server appears under its DNS name unless this name has been assigned to any 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 = 2
: This parameter triggers whether your Samba server tries to become LMB (local master browser) for its workgroup. Choose a very low value to spare the existing Windows network from any disturbances caused by a misconfigured Samba server. More information about this important topic can be found in the files `BROWSING.txt` and `BROWSING-Config.txt` under the `textdocs` subdirectory of the package documentation.

  If no other SMB server is present in your network (such as a Windows 2000 server) and you want the Samba server to keep a list of all systems present in the local environment, set the `os level` to a higher value (for example, `65`). Your Samba server is then chosen as LMB for your local network.

When changing this setting, consider carefully how this could affect an existing Windows network environment. First test the changes in an isolated network or at a noncritical time of day.

wins support and wins server
: To integrate your Samba server into an existing Windows network with an active WINS server, enable the `wins server` option and set its value to the IP address of that WINS server.

  If your Windows machines are connected to separate subnets and should still be aware of each other, you need to set up a WINS server. To turn a Samba server into such a WINS server, set the option `wins support = Yes`. Make sure that only one Samba server of the network has this setting enabled. The options `wins server` and `wins support` must never be enabled at the same time in your `smb.conf` file.

Shares

The following examples illustrate how a CD-ROM drive and the user directories (`homes`) are made available to the SMB clients.

[cdrom]
: To avoid having the CD-ROM drive accidentally made available, these lines are deactivated with comment marks (semicolons in this case). Remove the semicolons in the first column to share the CD-ROM drive with Samba.

Example 25.1 *A CD-ROM Share (deactivated)*

```
;[cdrom]
;       comment = Linux CD-ROM
;       path = /media/cdrom
;       locking = No
```

`[cdrom]` and `comment`
: The entry `[cdrom]` is the name of the share that can be seen by all SMB clients on the 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 25.2 *homes Share*

```
[homes]
 comment = Home Directories
 valid users = %S
 browseable = No
 read only = No
 create mask = 0640
 directory mask = 0750
```

[homes]
: As long as there is no other share using the share name of the user connecting to the SMB server, a share is dynamically generated using the `[homes]` share directives. The resulting name of the share is 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 the user.

`browseable = No`
: This setting makes the share invisible in the network environment.

`read only = No`
: By default, Samba prohibits write access to any exported share by means of the `read only = Yes` parameter. To make a share writable, set the value `read only = No`, which is synonymous with `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 has four possible ways of checking the permissions:

Share Level Security (security = share)
: A password is firmly assigned to a share. Everyone who knows this password has access to that share.

User Level Security (security = user)
: This variation introduces the concept of the user to SMB. Each user must register with the server with his own password. After registration, the server can grant access to individual exported shares dependent on 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 expects an additional parameter (`password server`).

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 HOWTO Collection. For multiple servers on one system, pay attention to the options `interfaces` and `bind interfaces only`.

25.5 Configuring Clients

Clients can only access the Samba server via TCP/IP. NetBEUI and NetBIOS via IPX cannot be used with Samba.

25.5.1 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.

25.6 Samba as Login Server

In networks where predominantly Windows clients are found, it is often preferable that users may only register with a valid account and password. In a Windows-based network, this task is handled by a primary domain controller (PDC). You can use a Windows NT server configured as PDC, but this task can also be done with the help of a Samba server. The entries that must be made in the [global] section of smb.conf are shown in Example 25.3, "Global Section in smb.conf" (page 391).

***Example 25.3** Global Section in smb.conf*

```
[global]
    workgroup = TUX-NET
    domain logons = Yes
    domain master = Yes
```

If encrypted passwords are used for verification purposes the Samba server must be able to handle these. The entry encrypt passwords = yes in the [global] section enables this (with Samba version 3, this is now the default). In addition, it is necessary to prepare user accounts and passwords in an encryption format that conforms with Windows. Do this with the command smbpasswd -a name. Create the domain account for the computers, required by the Windows domain concept, with the following commands:

***Example 25.4** Setting Up a Machine Account*

```
useradd hostname\$
smbpasswd -a -m hostname
```

With the useradd command, a dollar sign is added. The command smbpasswd inserts this automatically when the parameter -m is used. The commented configuration example (/usr/share/doc/packages/Samba/examples/smb.conf.SuSE) contains settings that automate this task.

***Example 25.5** Automated Setup of a Machine Account*

```
add machine script = /usr/sbin/useradd -g nogroup -c "NT Machine Account" \
-s /bin/false %m\$
```

To make sure that Samba can execute this script correctly, choose a Samba user with the required administrator permissions. To do so, select one user and add it to the ntadmin group. After that, all users belonging to this Linux group can be assigned Domain Admin status with the command:

```
net groupmap add ntgroup="Domain Admins" unixgroup=ntadmin
```

More information about this topic is provided in Chapter 12 of the Samba HOWTO Collection, found in /usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf.

25.7 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 HOWTO Collection provided by the Samba team includes a section about troubleshooting. In addition to that, Part V of the document provides a step-by-step guide to checking your configuration. You can find Samba HOWTO Collection in `/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf` after installing the package `samba-doc`.

Also read the Samba page in the openSUSE wiki at http://en.openSUSE.org/Samba.

The Apache HTTP Server 26

With a share of more than 70%, the Apache HTTP Server (Apache) is the world's most widely-used Web server according to the Survey from `http://www.netcraft.com/`. Apache, developed by the Apache Software Foundation (`http://www.apache.org/`), is available for most operating systems. openSUSE® includes Apache version 2.2. In this chapter, learn how to install, configure and set up a Web server; how to use SSL, CGI, and additional modules; and how to troubleshoot Apache.

26.1 Quick Start

With the help of this section, quickly set up and start Apache. You must be `root` to install and configure Apache.

26.1.1 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 19, *Basic Networking* (page 265).

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 23, *Time Synchronization with NTP* (page 359) 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 14, *Masquerading and Firewalls*, ↑Security Guide) gives details.

26.1.2 Installation

Apache on openSUSE 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 26.4, “Installing, Activating, and Configuring Modules” (page 413) for more information about modules.

26.1.3 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 openSUSE and a description of the YaST runlevel editor, refer to Section 14.2.3, “Configuring System Services (Runlevel) with YaST” (page 194).

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 have not received error messages when starting Apache, the Web server should be running now. 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 26.8, “Troubleshooting” (page 432).

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.

26.2 Configuring Apache

Apache in openSUSE can be configured in two different ways: with YaST or manually. Manual configuration offers a higher level of detail, but lacks the convenience of the YaST GUI.

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 26.3, "Starting and Stopping Apache" (page 410). Most configuration changes only require a reload with `rcapache2 reload`.

26.2.1 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 `vhosts.d/vhost.template` for examples. By doing so, you can provide different module sets for different virtual hosts.

`default-server.conf`
: Global configuration for all virtual hosts with reasonable defaults. Instead of changing the values, overwrite them with a virtual host configuration.

`errors.conf`
: Defines how Apache responds to errors. To customize these messages for all virtual hosts, edit this file. Otherwise overwrite these directives in your virtual host configurations.

`httpd.conf`
: The main Apache server configuration file. Avoid changing this file. It mainly contains include statements and global settings. Overwrite global settings in the respective 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 400) 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 26.4, “Installing, Activating, and Configuring Modules” (page 413) 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 26.4.4, “Multiprocessing Modules” (page 417)) 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 26.6, “Setting Up a Secure Web Server with SSL” (page 423) 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 398) 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`. It 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 406)) or by manually editing a configuration file. By default, Apache in openSUSE is prepared for one configuration file per virtual host in `/etc/apache2/vhosts.d/`. All files in this directory with the extension `.conf` are automatically included to the configuration. A basic template for a virtual host is provided in this directory (`vhost.template` or `vhost-ssl.template` for a virtual host with SSL support).

TIP: Always Create a Virtual Host Configuration

It is recommended to always create a virtual host configuration file, even if your Web server only hosts one domain. 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 26.1 *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 26.2 *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 26.3 *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 debugging of errors much easier. `/var/log/apache2/` is the default directory where Apache's log files should be kept.

`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 separate analysis of access statistics for each host. `/var/log/apache2/` is the default directory where Apache's log files should be kept.

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 26.4** 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>
```

26.2.2 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 just a few basic decisions concerning administration of the server. After having finished the wizard, the dialog in Section "HTTP Server Configuration" (page 408) starts every 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 Firewall for Selected Ports* 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

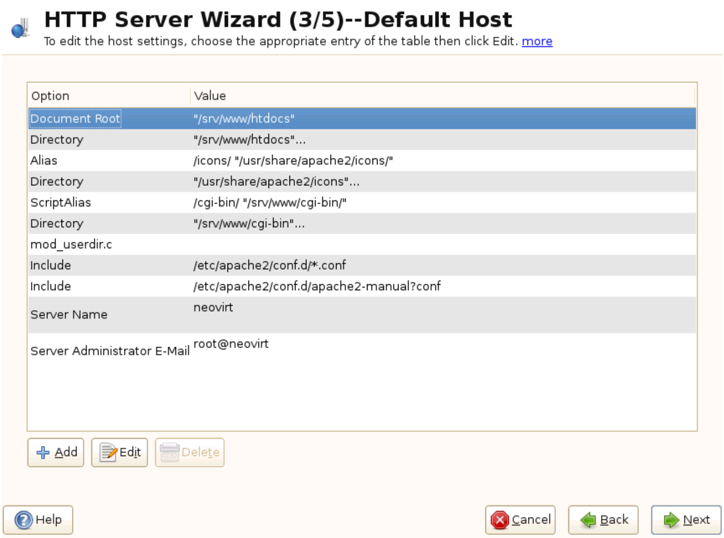
The *Modules* configuration option allows for the activation or deactivation of the script languages, the Web server should support. For the activation or deactivation of other modules, refer to Section “Server Modules” (page 409). Click *Next* to advance to the next dialog.

Default Host

This option pertains to the default Web server. As explained in Section “Virtual Host Configuration” (page 398), 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 26.1 HTTP Server Wizard: Default Host



Here is list of the default settings of the server:

`Document Root`
: Path to the directory from which Apache serves files for this host. `/srv/www/htdocs` is the default location.

`Alias`
: With the help of `Alias` directives, URLs can be mapped to physical file system locations. This means that a certain path even outside the `Document Root` in the file system can be accessed via a URL aliasing that path.

  The default openSUSE `Alias /icons` points to `/usr/share/apache2/icons` for the Apache icons displayed in the directory index view.

`ScriptAlias`
: Similar to the `Alias` directive, the `ScriptAlias` directive maps a URL to a file system location. The difference is that `ScriptAlias` designates the target directory as a CGI location, meaning that CGI scripts should be executed in that location.

`Directory`
: With the `Directory` setting, you can enclose a group of configuration options that will only apply to the specified directory.

  Access and display options for the directories `/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 398)). 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 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 26.6.2, "Configuring Apache with SSL" (page 428) 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 408).

Figure 26.2 *HTTP Server Wizard: Summary*

HTTP Server Configuration

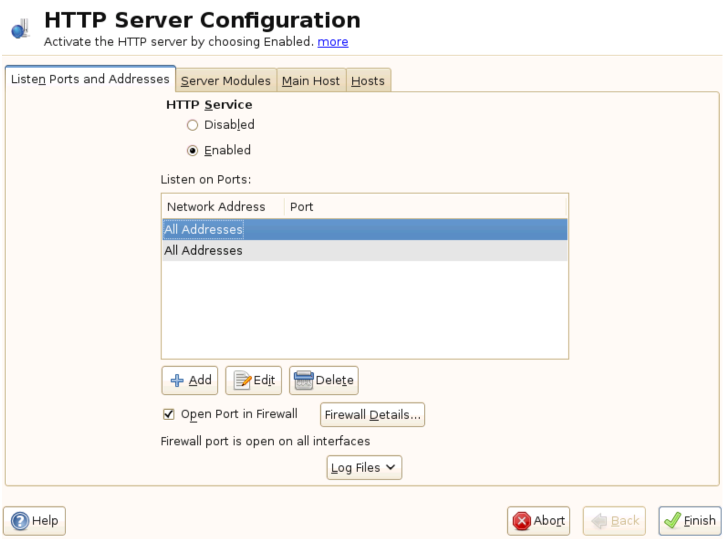
The *HTTP Server Configuration* dialog also lets you make even more adjustments to the configuration than the wizard (which only runs if you configure your Web server for the first time). It consists of four tabs described in the following. No configuration option you change here is effective immediately—you 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 Firewall on Selected Ports*, 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 26.3, "Starting and Stopping Apache" (page 410) for details). These commands are effective immediately.

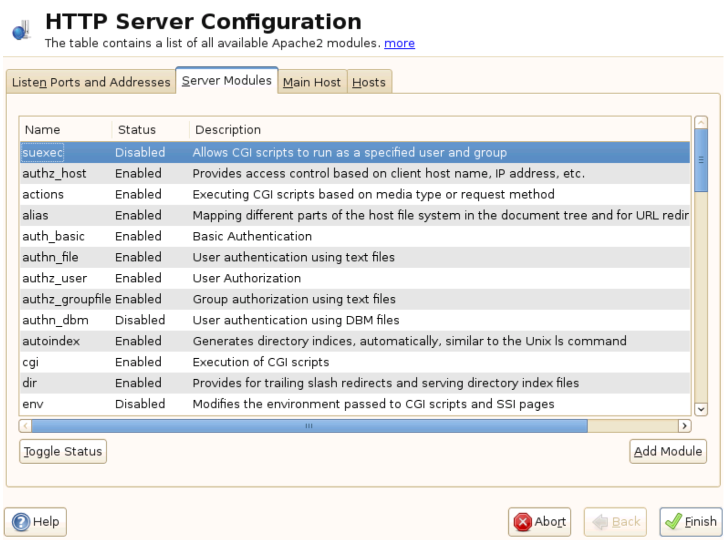
Figure 26.3 *HTTP Server Configuration: Listen Ports and Addresses*



Server Modules

You can change the status (enabled or disabled) of Apache2 modules by clicking *Toggle Status*. Click *Add Module* to add a new module that is already installed but not yet listed. Learn more about modules in Section 26.4, "Installing, Activating, and Configuring Modules" (page 413).

Figure 26.4 *HTTP Server Configuration: Server Modules*



Main Host or Hosts

These dialogs are identical to the ones already described. Refer to Section "Default Host" (page 404) and Section "Virtual Hosts" (page 406).

26.3 Starting and Stopping Apache

If configured with YaST (see Section 26.2.2, "Configuring Apache with YaST" (page 403)), 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 14.2.2, "Init Scripts" (page 190) for a general information about init scripts.). The `rcapache2` command takes the following parameters:

`status`
: Checks whether 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 26.6, “Setting Up a Secure Web Server with SSL” (page 423).

`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 has been running before.

`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

`rcapache2 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`.

: `rcapache2 restart-graceful` is either useful when upgrading to a new version or when having changed configuration options that require a restart. Using this option ensures a minimum server downtime.

`GracefulShutdownTimeout` needs to be set, otherwise `restart-graceful` will result in a regular restart. If set to zero, the server will wait indefinitely until all remaining requests have been fully served.

A graceful restart can fail in case the original Apache instance is not able to clear all necessary ressources. 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.

26.4 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 26.4.2, "Activation and Deactivation" (page 414) 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 has only mod_so (needed to load other modules) and http_core compiled in. All others are available as shared objects: rather than being included in the server binary itself, they can be included at runtime.

Extension Modules
: In general, modules labeled as extensions are included in the Apache software package, but are usually not compiled into the server statically. In openSUSE, they are available as shared objects that can be loaded into Apache at runtime.

External Modules
: Modules labeled external are not included in the official Apache distribution. openSUSE 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.

26.4.1 Module Installation

If you have followed the default way of installing Apache (described in Section 26.1.2, "Installation" (page 394)), 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 result list contains all available external Apache modules.

26.4.2 Activation and Deactivation

Using YaST, you can activate or deactivate the script language modules (PHP5, Perl, Python, and Ruby) with the module configuration described in Section "HTTP Server Wizard" (page 403). All other modules can be enabled or disabled as described in Section "Server Modules" (page 409).

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.

26.4.3 Base and Extension Modules

All base and extension modules are described in detail in the Apache documentation. Only a brief description of the most important modules is available here. Refer to `http://httpd.apache.org/docs/2.2/mod/` to learn details about each module.

mod_actions
: Provides methods to execute a script whenever a certain MIME type (such as `application/pdf`), a file with a specific extension (like `.rpm`), or a certain request method (such as `GET`) is requested. This module is enabled by default.

mod_alias
: Provides `Alias` and `Redirect` directives with which you can map a URl to a specific directory (`Alias`) or redirect a requested URL to another location. This module is enabled by default.

mod_auth*
: The authentication modules provide different authentication methods: basic authentication with mod_auth_basic or digest authentication with mod_auth_digest. Digest authentication in Apache 2.2 is considered experimental.

  mod_auth_basic and mod_auth_digest must be combined with an authentication provider module, mod_authn_* (for example, mod_authn_file for text file–based authentication) and with an authorization module mod_authz_* (for example, mod_authz_user for user authorization).

  More information about this topic is available in the “Authentication HOWTO” at http://httpd.apache.org/docs/2.2/howto/auth.html

mod_autoindex
: Autoindex generates directory listings when no index file (for example, `index.html`) is present. The look and feel of these indexes is configurable. This module is enabled by default. However, directory listings are disabled by default via the `Options` directive—overwrite this setting in your virtual host configuration. The default configuration file for this module is located at `/etc/apache2/mod_autoindex-defaults.conf`.

mod_cgi
: mod_cgi is needed to execute CGI scripts. This module is enabled by default.

mod_deflate
: Using this module, Apache can be configured to compress given file types on the fly before delivering them.

mod_dir
: mod_dir provides the `DirectoryIndex` directive with which you can configure which files are automatically delivered when a directory is requested (`index.html` by default). It also provides an automatic redirect to the correct URl when a directory request does not contain a trailing slash. This module is enabled by default.

mod_env
: Controls the environment that is passed to CGI scripts or SSI pages. Environment variables can be set or unset or passed from the shell that invoked the httpd process. This module is enabled by default.

mod_expires
: With mod_expires, you can control how often proxy and browser caches refresh your documents by sending an `Expires` header. This module is enabled by default.

mod_include
: mod_include lets you use Server Side Includes (SSI), which provide a basic functionality to generate HTML pages dynamically. This module is enabled by default.

mod_info
: Provides a comprehensive overview of the server configuration under http://localhost/server-info/. For security reasons, you should always limit access to this URL. By default only `localhost` is allowed to access this URL. mod_info is configured at `/etc/apache2/mod_info.conf`

mod_log_config
: With this module, you can configure the looks of the Apache log files. This module is enabled by default.

mod_mime
: The mime module takes care 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 26.6, “Setting Up a Secure Web Server with SSL” (page 423) 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.

26.4.4 Multiprocessing Modules

openSUSE 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 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.

26.4.5 External Modules

Find a list of all external modules shipped with openSUSE here. Find the module's documentation in the listed directory.

mod-apparmor
: Adds support to Apache to provide Novell AppArmor confinement to individual CGI scripts handled by modules like mod_php5 and mod_perl.

  Package Name: `apache2-mod_apparmor`
  More Information: Part "Confining Privileges with Novell AppArmor" (↑Security Guide)

mod_mono
: Using mod_mono allows you to run ASP.NET pages in your server.

  Package Name: `apache2-mod_mono`
  Configuration File: `/etc/apache2/conf.d/mod_mono.conf`

mod_perl
: mod_perl enables you to run Perl scripts in an embedded interpreter. The persistent interpreter embedded in the server avoids the overhead of starting an external interpreter and the penalty of Perl start-up time.

  Package Name: `apache2-mod_perl`
  Configuration File: `/etc/apache2/conf.d/mod_perl.conf`
  More Information: `/usr/share/doc/packages/apache2-mod_perl`

mod_php5
: PHP is a server-side, cross-platform HTML embedded scripting language.

  Package Name: `apache2-mod_php5`
  Configuration File: `/etc/apache2/conf.d/php5.conf`
  More Information: `/usr/share/doc/packages/apache2-mod_php5`

mod_python
: mod_python allows embedding Python within the Apache HTTP server for a considerable boost in performance and added flexibility in designing Web-based applications.

  Package Name: `apache2-mod_python`
  More Information: `/usr/share/doc/packages/apache2-mod_python`

mod_tidy
: mod_tidy validates each outgoing HTML page by means of the TidyLib. In case of a validation error, a page with an error list is delivered. Otherwise the original HTML page is delivered.

  Package Name: `apache2-mod_tidy`
  Configuration File: `/etc/apache2/mod_tidy.conf`
  More Information: `/usr/share/doc/packages/apache2-mod_tidy`

26.4.6 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.

26.5 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 26.4.2, "Activation and Deactivation" (page 414) for details on activating modules.

WARNING: CGI Security

Allowing the server to execute CGI scripts is a potential security hole. Refer to Section 26.7, "Avoiding Security Problems" (page 430) for additional information.

26.5.1 Apache Configuration

In openSUSE, the execution of CGI scripts is only allowed in the directory `/srv/www/cgi-bin/`. This location is already configured to execute CGI scripts. If you have created a virtual host configuration (see Section "Virtual Host Configuration" (page 398)) and want to place your scripts in a host-specific directory, you must unlock and configure this directory.

Example 26.5 *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 “deny” statements are evaluated before “allow” statements and access from everywhere is enabled.

26.5.2 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 26.7, “Avoiding Security Problems” (page 430) 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".

26.5.3 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.

26.6 Setting Up a Secure Web Server with SSL

Whenever sensitive data, such as credit card information, is transferred between Web server and client, it would be desirable to have a secure, encrypted connection with authentication. mod_ssl provides strong encryption using the secure sockets layer (SSL) and transport layer security (TLS) protocols for HTTP communication between a client and the Web server. Using SSL/TSL, a private connection between Web server and client is established. Data integrity is ensured and client and server are able to authenticate each other.

For this purpose, the server sends an SSL certificate that holds information proving the server's valid identity before any request to a URL is answered. In turn, this guarantees that the server is the uniquely correct end point for the communication. Additionally, the certificate generates an encrypted connection between client and server that can transport information without the risk of exposing sensitive, plain-text content.

mod_ssl does not implement the SSL/TSL protocols itself, but acts as an interface between Apache and an SSL library. In openSUSE, the OpenSSL library is used. OpenSSL is automatically installed with Apache.

The most visible effect of using mod_ssl with Apache is that URLs are prefixed with `https://` instead of `http://`.

26.6.1 Creating an SSL Certificate

In order to use SSL/TSL with the Web server, you need to create an SSL certificate. This certificate is needed for the authorization between Web server and client, so that each party can clearly identify the other party. To ensure the integrity of the certificate, it must be signed by a party every user trusts.

There are three types of certificates you can create: a "dummy" certificate for testing purposes only, a self-signed certificate for a defined circle of users that trust you, and a certificate signed by an independent, publicly-known certificate authority (CA).

Creating a certificate is basically a two step process. First, a private key for the certificate authority is generated then the server certificate is signed with this key.

TIP: For More Information

To learn more about concepts and definitions of SSL/TSL, refer to http://httpd.apache.org/docs/2.2/ssl/ssl_intro.html.

Creating a "Dummy" Certificate

Generating a dummy certificate is simple. Just call the script `/usr/bin/gensslcert`. It creates or overwrites the following files:

- `/etc/apache2/ssl.crt/ca.crt`

- `/etc/apache2/ssl.crt/server.crt`
- `/etc/apache2/ssl.key/server.key`
- `/etc/apache2/ssl.csr/server.csr`

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 26.1 *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`.

26.6.2 Configuring Apache with SSL

The default port for SSL and TLS requests on the Web server side is 443. There is no conflict between a "regular" Apache listening on port 80 and an SSL/TLS-enabled Apache listening on port 443. In fact, HTTP and HTTPS can be run with the same Apache instance. Usually separate virtual hosts are used to dispatch requests to port 80 and port 443 to separate virtual servers.

IMPORTANT: Firewall Configuration

Do not forget to open the firewall for SSL-enabled Apache on port 443. This can be done with YaST as described in Section “Configuring the Firewall with YaST” (Chapter 14, *Masquerading and Firewalls*, ↑Security Guide).

To use SSL, it must be activated in the global server configuration. Open `/etc/sysconfig/apache2` in an editor and search for `APACHE_MODULES`. Add “ssl” to the list of modules if it is not already present (mod_ssl is activated by default). Next, search for `APACHE_SERVER_FLAGS` and add “SSL”. If you have chosen to encrypt your server certificate with a password, you should also increase the value for `APACHE_TIMEOUT`, 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 398) 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.

26.7 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.

26.7.1 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 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`

26.7.2 DocumentRoot Permissions

By default in openSUSE, the `DocumentRoot` directory `/srv/www/htdocs` and the CGI directory `/srv/www/cgi-bin` belong to the user and group `root`. You should not change these permissions. If the directories 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`.

26.7.3 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 402) for details). In doing so, ensure that no critical files, such as password or system configuration files, can be read from the outside.

26.7.4 CGI Scripts

Interactive scripts in Perl, PHP, SSI, or any other programming language can essentially run arbitrary commands and therefore present a general security issue. Scripts that will be executed from the server should only be installed from sources the server administrator trusts—allowing users to run their own scripts is generally not a good idea. It is also recommended to do security audits for all scripts.

To make the administration of scripts as easy as possible, it is common practice to limit the execution of CGI scripts to specific directories instead of globally allowing them. The directives ScriptAlias and Option ExecCGI are used for configuration. The openSUSE default configuration does not allow execution of CGI scripts from everywhere.

All CGI scripts run as the same user, so different scripts can potentially conflict with each other. The module suEXEC lets you run CGI scripts under a different user and group.

26.7.5 User Directories

When enabling user directories (with mod_userdir or mod_rewrite) you should strongly consider not allowing .htaccess files, which would allow users to overwrite security settings. At least you should limit the user's engagement by using the directive AllowOverRide. In openSUSE, .htaccess files are enabled by default, but the user is not allowed to overwrite any Option directives when using mod_userdir (see the /etc/apache2/mod_userdir.conf configuration file).

26.8 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 26.3, "Starting and Stopping Apache" (page 410)) 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 not to 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 26.2.2, "Configuring Apache with YaST" (page 403)). 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`.

26.9 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.*.

26.9.1 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.

26.9.2 Apache Modules

More information about external Apache modules from Section 26.4.5, “External Modules” (page 419) 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/

mod_tidy
    http://mod-tidy.sourceforge.net/

26.9.3 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/

26.9.4 Miscellaneous Sources

If you experience difficulties specific to Apache in openSUSE, take a look at the openSUSE wiki at http://http://en.opensuse.org/Apache. The history of Apache is provided at http://httpd.apache.org/ABOUT_APACHE.html. This page also explains why the server is called Apache.

Setting up a FTP server with YaST

27

Using the YaST *FTP Server* module, you can configure your machine to function as a 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. Standard openSUSE media does not contain the pure-ftpd package. However, if the pure-ftpd package is installed from another repository, it can be configured using the YaST module.

The vsftpd and pure-ftpd servers have slightly different configuration options, especially in the *Experts Settings* dialog. This chapter describes the settings of the vsftpd for being the default server for openSUSE 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 (vsftpd is the standard server for openSUSE) and confirm the dialog.

3 In the *Start-Up* dialog, configure the starting of the FTP server. For more information, see Section 27.1, "Starting the FTP server" (page 436).

   In the *General* dialog, configure FTP directories, welcome message, file creation masks and various other paramaters. For more information, see Section 27.2, "FTP General Settings" (page 437).

   In the *Performance* dialog, set the parameters that affect the load on the FTP server. For more information, see Section 27.3, "FTP Performance Settings" (page 438).

   In the *Authentication* dialog, set whether the FTP server should be available for anonymous and/or authenticated users. For more information, see Section 27.4, "Authentication" (page 438).

   In the *Expert Settings* dialog, configure the operation mode of the FTP server, SSL connections and firewall settings. For more information, see Section 27.5, "Expert Settings" (page 439).

4 Press *Finish* to save the configurations.

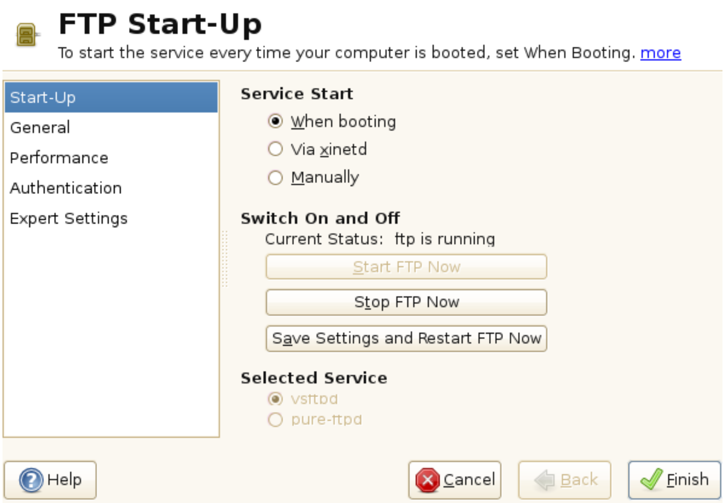
27.1 Starting the FTP server

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 *Accept* 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 pure-ftpd package is not included in the standard openSUSE media so you have to install it from a different installation source if you want to use it.

Figure 27.1 *FTP Server Configuration — Start-Up*



27.2 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.

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. 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. Make sure you removed the write permissions in the directory that was used with pure-ftpd before switching back to the vsftpd server.

27.3 FTP Performance Settings

In the *FTP Performance Settings* 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.

27.4 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.

27.5 Expert Settings

A FTP server can run in active or in passive mode. By default the server runs in passive mode. To switch into the 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 use the FTPS protocol (FTP/SSH). However note that FTPS is different from the much more common SFTP (SSH File Transport Protocol) protocol. If you want to use the FTPS, you can set SSL options in the *Expert Settings* dialog.

If your system is protected by a firewall, check *Open Port in Firewall* to enable a connection to the FTP server.

27.6 For more information

For more information about the vsftpd server read the manual pages of `vsftpd` and `vsftpd.conf`.

Part VI. Mobility

Wireless Communication 28

There are several possibilities for using your Linux system to communicate with other computers, cellular phones, or peripheral devices. WLAN (wireless LAN) can be used to network laptops.

28.1 Wireless LAN

Wireless LANs have become an indispensable aspect of mobile computing. Today, most laptops have built-in WLAN cards. The 802.11 standard for the wireless communication of WLAN cards was prepared by the IEEE organization. Originally, this standard provided for a maximum transmission rate of 2 Mbit/s. Meanwhile, several supplements have been added to increase the data rate. These supplements define details such as the modulation, transmission output, and transmission rates (see Table 28.1, "Overview of Various WLAN Standards" (page 443)). Additionally, a lot of companies implement hardware with proprietary or draft features.

***Table 28.1** 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 draft	2.4 and/or 5	300	Common

802.11 Legacy cards are not supported by openSUSE®. Most cards using 802.11a, 802.11b, 802.11g and 802.11n draft are supported. New cards usually comply with the 802.11n draft standard, but cards using 802.11g are still available.

28.1.1 Function

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

Basically, wireless networks can be classified as managed networks and ad-hoc networks. Managed networks have a managing element: the access point. In this mode (also referred to as infrastructure mode), all connections of the WLAN stations in the network run over the access point, which may also serve as a connection to an ethernet. Ad-hoc networks do not have an access point. The stations communicate directly with each other, therefore an ad-hoc network is usually faster than a managed network. However, the transmission range and number of participating stations are greatly limited in ad-hoc networks. They also do not support WPA authentication. Therefore, an access point is usually used. It is even possible to use a WLAN card as an access point. Some cards support this functionality.

Authentication

Because a wireless network is much easier to intercept and compromise than a wired network, the various standards include authentication and encryption methods. In the original version of the IEEE 802.11 standard, these are described under the term WEP.

However, because WEP has proven to be insecure (see Section “Security” (page 451)), the WLAN industry (joined under the name *Wi-Fi Alliance*) has defined a new extension called WPA, which is supposed to eliminate the weaknesses of WEP. The later IEEE 802.11i standard (also referred to as WPA2, because WPA is based on a draft version 802.11i) includes WPA and some other authentication and encryption methods.

To make sure that only authorized stations can connect, various authentication mechanisms are used in managed networks:

Open
: An open system is a system that does not require authentication. Any station can join the network. Nevertheless, WEP encryption (see Section “Encryption” (page 446)) can be used.

Shared Key (according to IEEE 802.11)
: In this procedure, the WEP key is used for the authentication. However, this procedure is not recommended, because it makes the WEP key more susceptible to attacks. All an attacker needs to do is to listen long enough to the communication between the station and the access point. During the authentication process, both sides exchange the same information, once in encrypted form and once in unencrypted form. This makes it possible for the key to be reconstructed with suitable tools. Because this method makes use of the WEP key for the authentication and for the encryption, it does not enhance the security of the network. A station that has the correct WEP key can authenticate, encrypt, and decrypt. A station that does not have the key cannot decrypt received packets. Accordingly, it cannot communicate, regardless of whether it had to authenticate itself.

WPA-PSK (according to IEEE 802.1x)
: WPA-PSK (PSK stands for preshared key) works similarly to the Shared Key procedure. All participating stations as well as the access point need the same key. The key is 256 bits in length and is usually entered as a passphrase. This system does not need a complex key management like WPA-EAP and is more suitable for private use. Therefore, WPA-PSK is sometimes referred to as WPA “Home”.

WPA-EAP (according to IEEE 802.1x)
: Actually, WPA-EAP is not an authentication system but a protocol for transporting authentication information. WPA-EAP is used to protect wireless networks in enterprises. In private networks, it is scarcely used. For this reason, WPA-EAP is sometimes referred to as WPA “Enterprise”.

WPA-EAP needs a Radius server to authenticate users. EAP offers three different methods for connecting and authenticating to the server: TLS (Transport Layer Security), TTLS (Tunneled Transport Layer Security), and PEAP (Protected Extensible Authentication Protocol). In a nutshell, these options work as follows:

EAP-TLS
: TLS authentication relies on the mutual exchange of certificates both for server and client. First, the server presents its certificate to the client where it is evaluated. If the certificate is considered valid, the client in turn presents its certificate to the server. While TLS is secure, it requires a working certification management infrastructure in your network. This infrastructure is rarely found in private networks.

EAP-TTLS and PEAP
: Both TTLS and PEAP are two-stage protocols. In the first stage, a secure connection is established and in the second one the client authentication data is exchanged. They require far less certification management overhead than TLS, if any.

Encryption

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

WEP (defined in IEEE 802.11)
: This standard makes use of the RC4 encryption algorithm, originally with a key length of 40 bits, later also with 104 bits. Often, the length is declared as 64 bits or 128 bits, depending on whether the 24 bits of the initialization vector are included. However, this standard has some weaknesses. Attacks against the keys generated by this system may be successful. Nevertheless, it is better to use WEP than not encrypt the network at all.

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

TKIP (defined in WPA/IEEE 802.11i)
: This key management protocol defined in the WPA standard uses the same encryption algorithm as WEP, but eliminates its weakness. Because a new key is generated for every data packet, attacks against these keys are in vain. TKIP is used together with WPA-PSK.

CCMP (defined in IEEE 802.11i)
: CCMP describes the key management. Usually, it is used in connection with WPA-EAP, but it can also be used with WPA-PSK. The encryption takes place according to AES and is stronger than the RC4 encryption of the WEP standard.

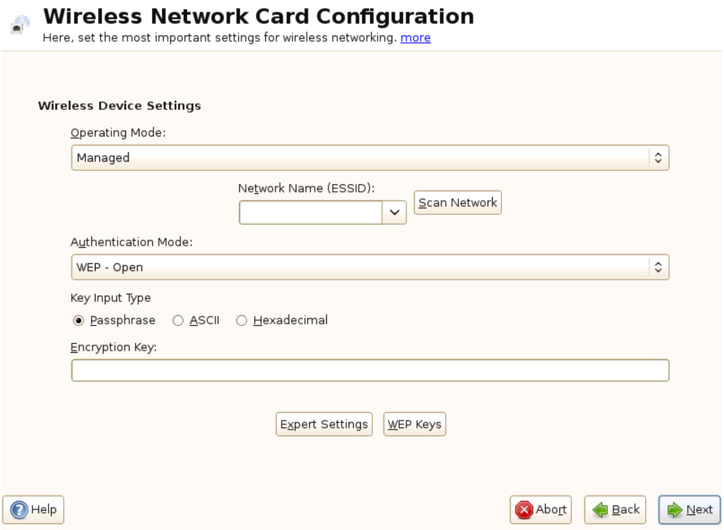
28.1.2 Configuration with YaST

To configure the wireless network card, select *Network Devices* > *Network Settings* in the YaST control center. The Network Settings dialog where you can configure general network settings opens. Please refer to Section 19.4, “Configuring a Network Connection with YaST” (page 283) for more information about the general network configuration. All network cards that have been detected by the system are listed under the *Overview* tab.

Choose your wireless card from the list and click *Edit* to open the Network Card Setup dialog. Configure whether to use a dynamic or a static IP address under the tab *Address*. You can also adjust *General* and *Hardware* settings such as *Device Activation* or *Firewall Zone* and driver settings. In most cases there is no need to change the preconfigured values.

Click *Next* to proceed to the wireless network card specific configuration dialog. If you are using NetworkManager (refer to Section 19.5, “NetworkManager” (page 302) for more information), there is no need to adjust the wireless device settings, since these will be set by NetworkManager on demand—proceed with *Next* and *Yes* to finish the configuration. If you are using your computer only in a specific wireless network, make the basic settings for WLAN operation here.

Figure 28.1 YaST: Configuring the Wireless Network Card



Operating Mode
: A station can be integrated in a WLAN in three different modes. The suitable mode depends on the network in which to communicate: *Ad-hoc* (peer-to-peer network without access point), *Managed* (network is managed by an access point), or *Master* (your network card should be used as the access point). To use any of the WPA-PSK or WPA-EAP modes, the operating mode must be set to *Managed*.

Network Name (ESSID)
: All stations in a wireless network need the same ESSID for communicating with each other. If nothing is specified, the card may automatically selects an access point, which may not be the one you intended to use. Use *Scan Network* for a list of available wireless networks.

Authentication Mode
: Select a suitable authentication method for your network: *No Encryption*, *WEP-Open*, *WEP-Shared Key*, *WPA-EAP*, or *WPA-PSK*. If you select WPA authentication, a network name (ESSID) must be set.

Key Input Type
: WEP and WPA-PSK authentication methods require to input a key. The key has to be entered as either a *Passphrase*, as an *ASCII* string, or *Hexadecimal* string.

WEP Keys

Either enter the default key here or click *WEP Keys* to enter the advanced key configuration dialog. Set the length of the key to *128 bit* or *64 bit*. The default setting is *128 bit*. In the list area at the bottom of the dialog, up to four different keys can be specified for your station to use for the encryption. Press *Set as Default* to define one of them as the default key. Unless you change this, YaST uses the first entered key as the default key. If the standard key is deleted, one of the other keys must be marked manually as the default key. Click *Edit* to modify existing list entries or create new keys. In this case, a pop-up window prompts you to select an input type (*Passphrase*, *ASCII*, or *Hexadecimal*). If you select *Passphrase*, enter a word or a character string from which a key is generated according to the length previously specified. *ASCII* requests an input of 5 characters for a 64-bit key and 13 characters for a 128-bit key. For *Hexadecimal*, enter 10 characters for a 64-bit key or 26 characters for a 128-bit key in hexadecimal notation.

WPA-PSK

To enter a key for WPA-PSK, select the input method *Passphrase* or *Hexadecimal*. In the *Passphrase* mode, the input must be 8 to 63 characters. In the *Hexadecimal* mode, enter 64 characters.

Expert Settings

This button opens a dialog for the detailed configuration of your WLAN connection. Usually there should be no need to change the preconfigured settings.

Channel

The specification of a channel on which the WLAN station should work is only needed in *Ad-hoc* and *Master* modes. In *Managed* mode, the card automatically searches the available channels for access points. In *Ad-hoc* mode, select one of the offered channels (11 to 14, depending on your country) for the communication of your station with the other stations. In *Master* mode, determine on which channel your card should offer access point functionality. The default setting for this option is *Auto*.

Bit Rate

Depending on the performance of your network, you may want to set a certain bit rate for the transmission from one point to another. In the default setting *Auto*, the system tries to use the highest possible data transmission rate. Some WLAN cards do not support the setting of bit rates.

Access Point
: In an environment with several access points, one of them can be preselected by specifying the MAC address.

Use Power Management
: When you are on the road, use power saving technologies to maximize the operating time of your battery. Using power management may affect the connection quality and increase the network latency.

Click next to finish the setup. If you have chosen WPA-EAP authentication, another configuration step is needed before your station is ready for deployment in the WLAN. Enter the credentials you have been given by your network administrator. For TLS, provide *Identity*, *Client Certificate*, *Client Key*, and *Server Certificate*. TTLS and PEAP require *Identity* and *Password*. *Server Certificate* and *Anonymous Identity* are optional. YaST searches for any certificate under `/etc/cert`. Therefore, save the certificates given to you to this location and restrict access to these files to `0600` (owner read and write). Click *Details* to enter the advanced authentication dialog for your WPA-EAP setup. Select the authentication method for the second stage of EAP-TTLS or EAP-PEAP communication. If you selected TTLS in the previous dialog, choose `any`, `MD5`, `GTC`, `CHAP`, `PAP`, `MSCHAPv1`, or `MSCHAPv2`. If you selected PEAP, choose `any`, `MD5`, `GTC`, or `MSCHAPv2`. *PEAP version* can be used to force the use of a certain PEAP implementation if the automatically-determined setting does not work for you.

IMPORTANT: Security in Wireless Networks

Be sure to use one of the supported authentication and encryption methods to protect your network traffic. Unencrypted WLAN connections allow third parties to intercept all network data. Even a weak encryption (WEP) is better than none at all. Refer to Section “Encryption” (page 446) and Section “Security” (page 451) for information.

28.1.3 Utilities

The package `wireless-tools` contains utilities that allow to set wireless LAN specific parameters and get statistics. See `http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html` for more information.

kismet (package `kismet`) is a network diagnosis tool with which to listen to the WLAN packet traffic. In this way, you can also detect any intrusion attempts in your network. More information is available at http://www.kismetwireless.net/ and in the manual page.

28.1.4 Tips and Tricks for Setting Up a WLAN

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

Stability and Speed

The performance and reliability of a wireless network mainly depend on whether the participating stations receive a clean signal from the other stations. Obstructions like walls greatly weaken the signal. The more the signal strength sinks, the more the transmission slows down. During operation, check the signal strength with the iwconfig utility on the command line (`Link Quality` field) or with NetworkManager or KNetworkManager. If you have problems with the signal quality, try to set up the devices somewhere else or adjust the position of the antennas of your access points. Auxiliary antennas that substantially improve the reception are available for a number of PCMCIA WLAN cards. The rate specified by the manufacturer, such as 54 Mbit/s, is a nominal value that represents the theoretical maximum. In practice, the maximum data throughput is no more than half this value.

Security

If you want to set up a wireless network, remember that anybody within the transmission range can easily access it if no security measures are implemented. Therefore, be sure to activate an encryption method. All WLAN cards and access points support WEP encryption. Although this is not entirely safe, it does present an obstacle for a potential attacker. WEP is usually adequate for private use. WPA-PSK would be even better, but it is not implemented in older access points or routers with WLAN functionality. On some devices, WPA can be implemented by means of a firmware update. Furthermore, although Linux supports WPA on most hardware components, some drivers do not offer WPA support. If WPA is not available, WEP is better than no encryption. In enterprises with advanced security requirements, wireless networks should only be operated with WPA.

28.1.5 Troubleshooting

If your WLAN card is not automatically detected, check whether it is supported by openSUSE. A list of supported WLAN network cards is available under http://en.opensuse.org/HCL/Network_Adapters_(Wireless). If your card is not supported, it may be possible to make it work using the Microsoft Windows drivers with Ndiswrapper. Please refer to http://en.opensuse.org/Ndiswrapper for detailed information.

If your WLAN card fails to respond, check if you have downloaded the needed firmware. Refer to `/usr/share/doc/packages/wireless-tools/README.firmware` for more information.

Multiple Network Devices

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

Problems with Prism2 Cards

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

28.1.6 For More Information

The Internet pages of Jean Tourrilhes, who developed the *Wireless Tools* for Linux, present a wealth of useful information about wireless networks. See http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html.

29 Using Tablet PCs

openSUSE® comes with support for Tablet PCs. In the following, learn how to install and configure your Tablet PC and discover some useful Linux* applications which accept input from digital pens.

The following Tablet PCs are supported:

- Tablet PCs with serial Wacom devices, such as ACER TM C30x series, Fujitsu Lifebook T series (T30xx/T40xx/T50xx), Gateway C-140X/E-295C, HP Compaq TC1100/TC4200/TC4400, 2710p/2730p , IBM/Lenovo X41t/X61t, LG LT20, Motion M1200/M1400, OQO 02, Panasonic Toughbook CF-18, Toshiba Portege/Tecra M series, Satellite R15/R20.

- Tablet PCs with Wacom USB devices, such as ASUS R1E/R1F, Gateway C-120X/E-155C, HP Pavilion tx2000/tx2100/tx2500 series.

- Tablet PCs with FinePoint devices, such as Gateway C210X/M280E/CX2724, HP Compaq TC1000.

- Tablet PCs with touch screen devices, such as Asus R2H, Clevo TN120R, Fujitsu Siemens Computers P-Series, LG C1, Samsung Q1/Q1-Ultra.

After you have installed the Tablet PC packages and configured your digitizer correctly, input with the pen, also called a stylus, can be used for the following actions and applications:

- Logging in to KDM or GDM
- Unlocking your screen on the KDE and GNOME desktops
- Actions that can also be triggered by other pointing devices (such as mouse or touch pad), for example, moving the cursor on the screen, starting applications, closing, resizing and moving windows, shifting window focus, dragging and dropping objects
- Using gesture recognition in applications of the X Window System
- Drawing with The GIMP
- Taking notes or sketching with applications like Jarnal or Xournal or editing larger amounts of text with Dasher

---

NOTE: Keyboard or Mouse Needed for Installation

During installation of openSUSE, 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.

29.1 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.

29.2 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.

29.3 Using the Virtual Keyboard

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

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

***Figure 29.1** xvkbd Virtual Keyboard*

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

29.4 Rotating Your Display

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

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

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

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

If you have problems changing the orientation of your desktop, refer to Section 29.7, "Troubleshooting" (page 462) for more information.

29.5 Using Gesture Recognition

openSUSE includes both CellWriter and xstroke for gesture recognition. Both applications accept gestures executed with the pen or other pointing devices as input for applications on the X Window System.

29.5.1 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 29.1 *Training CellWriter*

1. Start CellWriter from the main menu or with `cellwriter` from the command line. On the fist start, CellWriter automatically starts in the training mode. In the training mode, it shows a set of characters of the currently chosen key map.

2. Enter the gesture you would like to use for a character into the respective character's cell. With the first input, the background changes its color to white, whereas the character itself is shown in light grey. Repeat the gesture multiple times until the character changes its color to black. Untrained characters are shown on a light grey or brown background (depending on the desktop's color scheme).

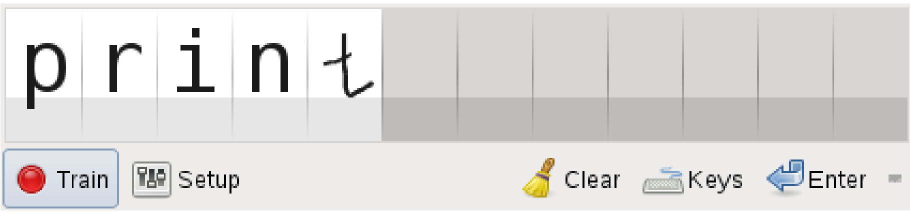
3. Repeat this step until you have trained CellWriter for all characters you need.

4. If you want to train CellWriter for another language, click the *Setup* button and select a language from the *Languages* tab. *Close* the configuration dialog. Click the *Train* button and select the key map from the drop-down box at the bottom right corner of the *CellWriter* window. Now repeat your training for the new map of keys.

5. After having finished the training for the map of keys, click the *Train* button to switch to the normal mode.

In the normal mode, the CellWriter windows shows a couple of empty cells in which to enter the gestures. The characters are not send to another application until you click the *Enter* button, so you can correct or delete characters before you use them as input. Characters that have been recognized with a low degree of confidence will appear highlighted. To correct your input, use the context menu that appears on right-clicking a cell. To delete a character, either use your pen's eraser, or middle-click with the mouse to clear the cell. After finishing your input in CellWriter, define which application should receive the input by clicking into the application's window. Then send the input to the application by clicking *Enter*.

***Figure 29.2** 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.

29.5.2 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).

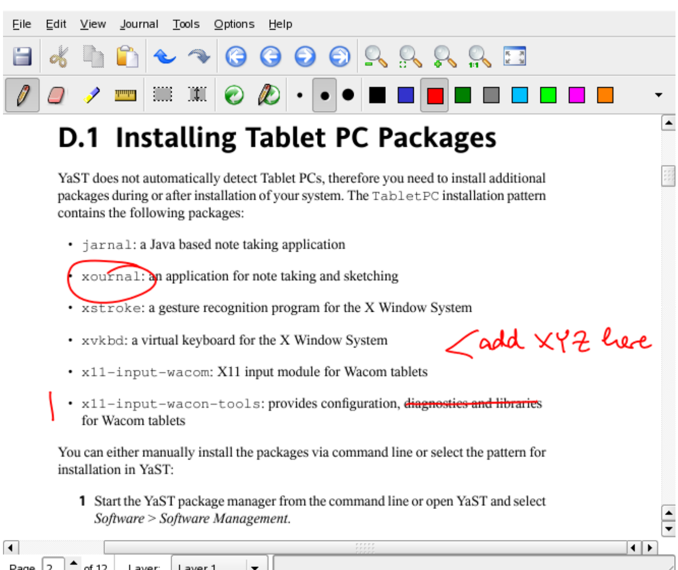
6 To deactivate the gesture recognition mode, click the pencil icon again.

29.6 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*.

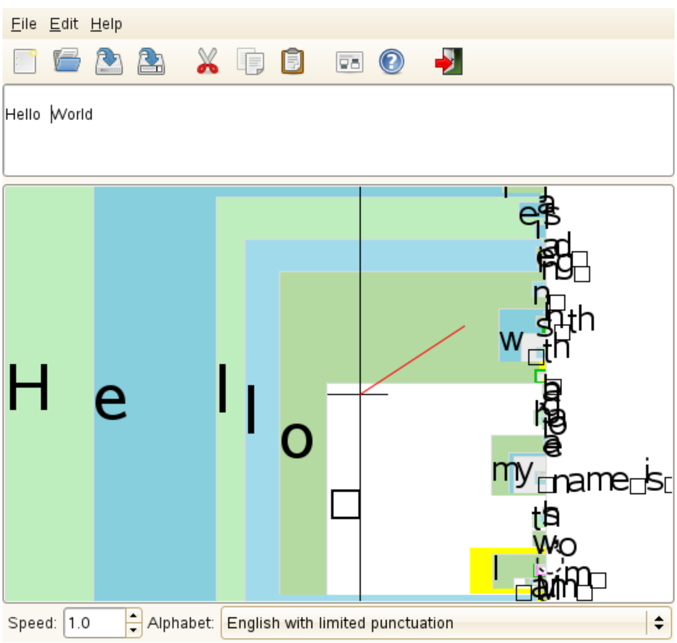
Figure 29.3 *Annotating a PDF with Xournal*



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 29.4 *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/

29.7 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 29.2, "Configuring Your Tablet Device" (page 455), 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.

29.8 For More Information

Some of the applications mentioned here do not offer integrated online help, but you can find some useful information about usage and configuration in your installed system in `/usr/share/doc/package/`*`packagename`* or on the Web:

- For the Xournal manual, refer to http://xournal.sourceforge.net/manual.html
- The Jarnal documentation is located at http://www.dklevine.com/general/software/tc1000/jarnal.htm#documentation
- Find the xstroke man page at http://davesource.com/Projects/xstroke/xstroke.txt
- Find a HOWTO for configuring X on the Linux Wacom Web site: http://linuxwacom.sourceforge.net/index.php/howto/x11
- Find a very informative Web site about the Dasher project at http://www.inference.phy.cam.ac.uk/dasher/
- Find more information and documentation about CellWriter at http://risujin.org/cellwriter/
- Information on gnome-display-properties can be found at http://en.opensuse.org/GNOME/Multiscreen

30 Help and Documentation

openSUSE® comes with various sources of information and documentation, many of which are already integrated in your installed system:

Documentation in `/usr/share/doc`
: This traditional help directory holds various documentation files and the release notes for your system. Find more detailed information in Section 30.1, "Documentation Directory" (page 466).

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 30.2, "Man Pages" (page 468) and Section 30.3, "Info Pages" (page 469).

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 The GIMP, may have different online help packages that can be installed separately with YaST and do not integrate into the help centers.

30.1 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, 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 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.

30.1.1 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 where to find the PDF and HTML versions of the books on your installation media, refer to the openSUSE 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/.

30.1.2 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.

30.1.3 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 is 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.

30.2 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 30.1, "Man Pages—Categories and Descriptions" (page 468) (taken from the man page for man itself).

Table 30.1 *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 `/dev`)
5	File formats and conventions (`/etc/fstab`)
6	Games
7	Miscellaneous (including macro packages and conventions), for example, man(7), groff(7)
8	System administration commands (usually only for `root`)

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.

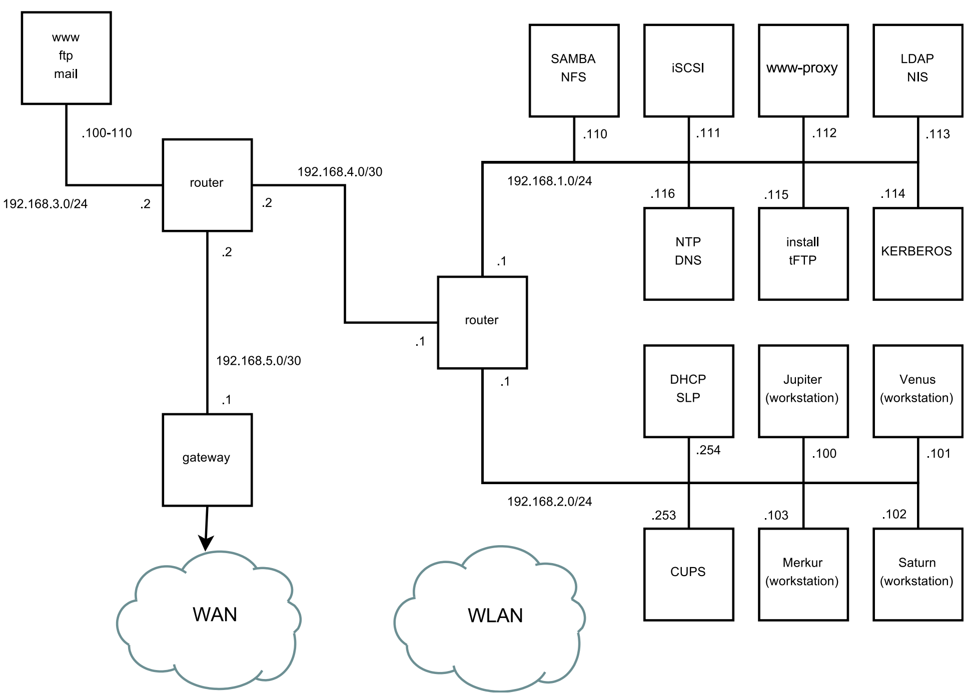
30.3 Info Pages

Info pages are another important source of information on your system. Usually, they are more verbose 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.

A

An Example Network

This example network is used across all network-related chapters of the openSUSE® documentation.



GNU Licenses

B

This appendix contains the GNU General Public License and the GNU Free Documentation License.

GNU General Public License

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and an idea of what it does. Copyright (C) yyyy name of author

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.


This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.


You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'. This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program `Gnomovision'
(which makes passes at compilers) written
by James Hacker.


signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License [http://www.fsf.org/licenses/lgpl.html] instead of this License.

GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c) YEAR YOUR NAME.
```

```
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
only as published by the Free Software Foundation;
with the Invariant Section being this copyright notice and license.
A copy of the license is included in the section entitled “GNU
Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.