

SUSE Linux Enterprise Server 10 Administration



COURSE 3072

Novell Training Services

www.novell.com

AUTHORIZED COURSEWARE

Proprietary Statement

Copyright © 2006 Novell, Inc. All rights reserved.

No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express prior consent of the publisher. This manual, and any portion thereof, may not be copied without the express written permission of Novell, Inc. Novell, Inc.

1800 South Novell Place
Provo, UT 84606-2399

Disclaimer

Novell, Inc. makes no representations or warranties with respect to the contents or use of this manual, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

Further, Novell, Inc. reserves the right to revise this publication and to make changes in its content at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any NetWare software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

Further, Novell, Inc. reserves the right to make changes to any and all parts of NetWare software at any time, without obligation to notify any person or entity of such changes.

This Novell Training Manual is published solely to instruct students in the use of Novell networking software. Although third-party application software packages are used in Novell training courses, this is for demonstration purposes only and shall not constitute an endorsement of any of these software applications.

Further, Novell, Inc. does not represent itself as having any particular expertise in these application software packages and any use by students of the same shall be done at the students' own risk.

Software Piracy

Throughout the world, unauthorized duplication of software is subject to both criminal and civil penalties.

If you know of illegal copying of software, contact your local Software Antipiracy Hotline.

For the Hotline number for your area, access Novell's World Wide Web page at <http://www.novell.com> and look for the piracy page under "Programs."

Or, contact Novell's anti-piracy headquarters in the U.S. at 800-PIRATES (747-2837) or 801-861-7101.

Trademarks

Novell, Inc. has attempted to supply trademark information about company names, products, and services mentioned in this manual. The following list of trademarks was derived from various sources.

Novell, Inc. Trademarks

Novell, the Novell logo, NetWare, BorderManager, ConsoleOne, DirXML, GroupWise, iChain, ManageWise, NDPS, NDS, NetMail, Novell Directory Services, Novell iFolder, Novell SecretStore, Ximian, Ximian Evolution and ZENworks are registered trademarks; CDE, Certified Directory Engineer and CNE are registered service marks; eDirectory, Evolution, exteNd, exteNd Composer, exteNd Directory, exteNd Workbench, Mono, NIMS, NLM, NMAS, Novell Certificate Server, Novell Client, Novell Cluster Services, Novell Distributed Print Services, Novell Internet Messaging System, Novell Storage Services, Nsure, Nsure Resources, Nterprise, Nterprise Branch Office, Red Carpet and Red Carpet Enterprise are trademarks; and Certified Novell Administrator, CNA, Certified Novell Engineer, Certified Novell Instructor, CNI, Master CNE, Master CNI, MCNE, MCNI, Novell Education Academic Partner, NEAP, Ngage, Novell Online Training Provider, NOTP and Novell Technical Services are service marks of Novell, Inc. in the United States and other countries. SUSE is a registered trademark of SUSE LINUX GmbH, a Novell company. For more information on Novell trademarks, please visit <http://www.novell.com/company/legal/trademarks/tmlist.html>.

Other Trademarks

Adaptec is a registered trademark of Adaptec, Inc. AMD is a trademark of Advanced Micro Devices. AppleShare and AppleTalk are registered trademarks of Apple Computer, Inc. ARCserv is a registered trademark of Cheyenne Software, Inc. Btrieve is a registered trademark of Pervasive Software, Inc. EtherTalk is a registered trademark of Apple Computer, Inc. Java is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries. Linux is a registered trademark of Linus Torvalds. LocalTalk is a registered trademark of Apple Computer, Inc. Lotus Notes is a registered trademark of Lotus Development Corporation. Macintosh is a registered trademark of Apple Computer, Inc. Netscape Communicator is a trademark of Netscape Communications Corporation. Netscape Navigator is a registered trademark of Netscape Communications Corporation. Pentium is a registered trademark of Intel Corporation. Solaris is a registered trademark of Sun Microsystems, Inc. The Norton AntiVirus is a trademark of Symantec Corporation. TokenTalk is a registered trademark of Apple Computer, Inc. Tru64 is a trademark of Digital Equipment Corp. UnitedLinux is a registered trademark of UnitedLinux. UNIX is a registered trademark of the Open Group. WebSphere is a trademark of International Business Machines Corporation. Windows and Windows NT are registered trademarks of Microsoft Corporation.

All other third-party trademarks are the property of their respective owners.

Contents

Introduction

Course Objectives	Intro-2
Audience	Intro-2
Certification and Prerequisites	Intro-3
SUSE Linux Enterprise Server 10 Support and Maintenance	Intro-5
Novell Customer Center	Intro-6
SUSE Linux Enterprise Server 10 Online Resources	Intro-7
Agenda	Intro-8
Scenario	Intro-9
Exercises	Intro-9
Exercise Conventions	Intro-10

SECTION 1 **Install SUSE Linux Enterprise Server 10**

Objectives	1-1
Objective 1 Perform a SLES 10 Installation	1-2
Boot From the Installation Media	1-2
Select the System Language	1-5
Select the Installation Mode	1-6
Set the Clock and Time Zone	1-8
Understand and Change the Installation Settings	1-9
Verify Partitioning	1-10
Select Software	1-25

	Start the Installation Process	1-28
Objective 2	Configure the SLES 10 Installation	1-29
	Set the Hostname	1-29
	Set the root Password	1-29
	Configure the Network	1-31
	Test the Internet Connection	1-38
	Novell Customer Center Configuration and Online Update .	1-39
	Configure Network Services	1-42
	Manage Users	1-43
	Configure Hardware	1-47
	Finalize the Installation Process	1-49
Objective 3	Troubleshoot the Installation Process	1-50
	Exercise 1-1 Install SUSE Linux Enterprise Server 10	1-53
	Summary	1-54

SECTION 2 Administer the Linux File System

	Objectives	2-1
Objective 1	Select a Linux File System	2-2
	Linux File Systems	2-3
	Virtual Filesystem Switch	2-5
	Linux File System Internals	2-6
	File System Journaling	2-13
	Additional File System Documentation	2-14
Objective 2	Configure Linux File System Partitions	2-16
	Linux Device and Partition Names	2-16
	Design Guidelines for Implementing Partitions	2-18
	Manage Partitions with YaST	2-21
	Manage Partitions with fdisk	2-23
Objective 3	Manage Linux File Systems.	2-30
	Create a File System Using YaST	2-30
	Create a File System Using Command Line Tools	2-32

	Mount File Systems	2-36
	Exercise 2-1 Configure Partitions on Your Hard Drive	2-43
	Monitor and Check a File System	2-44
	Exercise 2-2 Manage File Systems from the Command Line	2-50
Objective 4	Configure Logical Volume Manager (LVM) and Software RAID	2-51
	How to Use VM Components	2-51
	How to Use VM Features	2-53
	How to Configure Logical Volumes With YaST	2-54
	How to Configure LVM with Command Line Tools	2-60
	Manage Software RAID	2-63
	Exercise 2-3 Create Logical Volumes	2-66
Objective 5	Set Up and Configure Disk Quotas	2-67
	Prepare the File System	2-68
	Initialize the Quota System	2-69
	Start and Activate the Quota Service	2-69
	Configure and Manage User and Group Quotas	2-70
	Exercise 2-4 Set Up and Configure Disk Quotas	2-74
	Summary	2-75
SECTION 3	Administer User Access and Security	
	Objectives	3-1
Objective 1	Configure User Authentication with PAM	3-2
	Location and Purpose of PAM Configuration Files	3-4
	PAM Configuration	3-5
	PAM Configuration File Examples	3-8
	Secure Password Guidelines	3-11
	PAM Documentation Resources	3-12
	Exercise 3-1 Configure PAM Authentication	3-13
Objective 2	Manage and Secure the Linux User Environment	3-14
	Perform Administrative Tasks as root	3-14

	Delegate Administrative Tasks With sudo	3-16
	Set Defaults for New User Accounts	3-19
	Configure Security Settings	3-22
	Exercise 3-2 Configure the Password Security Settings	3-33
Objective 3	Use Access Control Lists (ACLs) for Advanced Access Control	3-34
	The Basics of ACLs	3-34
	Basic ACL commands	3-35
	Important ACL Terms	3-36
	ACL Types	3-37
	How ACLs and Permission Bits Map to Each Other	3-39
	How to Use the ACL Command Line Tools	3-41
	How to Configure a Directory with an Access ACL	3-42
	How to Configure a Directory with a Default ACL	3-47
	Additional setfacl Options	3-51
	The ACL Check Algorithm	3-51
	How Applications Handle ACLs	3-52
	Exercise 3-3 Use ACLs	3-53
	Summary	3-54
SECTION 4	Configure the Network Manually	
	Objectives	4-1
Objective 1	Understand Linux Network Terms	4-2
Objective 2	Set Up Network Interfaces with the ip Tool	4-3
	Display the Current Network Configuration	4-3
	Change the Current Network Configuration	4-8
	Save Device Settings to a Configuration File	4-10
Objective 3	Set Up Routing with the ip Tool	4-15
	View the Routing Table	4-15
	Add Routes to the Routing Table	4-16
	Delete Routes from the Routing Table	4-18

	Save Routing Settings to a Configuration File	4-18
Objective 4	Test the Network Connection With Command Line Tools . . .	4-20
	Test Network Connections with ping	4-20
	Trace Network Packets with traceroute	4-22
	Exercise 4-1 Configure the Network Connection Manually . .	4-24
Objective 5	Configure Host Name and Name Resolution.	4-25
	Set the Host and Domain Name	4-25
	Configure Name Resolution	4-25
Objective 6	Use the NetworkManager to Configure the Network	4-27
	Summary	4-30

SECTION 5 Administer Linux Processes and Services

	Objectives	5-1
Objective 1	View and Manage Processes	5-2
	Understand Process Definitions	5-2
	Learn Jobs and Processes	5-5
	Manage Foreground and Background Processes	5-5
	View and Prioritize Processes	5-8
	End a Process	5-16
	Understand Services (Daemons)	5-20
	Manage a Daemon Process	5-21
	Exercise 5-1 Manage Linux Processes	5-24
Objective 2	Schedule Jobs.	5-25
	Schedule a Job (cron)	5-25
	Run a Job One Time Only (at)	5-30
	Exercise 5-2 Schedule Jobs with cron and at	5-32
	Summary	5-33

SECTION 6 **Monitor SUSE Linux Enterprise Server 10**

	Objectives	6-1
Objective 1	Monitor a SUSE Linux Enterprise Server 10 System	6-2
	Boot Log Information	6-2
	Hardware Information (/proc/)	6-5
	Hardware Information (Command Line Utilities)	6-5
	System and Process Information (Command Line Utilities) ..	6-7
	Monitor Hard Drive Space	6-10
	Exercise 6-1 Gather Information About Your SUSE Linux Enterprise Server 10 Server	6-11
Objective 2	Use System Logging Services	6-12
	The Syslog Daemon syslog-ng	6-12
	Important Log Files	6-21
	Archive Log Files (logrotate)	6-23
	Exercise 6-2 Manage System Logging	6-27
Objective 3	Monitor Login Activity	6-28
	Summary	6-33

SECTION 7 **Manage System Initialization**

	Objectives	7-1
Objective 1	Describe the Linux Load Procedure	7-2
Objective 2	GRUB (Grand Unified Bootloader)	7-7
	What a Boot Manager Is	7-7
	Boot Managers in SUSE Linux	7-8
	Start the GRUB Shell	7-10
	Modify the GRUB Configuration File	7-11
	Configure GRUB with YaST	7-13
	Boot a System Directly into a Shell	7-18
	Exercise 7-1 Manage the Boot Loader	7-21

Objective 3	Manage Runlevels	7-22
	The init Program and Linux Runlevels	7-22
	init Scripts and Runlevel Directories	7-27
	Change the Runlevel	7-39
	Exercise 7-2 Manage Runlevels	7-42
	Summary	7-43
 SECTION 8 Manage Software for SUSE Linux Enterprise Server		
	Objectives	8-1
Objective 1	Manage RPM Software Packages	8-2
	RPM Components and Features	8-2
	RPM Basics	8-4
	Manage Software Packages with rpm	8-6
	Exercise 8-1 Manage Software with RPM	8-17
Objective 2	Verify and Update Software Library Access	8-18
	Software Library Basics	8-18
	View Shared Library Dependencies (ldd)	8-20
	Modify the Software Library Configuration File (/etc/ld.so.conf)	8-22
	Update the Library Cache (/etc/ld.so.cache)	8-23
	Exercise 8-2 Manage Shared Libraries	8-24
	Summary	8-25
 SECTION 9 Manage Backup and Recovery		
	Objectives	9-1
	Introduction	9-2
Objective 1	Develop a Backup Strategy	9-3
	Choose a Backup Method	9-3
	Choose the Right Backup Media	9-6

Objective 2	Backup Files with YaST	9-7
	Back Up System Data with YaST	9-7
	Restore System Data with YaST	9-13
	Exercise 9-1 Backup Files with YaST	9-18
Objective 3	Create Backups with tar	9-19
	Create tar Archives	9-19
	Unpack tar Archives	9-20
	Exclude Files from Backup	9-21
	Perform Incremental and Differential Backups	9-21
	Use tar Command Line Options	9-24
	Exercise 9-2 Create Backup Files with tar	9-25
Objective 4	Work with Magnetic Tapes	9-26
Objective 5	Copy Data with dd	9-29
	Exercise 9-3 Create Drive Images with dd	9-31
Objective 6	Mirror Directories with rsync	9-32
	Perform Local Copying with rsync	9-32
	Perform Remote Copying with rsync	9-34
	Exercise 9-4 Create a Backup of a Home Directory with rsync	9-36
Objective 7	Automate Data Backups with cron	9-37
	Exercise 9-5 Configure a cron Job for Data Backups	9-38
	Summary	9-39
 SECTION 10 Manage Printing		
	Objectives	10-1
Objective 1	Configure Local Printing	10-2
	When to Configure a Printer	10-2
	Required Printing Software	10-3
	Add a Printer	10-4
	Exercise 10-1 Change Your Printer Configuration	10-20

Objective 2	Manage Print Jobs and Queues	10-21
	Generate a Print Job	10-22
	Display Information on Print Jobs	10-23
	Cancel Print Jobs	10-24
	Manage Queues	10-25
	Configure Queues	10-26
	Start and Stop CUPS	10-30
	Exercise 10-2 Manage Printers from the Command Line. . .	10-31
Objective 3	Understand How CUPS Works	10-32
	Steps of the Printing Process	10-32
	Print Queues	10-34
	Log Files	10-37
	Configuration File	10-41
Objective 4	Configure and Manage a Print Server	10-42
	Broadcast Information about Printers to other Computers .	10-43
	Access Restrictions	10-47
	Restrict Access to Printers for Users and Groups	10-50
	Restrict Access to the Web Interface	10-52
	Exercise 10-3 Restrict Access	10-54
Objective 5	Use the Web Interface to Manage a CUPS Server	10-55
	Do Administration Tasks	10-56
	Manage Printer Classes	10-57
	On-Line Help	10-58
	Manage Jobs	10-58
	Manage Printers	10-59
	Exercise 10-4 Use the Web Interface to Manage a CUPS Server	10-61
	Summary	10-62

SECTION 11 **Configure Remote Access**

	Objectives	11-1
Objective 1	Provide Secure Remote Access with OpenSSH	11-2
	Cryptography Basics	11-3
	SSH Features and Architecture	11-6
	Configure the SSH Server	11-14
	Configure the SSH Client	11-15
	SSH-related Commands	11-16
	Exercise 11-1 Practice Using OpenSSH.	11-21
	Public Key Authentication Management	11-22
	Exercise 11-2 Perform Public Key Authentication	11-27
Objective 2	Enable Remote Administration with YaST	11-28
	VNC and YaST Remote Administration	11-28
	Configure Your Server for Remote Administration	11-29
	Access Your Server for Remote Administration	11-31
	Exercise 11-3 Use Remote Administration	11-33
	Summary	11-34

Introduction

SUSE Linux Enterprise Server 10 Administration (Course 3072) focuses on the routine system administration of SUSE Linux Enterprise Server 10.

This course covers common tasks a system administrator of SUSE Linux Enterprise Server 10 has to perform, like installation and configuration of the system, maintenance of the file system, software management, management of processes, and printing.

These skills, along with those taught in *SUSE Linux Enterprise Server 10 Fundamentals* (Course 3071) and *SUSE Linux Enterprise Server 10 Advanced Administration* (Course 3073), prepare you to take the Novell Certified Linux Professional 10 (Novell CLP 10) certification practicum test.

The contents of your student kit include the following:

- *SUSE Linux Enterprise Server 10 Administration Manual*
- *SUSE Linux Enterprise Server 10 Administration Workbook*
- *SUSE Linux Enterprise Server 10 Administration Course DVD*
- *SUSE LINUX Enterprise Server 10 Product DVD*
- *SUSE LINUX Enterprise Desktop 10 Product DVD*

The *SUSE Linux Enterprise Server 10 Administration Course DVD* contains an image of a SUSE Linux Enterprise Server 10 installation that you can use with the *SUSE Linux Enterprise Server 10 Administration Workbook* outside the classroom to practice the skills you need to take the Novell CLP 10 Practicum exam.



Instructions for setting up a self-study environment are in the setup directory on the Course DVD.

Course Objectives

This course teaches theory as well as practical application with hands-on labs of the following *SUSE Linux Enterprise Server 10 Administration* topics on SUSE Linux Enterprise Server 10:

1. Install SUSE Linux Enterprise Server
2. Administer the Linux File System
3. Administer User Access and Security
4. Configure the Network Manually
5. Administer Linux Processes and Services
6. Monitor SUSE Linux Enterprise Server
7. Configure System Initialization
8. Manage Software for SUSE Linux Enterprise Server
9. Manage Backup and Recovery
10. Administer Printing
11. Configure Remote Access

These are tasks a SUSE Linux administrator in an enterprise environment routinely has to deal with.

Audience

The primary audience for this course are those who completed *SUSE Linux Enterprise Server 10 Fundamentals* (Course 3071), or those with comparable knowledge.

Certification and Prerequisites

This course helps to prepare for the Novell Certified Linux Professional 10 (CLP 10) Practicum Exam, called the *Practicum*. The Novell CLP 10 is a prerequisite for the higher level certification Novell CLE 10 Practicum.

As with all Novell certifications, course work is recommended. To achieve the certification, you are required to pass the Novell CLP 10 Practicum (050-697).

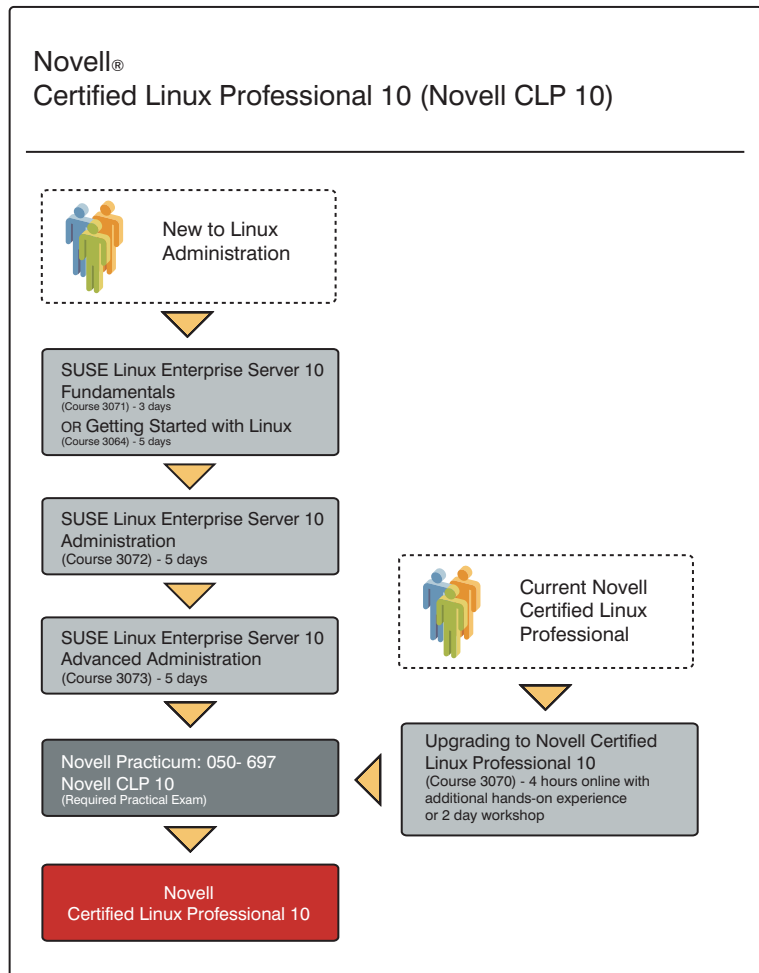
The Novell CLP 10 Practicum is a hands-on, scenario-based exam where you apply the knowledge you have learned to solve real-life problems—demonstrating that you know what to do and how to do it.

The practicum tests you on objectives of this course and those covered in:

- *SUSE Linux Enterprise Server Fundamentals* (Course 3071)
- *SUSE Linux Enterprise Server Advanced Administration* (Course 3073)

The following illustrates the training and testing path for Novell CLP 10:

Figure Intro-1





For more information about Novell certification programs and taking the Novell CLP 10 and CLE 10 Practicum exam, see <http://www.novell.com/training/certinfo/>, <http://www.novell.com/training/certinfo/clp10>, and <http://www.novell.com/training/certinfo/cle10>.

Before attending this course, you should have attended the courses:

- *SUSE Linux Enterprise Server 10 Fundamentals* (Course 3071)

SUSE Linux Enterprise Server 10 Support and Maintenance

The copy of SUSE Linux Enterprise Server 10 you received in your student kit is a fully functioning copy of the SUSE Linux Enterprise Server 10 product.

However, to receive official support and maintenance updates, you need to do one of the following:

- Register for a free registration/serial code that provides you with 30 days of support and maintenance.
- Purchase a copy of SUSE Linux Enterprise Server 10 from Novell (or an authorized dealer).

You can obtain your free 30-day support and maintenance code at <http://www.novell.com/products/linuxenterpriseserver/>.



You will need to have or create a Novell login account to access the 30-day evaluation.

Novell Customer Center

Novell Customer Center is an intuitive, web-based interface that helps you to manage your business and technical interactions with Novell. Novell Customer Center consolidates access to information, tools and services such as:

- Automated registration for new SUSE Linux Enterprise products
- Patches and updates for all shipping Linux products from Novell
- Order history for all Novell products, subscriptions and services
- Entitlement visibility for new SUSE Linux Enterprise products
- Linux subscription-renewal status
- Subscription renewals via partners or Novell

For example, a company might have an administrator who needs to download SUSE Linux Enterprise software updates, a purchaser who wants to review the order history and an IT manager who has to reconcile licensing. With Novell Customer Center, the company can meet all these needs in one location and can give each user access rights appropriate to their roles.

You can access the Novell Customer Center at <http://www.novell.com/center>.

SUSE Linux Enterprise Server 10 Online Resources

Novell provides a variety of online resources to help you configure and implement SUSE Linux Enterprise Server 10.

These include the following:

- <http://www.novell.com/products/linuxenterpriseserver/>
This is the Novell home page for SUSE Linux Enterprise Server.
- <http://www.novell.com/documentation/sles10/index.html>
This is the Novell Documentation web site for SLES 10.
- <http://support.novell.com/linux/>
This is the home page for all Novell Linux support, and includes links to support options such as the Knowledgebase, downloads, and FAQs.
- <http://www.novell.com/coolsolutions/>
This Novell web site provides the latest implementation guidelines and suggestions from Novell on a variety of products, including SUSE Linux.

Agenda

The following is the agenda for this 5-day course:

TableIntro-1

	Section	Duration
Day 1	Introduction	00:30
	Section 1: Install SUSE Linux Enterprise Server 10	02:30
	Section 2: Administer the Linux File System	03:00
Day 2	Section 2: Administer the Linux File System (contd.)	01:00
	Section 3: Administer User Access and Security	03:30
	Section 4: Configure the Network Manually	02:00
Day 3	Section 5: Administer Linux Processes and Services	02:00
	Section 6: Monitor SUSE Linux Enterprise Server	02:00
	Section 7: Configure System Initialization	02:30
Day 4	Section 7: Configure System Initialization (contd.)	03:00
	Section 8: Manage Software for SUSE Linux Enterprise Server	01:30
	Section 9: Manage Backup and Recovery	02:00
Day 5	Section 10: Administer Printing	03:00
	Section 11: Configure Remote Access	02:00

Scenario

The IT department of Digital Airlines is rolling out more and more SUSE Linux Enterprise Server 10 installations. Your task is to familiarize yourself with SLES 10 to be able to take on more and more system administrator tasks on this platform.

You need additional experience in the following areas:

- Installation and configuration of SLES 10
- File system maintenance
- Specialized aspects of User Management, like POSIX ACLs
- Manual network configuration and fundamental network services
- Software management
- Printing
- Management of services and processes
- Remote administration

You decide to set up test servers in the lab to enhance your skills in these areas.

Exercises

The exercises in this course consist of a description of the exercise, and step-by-step instructions on how to complete the task.

You should first try to complete the task described on your own, based on what is covered in the manual in the respective section. Resort to the step-by-step instruction only if you feel unable to complete the task or to find out if what you did was correct.

Exercise Conventions

When working through an exercise, you will see conventions that indicate information you need to enter that is specific to your server.

The following describes the most common conventions:

- ***italicized/bolded text***. This is a reference to your unique situation, such as the host name of your server.

For example, if the host name of your server is `da10`, and you see the following:

hostname.digitalairlines.com

you would enter

da10.digitalairlines.com

- ***10.0.0.xx***. This is the IP address that is assigned to your SLES 10 server.

For example, if your IP address is `10.0.0.10`, and you see the following:

10.0.0.xx

you would enter

10.0.0.10

- **Select**. The word *select* is used in exercise steps to indicate a variety of actions including clicking a button on the interface and selecting a menu item.
- **Enter and Type**. The words *enter* and *type* have distinct meanings.

The word *enter* means to type text in a field or at a command line and press the Enter key when necessary. The word *type* means to type text without pressing the Enter key.

If you are directed to type a value, make sure you do not press the Enter key or you might activate a process that you are not ready to start.

SECTION 1 Install SUSE Linux Enterprise Server 10

YaST (Yet another Setup Tool) provides options that make installation simple and quick.

However, you also need to understand the more advanced installation options available. By changing installation mode, partitioning, software selection, authentication method, or hardware setup, you can install servers that meet a variety of needs.

In this section, you install SUSE Linux Enterprise Server 10 (SLES 10). You also learn how to use advanced installation options and to troubleshoot the installation process.

Objectives

1. Perform a SLES 10 Installation
2. Configure the SLES 10 Installation
3. Troubleshoot the Installation Process

Objective 1 **Perform a SLES 10 Installation**

Installing SLES 10 consists of a base installation phase and a configuration phase.

To perform the base installation do the following:

- Boot From the Installation Media
- Select the System Language
- Select the Installation Mode
- Set the Clock and Time Zone
- Understand and Change the Installation Settings
- Verify Partitioning
- Select Software
- Start the Installation Process

Boot From the Installation Media

To start the installation process, insert the *SUSE Linux Enterprise Server* Product DVD into the DVD drive and then reboot the computer to start the installation program.

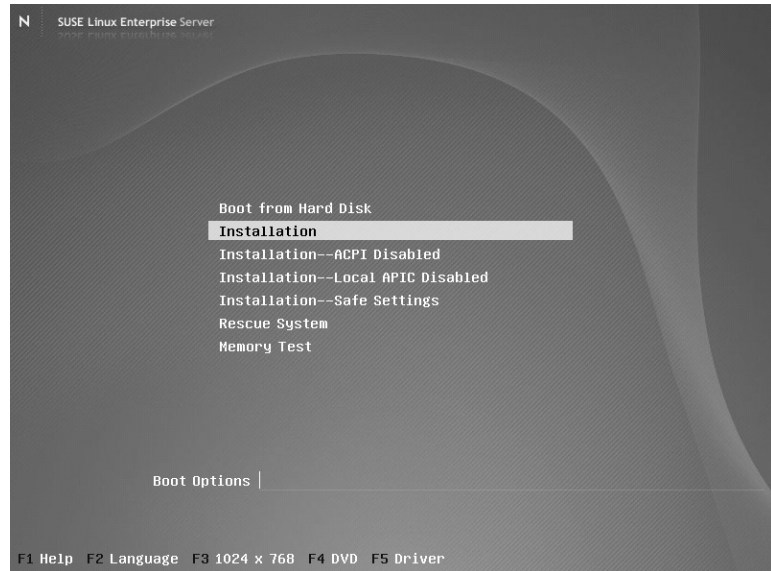


To start the installation program, your computer needs to be configured to start from a DVD drive. You might need to change the boot drive order in the BIOS setup of your system to boot from the drive.

Consult the manual shipped with your hardware for further information.

When your system has started from the installation CD, the following appears:

Figure 1-1



You can use the arrow keys to select one of the following options:

- **Boot from Hard Disk.** Boots the system installed on the hard disk (the system normally booted when the machine is started). This is the default option.
- **Installation.** Starts the normal installation process. All modern hardware functions are enabled.
- **Installation - ACPI Disabled.** Starts the installation process with ACPI (Advanced Configuration and Power Interface) disabled. If the normal installation fails, the reason might be that the system hardware does not support ACPI. In this case, you can use this option to install without ACPI support.

- **Installation - Local APIC Disabled.** Starts the installation process with local APIC (Advanced Programmable Interrupt Controller) disabled.
- **Installation - Safe Settings.** Starts the installation process with the DMA (Direct Memory Access) mode and any interfering power management functions disabled. Use this option if the installation fails with the other options.
- **Rescue System.** Starts the SLES 10 rescue system. If you cannot boot your installed Linux system, you can boot the computer from the DVD (or the first CD if you are using a CD set) and select this option. This starts a minimal Linux system without a graphical user interface to allow experts to access disk partitions for troubleshooting and repairing an installed system.
- **Memory Test.** Starts a memory testing program, which tests system RAM by using repeated read and write cycles. This is done in an endless loop, because memory corruption often shows up sporadically and many read and write cycles might be necessary to detect it.

If you suspect that your RAM might be defective, start this test and let it run for several hours. If no errors are detected, you can assume that the memory is intact. Terminate the test by rebooting the system.

Use the function keys, as indicated in the bar at the bottom of the screen, to change a number of installation settings:

- **F1.** Opens context-sensitive help for the currently selected option of the boot screen.
- **F2.** Select a installation language.
- **F3.** Select a graphical display mode (such as 640x480 or 1024X768) for the installation. You can select one of these, or select text mode, which is useful if the graphical modes cause display problems.

- **F4.** Select an installation media type. Normally, you install from the inserted installation disk, but in some cases you might want to select another source, such as FTP or NFS.
- **F5.** Add a driver update CD to the installation process. You are asked to insert the update disk at the appropriate point in the installation process.

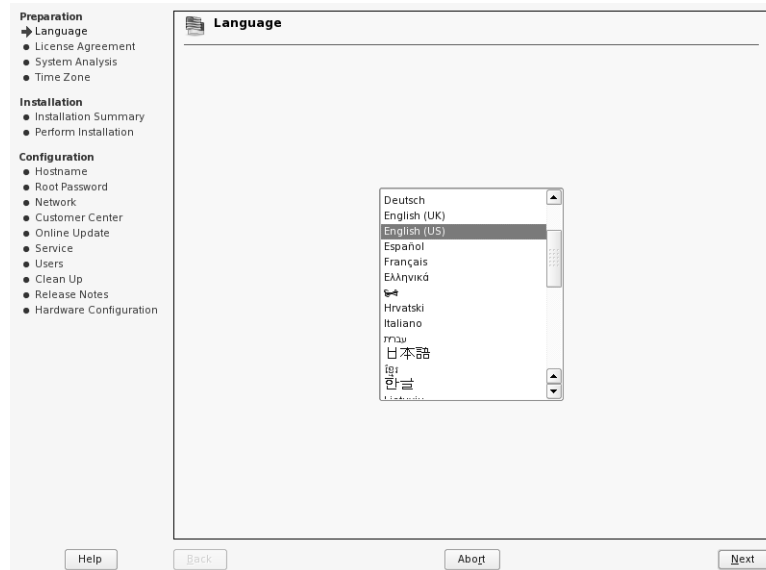
Select the **Installation** option to start the installation process. If the installation fails for some reason, try to install with the options **Installation - ACPI Disabled**, **Installation - Local APIC Disabled**, or **Installation - Safe Settings**.

After you select an installation option, a minimal Linux system loads to run the YaST installation program.

Select the System Language

After YaST starts, the following appears:

Figure 1-2



Almost all YaST installation dialogs use the same format:

- The left side displays an overview of the installation status.
- From the lower left side, you can select a help button to get information about the current installation step.
- The right side displays the current installation step.
- The lower right side provides buttons for navigating to the previous or next installation steps or for aborting the installation.



If the installation program does not detect your mouse, you can use the Tab key to navigate through the dialog elements, the arrow keys to scroll in lists and Enter to select buttons. You can change the mouse settings later in the installation process.

From the language dialog, select the language of your choice, and then select **Next** to continue to the next step, the License Agreement.

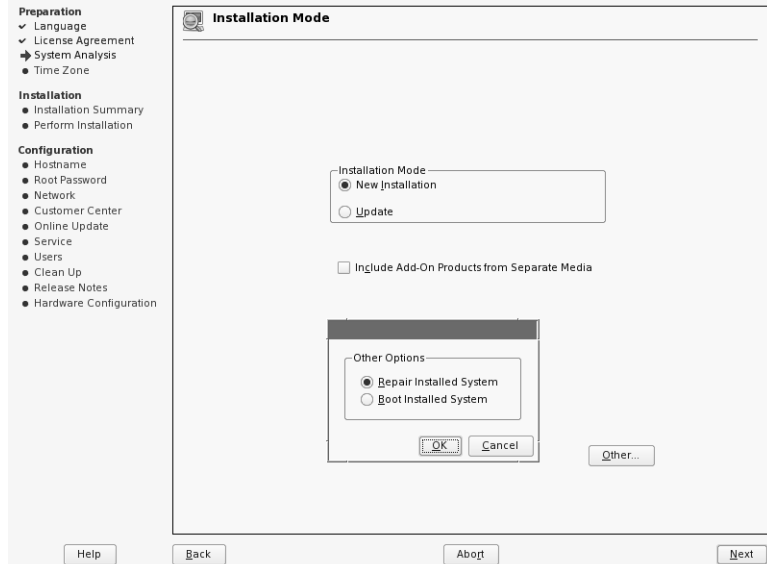
You have to select **Yes, I Agree to the License Agreement** to get to the next step by selecting **Next**.

Select the Installation Mode

If there is no operating system installed on your computer, the installation mode dialog offers only **New Installation**. (Update and Other cannot be selected in this case.)

If YaST detects another SUSE Linux installation, you are offered more options, some of which are only available after selecting **Other**, like in the following:

Figure 1-3



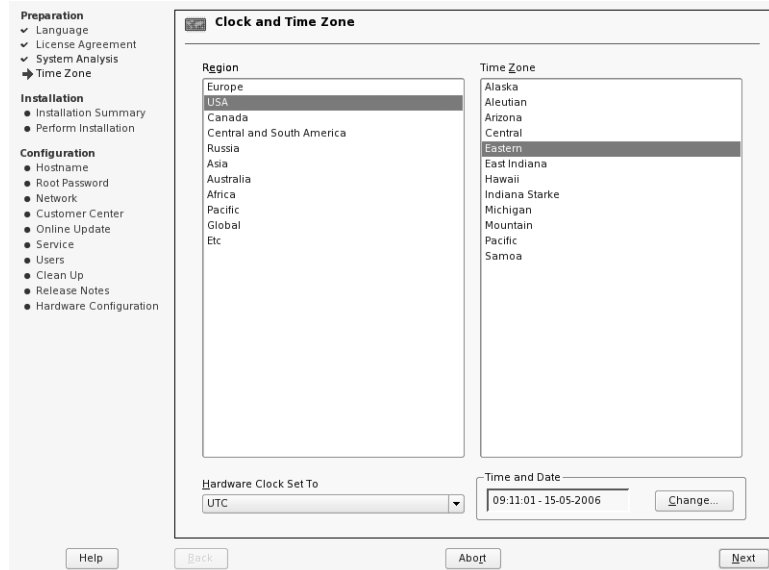
- **New installation.** Performs a normal new installation of SLES 10. This is the default option.
- **Update.** Updates a previously installed SLES 9 installation.
- **Other.** Offers two more options:
 - **Repair Installed System.** Repairs a previously installed SLES 10 installation.
 - **Boot Installed System.** Boots a previously installed Linux installation.
- **Abort Installation.** Terminates the installation process.

For a normal installation, select **New Installation** and then select **Next** to proceed to the next step.

Set the Clock and Time Zone

YaST selects the time zone of the installed system according to your language selection. Change the time zone if you are located in a different one.

Figure 1-4

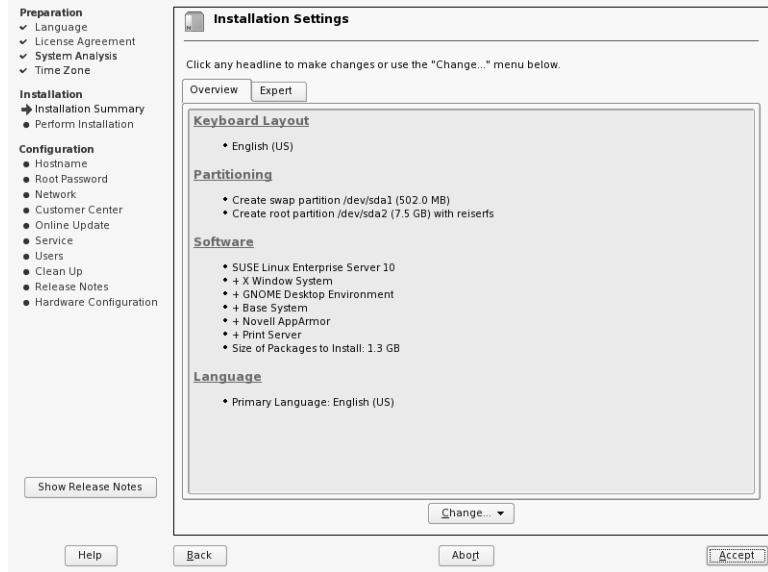


If your hardware clock is set to UTC (Universal Time Coordinated) the system time is set according to your time zone and automatically adjusted to daylight saving time. If your hardware clock is set to local time, select Local Time instead of UTC in the drop-down menu.

Understand and Change the Installation Settings

YaST analyzes the system and creates an installation proposal. The proposed settings are displayed on two tabs, as in the following figure; **Overview** shows the main categories:

Figure 1-5



The proposal displays installation settings that are necessary for a base installation. You can change these settings by selecting the following headings:

- **Keyboard layout.** Changes the keyboard layout. YaST selects the keyboard layout according to your language settings. Change the keyboard settings if you prefer a different layout.
- **Partitioning.** Changes the hard drive partitioning. If the automatically generated partitioning scheme does not fit your needs, you can change it by selecting this headline.
- **Software.** Changes the software selection. You can select or deselect software.

- **Language.** Changes the default language.

The Experts tab shows the above options, plus the following:

- **System.** Restarts the hardware detection process and displays a list of all available hardware components. You can change the PCI-ID setup, select single components and view details, or save the list to a file.
- **Add-on Products.** Choose this option to include any add-on products.
- **Booting.** Select this if you want to change any boot loader settings or use Lilo (Linux Loader) instead of GRUB (Grand Unified Bootloader) as boot loader.
- **Time zone.** Opens the Clock and Time Zone dialog described earlier.
- **Default Runlevel.** Changes the runlevel. If a graphical environment is installed, the default is runlevel 5, otherwise it is 3.

Of the settings described above, partitioning and software will be discussed in more detail.

Verify Partitioning

In most cases, YaST proposes a reasonable partitioning scheme that you can accept without change. However, you might need to change the partitioning manually if

- You want to optimize the partitioning scheme for a special purpose server (such as a file server).
- You want to configure LVM (Logical Volume Manager).
- You have more than one hard drive and want to configure RAID (Redundant Array of Independent Disks).

- You want to delete existing operating systems so you have more space available for your SLES 10 installation.

To partition the hard drive manually, you need to know the following:

- The Basics of Hard Drive Partitioning
- The Basic Linux Partitioning Scheme
- How to Change YaST's Partitioning Proposal
- Use the YaST Expert Partitioner

The Basics of Hard Drive Partitioning

Partitions divide the available space of a hard drive into smaller portions. This lets you install more than one operating system on a hard drive or use different areas for programs and data.

Every hard disk (on an Intel platform) has a partition table with space for four entries. An entry in the partition table can correspond to a primary partition or an extended partition. However, only one extended partition entry is allowed.

A *primary partition* consists of a continuous range of cylinders (physical disk areas) assigned to a particular file system. If you use only primary partitions, you are limited to four partitions per hard disk (because the partition table can only hold four primary partitions).

This is why extended partitions are used. *Extended partitions* are also continuous ranges of disk cylinders, but can be subdivided into logical partitions. *Logical partitions* do not require entries in the main 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 instead of a fourth primary partition. This extended partition should include the entire remaining free cylinder range. Then create multiple logical partitions within the extended partition. The maximum number of partitions is 15 on SCSI disks and 63 on (E)IDE disks.

It does not matter which type of partitions you use on Linux systems; primary and logical partitions both work well.

The Basic Linux Partitioning Scheme

The optimal partitioning scheme for a server depends on the purpose of the server.

A SLES 10 installation needs at least two partitions:

- **Swap partition.** This partition is used by Linux to move unused data from the main memory to the hard drive, thus freeing main memory which then can be used by other processes.
- **Root partition.** This is the partition that holds the top (/) of the file system hierarchy, the so-called root directory.

No matter what partition scheme you choose, you always need at least one swap partition and a root partition.

The following guidelines help you determine what you can install depending on the space available on your hard disk for your file system:

- **800 MB.** This allows for a minimal installation with no graphical interface. With this configuration, you can only use console applications.
- **1300 MB.** This allows for an installation with a minimum graphical interface. This includes the X Window system and a few graphical applications.

- **2 GB.** This holds the default installation proposed by YaST. This configuration includes a modern desktop environment (such as KDE or GNOME), and provides enough space for several additional applications.
- **4 GB.** This allows for a full installation, including all software packages shipped with SLES 10.

You can put certain directories on separate partitions. If you do this, your root partition can be smaller than outlined above. Any space for data needs to be added to the above.



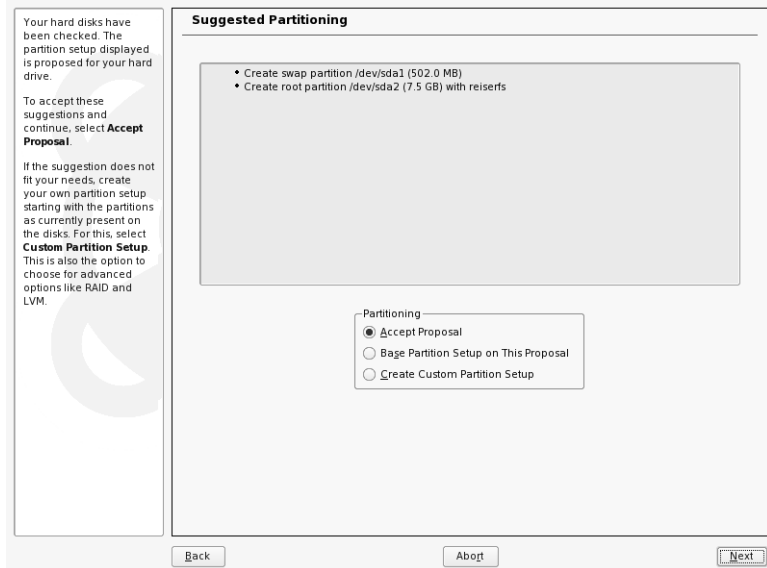
As today's computers are equipped with hard disks with capacities of 100 GB and more, there is still plenty of space for data. Considering the difficulties involved with changing partitions in an installed system and the size of current hard disks, you should therefore allocate much more space than the above minimum when deciding on the hard disk layout.

Partitions and partitioning schemes will be covered more extensively in the objective "Configure Linux File System Partitions" on page 2-16.

How to Change YaST's Partitioning Proposal

To use YaST to change the partition scheme, select the **Partitioning** headline in the installation proposal. The following appears:

Figure 1-6

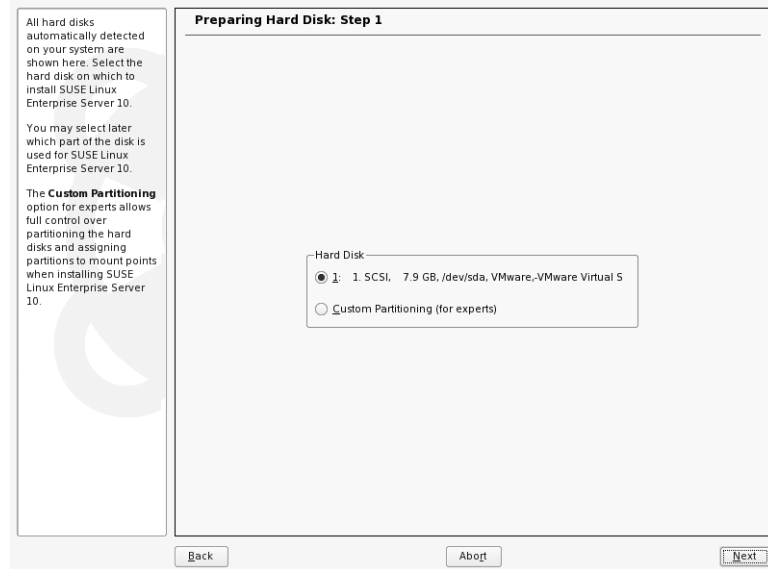


In the top part of the dialog, YaST displays the automatically generated partitioning proposal. The lower part of the dialog provides the following options:

- **Accept Proposal.** Accepts the partitioning scheme and returns to the main installation proposal.
- **Base Partition Setup on This Proposal.** Starts the YaST Expert Partitioner, using the partition proposal as base setup.

■ **Create Custom Partition Setup.** Displays the following:

Figure 1-7



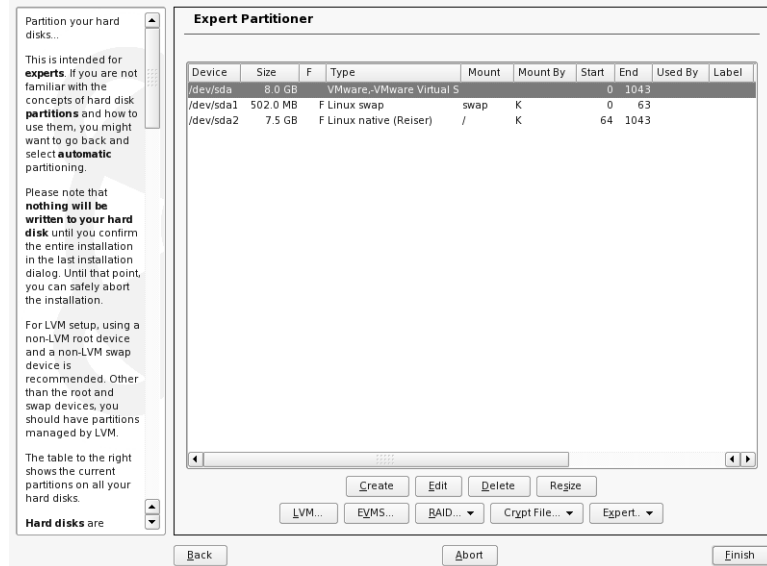
In this dialog, you can select

- ❑ A hard disk; selecting Next opens a dialog where you can choose to use the entire hard disk or some of the existing partitions for the installation of SLES 10.
- ❑ **Custom Partitioning**; selecting Next opens the YaST Expert Partitioner, displaying the existing partition layout.

Use the YaST Expert Partitioner

When you start the YaST Expert Partitioner, the following appears:

Figure 1-8



In the top part of the dialog, YaST lists details of the current partition setup. Depending on your previous choice, the list contains the partitioning proposal created by YaST or the current physical disk setup.

The buttons in the lower part of the dialog are used to create, edit, delete, and resize partitions, as well as to administer LVM (Logical Volume Manager), EVMS (Enterprise Volume Management System), RAID (Redundant Array of Independent Disks).



The changes made with the YaST Expert Partitioner are not written to disk until the installation process is started. You can always discard your changes by selecting **Back** or you can restart the Expert Partitioner to make more changes.

The following entries are displayed for every hard disk in your system:

- One entry for the hard disk itself, which has the corresponding device name in the Device column (such as **/dev/sda**).
- One entry for every partition on the hard disk with the corresponding device name and the partition number in the Device column (such as **/dev/sda1**).

Each entry in the list includes information in the following columns:

- **Device.** Displays the device name of the hard disk or the partition.
- **Size.** Displays the size for the hard disk or partition.
- **F.** When the character “F” is displayed in this column, the partition will be formatted during the installation process.
- **Type.** Displays the partition or hard disk type. Depending on the operating system and the architecture, partitions can have various types, like Linux native, Linux swap, Win95 FAT 32, NTFS, etc.
- **Mount.** Displays the mount point of a partition. For swap partitions, the keyword *swap* is used instead.
- **Mount By.** Indicates how the file system is mounted: K—Kernel Name, L—Label, U—UUID, I—Device ID, and P—Device Path.
- **Start.** Displays the start cylinder of a hard disk or partition. Hard disk entries always start with 0.
- **End.** Displays the end cylinder of a hard disk or partition.
- **Used By.** This column holds information about the system using this partition, like LVM-system.
- **Label, Device ID, Device Path.** These columns list the respective information.

The buttons in the lower part of the dialog let you

- Create New Partitions
- Edit Existing Partitions
- Delete Existing Partitions
- Resize Existing Partitions
- Perform Expert Tasks

These administrative tasks are covered in more detail below.

Managing LVM Volumes and Software Raid are covered in Section 2, “Administer the Linux File System” on page 2-1. EVMS (<http://evms.sourceforge.net/>) and Crypt File Partitions are not covered in this course.

Create New Partitions

Create a new partition by selecting **Create**. A dialog with one of the following options appears (the options you see depend on your hard disk setup):

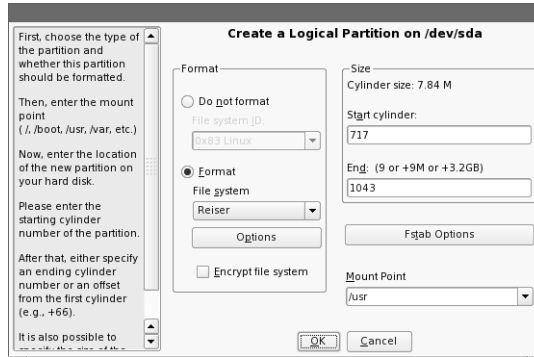
- If you have more than one disk in your system, you are asked to select a disk for the new partition first.
- If you do not have an extended partition, you are asked if you want to create a primary or an extended partition.
- If you have an extended partition, and there is space on the hard drive outside the extended partition for additional primary partitions, you are asked if you want to create a primary or a logical partition.
- If you have 3 primary partitions and an extended partition, you can only create logical partitions.



You need enough space on your hard disk to create a new partition. You learn later in this section how to delete existing partitions to free used disk space.

If you choose to create a primary or a logical partition, the following appears:

Figure 1-9



This dialog provides the following options:

- **Format.** This lets you choose one of the following options:
 - **Do not format.** Do not format the newly created partition. No file system will be created on this new partition. You can select the partition type in the drop-down list.
 - **Format.** Formats the new partition with the file system you select from the File System drop-down list.

You can choose from the following file systems:

- **Ext2.** Formats the partition with the Ext2 file system. Ext2 is an old and proven file system, but it does not include journaling.
- **Ext3.** Formats the partition with the Ext3 file system. Ext3 is the successor of Ext2 and offers a journaling feature.
- **Reiser.** Formats the partition with ReiserFS, a modern journaling file system. (This is the default option.)

- ❑ **FAT.** Formats the partition with the FAT file system. FAT is an older file system used in DOS and Windows. You can use this option to create a data partition, which is accessible from Windows and Linux. You must not create a root partition with this file system.
- ❑ **XFS.** Formats the partition with XFS, a journaling file system originally developed by SGI.
- ❑ **Swap.** Formats the partition as a swap partition.

If you are not sure which file system to choose, select Reiser for root and data partitions and Swap for swap partitions.

Journaling is explained in more detail in Section 2, “Administer the Linux File System” on page 2-1.

- ❑ **Options.** By selecting Options, you can change parameters for the file system you selected. You can use the default parameters in most cases.
- ❑ **Encrypt file system.** If you select this option, the partition with the file system is encrypted. Encrypting a file system prevents unauthorized mounting only; once mounted the files are accessible like any other file on the system.

You should only use this option for non-system partitions such as user home directories.

- **Size.** Lets you configure the size of the new partition with the following:

- ❑ **Start Cylinder.** Determines the first cylinder of the new partition. YaST normally preselects the first available free cylinder of the hard disk.
- ❑ **End.** Determines the size of the new partition. YaST normally preselects the last available free cylinder.

To configure the end cylinder, do one of the following:

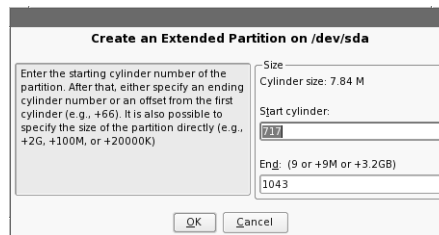
- ❑ Enter the cylinder number.

- ❑ Enter a plus sign (+) followed by the amount of disk space for the new partition. Use M for MB and G for GB. YaST calculates the last cylinder number. For example, enter **+5G** for a partition size of 5 GB.
- **Fstab Options.** Select this option to edit the fstab entry for this partition. The default setting should work in most cases.
- **Mount Point.** Select the mount point of the new partition from this drop-down list. You can also enter a mount point manually, if it's not available in the list. The mount point will be created automatically during installation.

After changing the parameters, select **OK** to add the new partition to the partition list.

If you chose to create an extended partition, the following appears:

Figure 1-10



You can enter the following:

- **Start cylinder.** The start cylinder determines the first cylinder of the new partition. YaST normally preselects the first available free cylinder of the hard disk.
- **End.** The end cylinder determines the size of the new partition. YaST normally preselects the last available cylinder of the hard disk.

To configure the end cylinder, do one of the following:

- ❑ Enter the cylinder number.

- Enter a plus sign (+) followed by the amount of disk space for the new partition. Use M for MB and G for GB. YaST calculates the last cylinder number.

For example, enter **+5G** for a partition size of 5 GB.

After entering the size, select **OK** to add the new extended partition to the partition list.

Edit Existing Partitions

Select a partition from the list and select **Edit**. You can edit only primary and logical partitions with the Expert Partitioner. You cannot edit extended partitions or the entry for the entire hard disk.

If you edit a primary or logical partition, a dialog appears which is very similar to the Create Partition dialog described above. You can change all options except the partition size.

After changing the partition parameters, select **OK** to save your changes to the partition list.

Delete Existing Partitions

To delete a partition, select a partition from the list, select **Delete**, and then select **Yes** in the confirmation dialog. The partition is deleted from the partition list.

Remember that you also delete all logical partitions when you delete an extended partition.

If you select the entry for the entire hard disk and select **Delete**, all partitions on the disk are deleted.

Resize Existing Partitions

Select a partition from the list and select **Resize**.



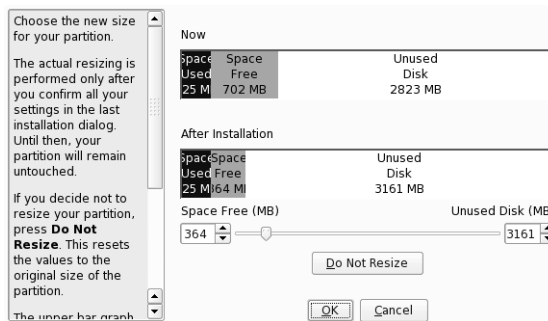
Although you can reduce a partition's size without deleting it to increase free space on the hard disk, you should always back up the data on the partition before resizing it.



If the selected partitions are formatted with the FAT or NTFS file system, there are certain steps you should take in Windows before resizing (scandisk and defrag). See the section on installation in the *SUSE Linux Enterprise Server 10 Administration Manual* (/usr/share/doc/manual/sles-admin_en/, package sles-admin_en) for details.

After you select **Resize**, the following appears:

Figure 1-11



This dialog includes the following:

- Two bars representing the partition before and after the resizing process
 - **Now.** Used space is designated by dark blue and the available space by light blue. If there is space not assigned to a partition it is designated by white.
 - **After installation.** Used space is designated by dark blue and the free space by light blue. The space that is available for a new partition is designated by white.
- A slider to change the size of the partition

- Two text fields that display the amount of free space on the partition being resized and the space available for a new partition after the resizing process
- A Do Not Resize button used to reset the partition to the original size

To resize the partition, move the slider until enough unused disk space is available for a new partition. When you select **OK**, the partition size changes in the partition list.

Perform Expert Tasks

When you select **Expert**, the following options are available:

- **Reread the Partition Table.** Resets the partition list to the current physical disk setup. All changes will be lost.
- **Import Mount Points from Existing /etc/fstab.** Scans the hard disks for an /etc/fstab file. You can load this file and set the mount points accordingly.
- **Delete Partition Table and Disk Label.** Deletes the partition table and the disk label of the selected hard disk. *All data on that disk will be lost.*

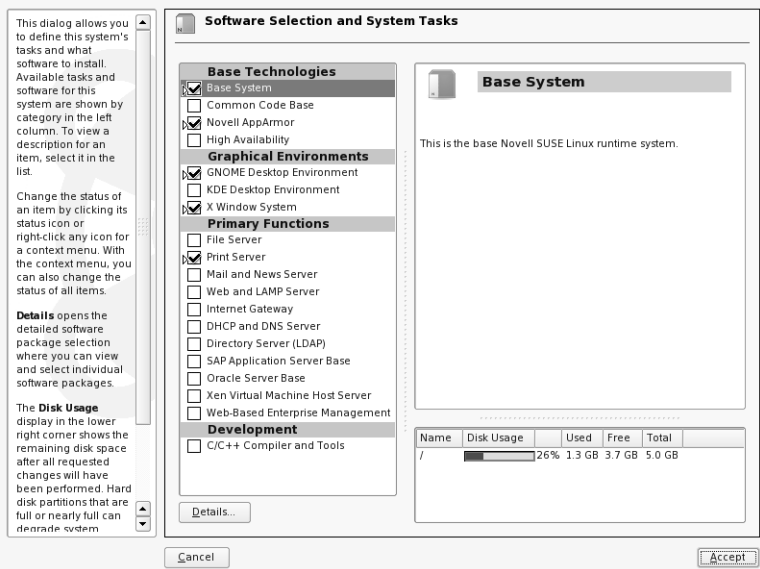
When you finish configuring settings in the Expert Partitioner, return to the installation proposal by selecting **Finish**.

Select Software

SLES 10 contains many software packages for various application purposes. Instead of selecting needed packages one by one, you can select various software categories.

Depending on the available disk space, YaST preselects several of these categories. Selecting **Software** in the installation overview opens the following dialog:

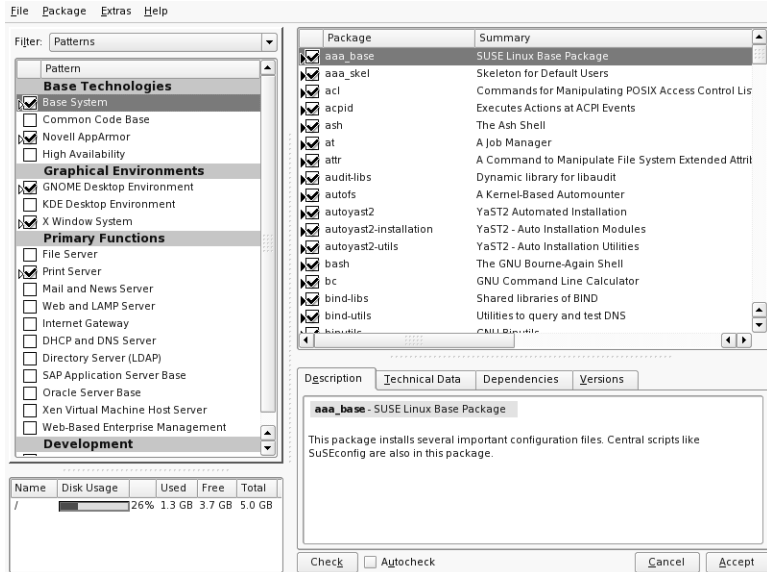
Figure 1-12



The figure above shows the default selection. A brief description appears on the right when you highlight a category in the center column.

To find out which packages are contained in the various categories, select Details, which opens the following dialog:

Figure 1-13



Selecting one pattern on the left shows the software packages contained in that category on the right. Selecting the square to the left of the pattern selects it for installation or deselects it.

A package typically contains an application and all additional files required to use the software. Sometimes larger applications can be split into multiple packages and several small applications can be bundled into a single package. SUSE Linux Enterprise Software uses the RPM Package Manager for software management.

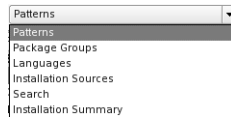
Sometimes one software package needs another one to run. Information on these dependencies is stored in the RPM packages. YaST can automatically select software packages when another package requires them.

You can install a package by selecting the check box for that package in the package list on the right.

To view details for a package, highlight its entry in the package list. The details for the currently selected package are displayed below the package list.

The Filter drop-down menu offers different views on the software packages available and the software scheduled for installation.

Figure 1-14



- **Patterns.** This leads to the dialog shown in Figure 1-13.
- **Package Groups.** Displays the packages in a hierarchical tree view. There are main categories, like Productivity, Programming, System, Hardware, etc. and subcategories. Selecting a category on the left displays the software packages belonging to that category on the right.
- **Languages.** You can select support for additional languages.
- **Installation Sources.** Displays the installation sources configured.
- **Search.** Displays a search dialog to search for packages.
- **Installation Summary.** Displays a summary of the packages selected for installation.

The disk usage of the software packages selected for installation is displayed in the lower left corner of the dialog.

Select the option **Check** to check the dependencies of the selected packages. This check is also done when you confirm the package selection dialog.

If the check box **Autocheck** is selected, dependencies are checked every time you select or deselect a package.

Confirm your package selection and return to the installation proposal by selecting **Accept**.

Start the Installation Process

After customizing the installation proposal, select **Accept**. A dialog appears asking you to confirm the proposal. Start the installation process by selecting **Install**; return to the installation proposal by selecting **Back**.

Before installing software packages, YaST changes the hard disk partitioning.

Depending on your software selection and the performance of your system, the installation process takes 15–45 minutes.

If you are using the product CD set instead of the DVD, YaST asks you to change the installation CDs. Insert the requested CD and continue the installation by selecting **OK**.

After all software packages are installed, YaST reboots the computer and prompts you for the hostname, root password, network configuration details, etc., to further customize your installation.

Objective 2 **Configure the SLES 10 Installation**

In this part of the installation process, you use YaST to perform the following configuration tasks:

- Set the Hostname
- Set the root Password
- Configure the Network
- Test the Internet Connection
- Novell Customer Center Configuration and Online Update
- Manage Users
- Configure Network Services
- Configure Hardware
- Finalize the Installation Process

Set the Hostname

YaST suggests a hostname **linux-xxxx**, with xxxx being composed of random characters. The domain defaults to **site**. Change the hostname and the domain name to the correct values for the computer and remove the check mark in front of **Change Hostname via DHCP**.

If the computer gets its hostname and domain via DHCP you do not need to change anything in this dialog.

Set the root Password

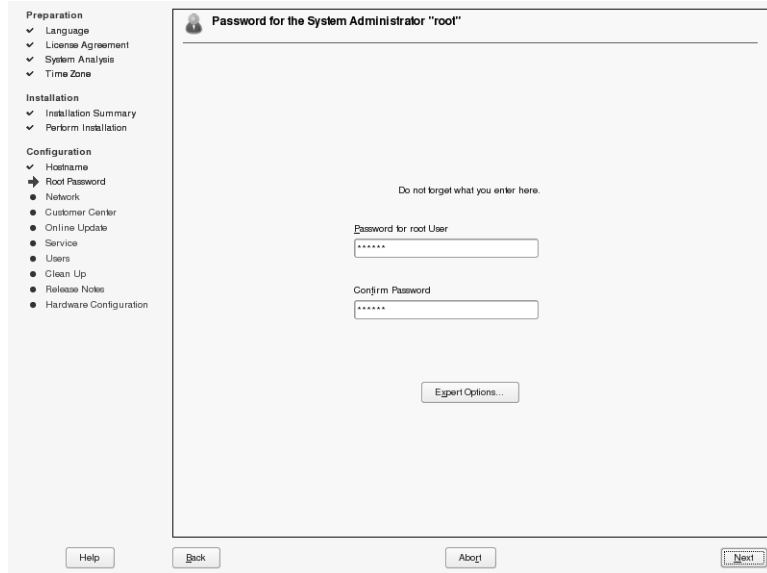
root is the name of the administrator of the system. Unlike regular users, who might not have permission to do certain things on the system, root has unlimited power to do anything, including the following:

- Access every file and device in the system.
- Change the system configuration.
- Install software.
- Set up hardware.

The root account should only be used for system administration, maintenance, and repair. Logging in as root for daily work is risky: a single mistake can lead to irretrievable loss of many system files.

To let you set the root password during the installation process, YaST displays the following:

Figure 1-15



Enter the same password in both text fields of the dialog.

You should choose a password that cannot be guessed easily. Use numbers, lowercase and uppercase characters to avoid dictionary attacks.

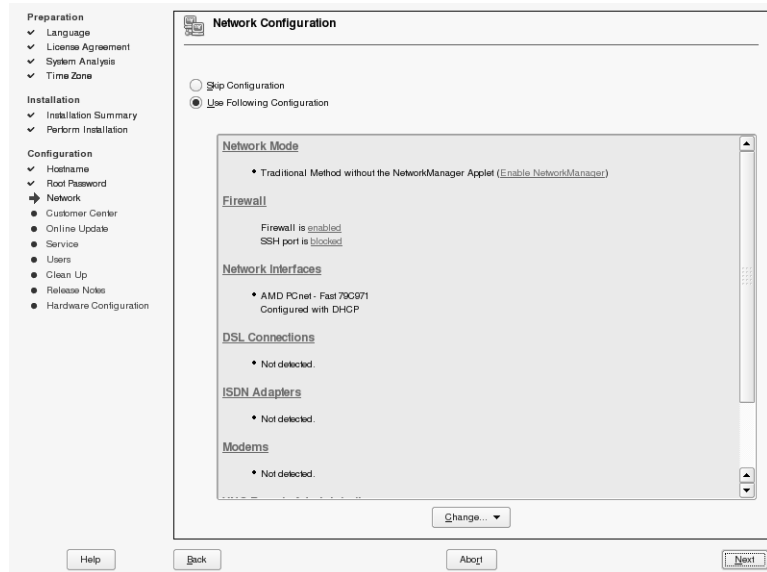
By selecting Expert Options, you can choose the password encryption algorithm. In most cases you can use the default setting, which is **Blowfish**.

After entering the root password, continue to the next configuration step by selecting **Next**. In case your password is too simple or weak, you are shown a warning. Go back to enter a better password, or accept the weakness and go on.

Configure the Network

To let you configure the network connection of your system, YaST displays the following:

Figure 1-16



In the top part of the dialog, you can choose one of the following options:

- **Skip Configuration.** Skip the network configuration for now. You can configure the network connection later in the installed system.
- **Use Following Configuration.** Use the network configuration proposal displayed in the area below.

The network configuration proposal is similar to the installation proposal at the beginning of the base installation, with headings that can be selected to view and configure further details. and includes the following entries:

- **Network Mode.** Switch between the traditional method of managing the network and network manager. On a server use the traditional method. Network manager is more suitable for a notebook, enabling users to easily switch between for instance Ethernet and wireless access.
- **Firewall.** If you want to administer the computer via SSH, toggle **SSH port is blocked** to **SSH port is open** by clicking on **blocked**. You can disable the firewall by clicking on **enabled**. When the firewall is disabled, then SSH is accessible as well.

Selecting **Firewall** itself opens a dialog allowing detailed configuration of the firewall settings.

- **Network Interfaces.** Displays the network interfaces found (such as Ethernet or a Wireless-LAN adapter) and their configuration (like DHCP).
- **DSL Connections.** Displays the configuration of DSL devices. These can be DSL modems connected with an Ethernet adapter or internal DSL modems.
- **ISDN Adapters.** Displays the configuration of ISDN devices.
- **Modems.** Displays the configuration of analog modems.
- **VNC Remote Administration.** Displays the configuration of remote administration using VNC.

CNI USE ONLY-1 HARDCOPY PERMITTED

- **Proxy.** Displays the HTTP and FTP proxy settings.

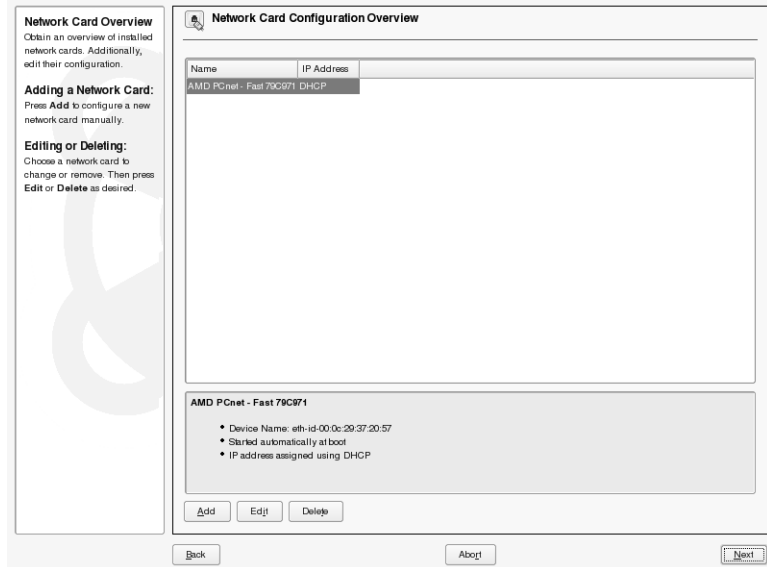
You can change a configuration by selecting the headline of the entry or by selecting the entry from the Change drop-down list. This menu also lets you reset all settings to the defaults generated by YaST.

If you are not sure which settings to use, stay with the defaults generated by YaST.

Configure Network Interfaces

After starting the network interface configuration, YaST displays the **Network Card Configuration Overview**. It lists all network cards, the configured ones as well as those which are not yet configured:

Figure 1-17



The upper part lists the cards found, the lower part show details for the network card that is highlighted in the upper part.

At this point, you can do one of the following:

- Add a Network Card Manually
- Edit an Existing Configuration

Add a Network Card Manually

If you want to configure a network card that was not automatically detected, select **Add** to display the following:

Figure 1-18

Here, set up your networking device. The values are written to `/etc/sysconfig/hardware/hw`.

Options for the module should be written in the format `option=value`. Each entry should be space separated, for example, `ip=0x300 irq=5`.

Note: If two cards are configured with the same module name, options will be merged while saving.

Get a list of available network cards by pressing **Select from List**.

If you have a **PCMCIA** network card, select PCMCIA. If you have a **USB** network card, select USB.

Manual Network Card Configuration

Network Configuration

Device Type: Ethernet Configuration Name: 0

Kernel Module

Hardware Configuration Name: static-0

Module Name: Options:

☐ PCMCIA ☐ USB

Select from List

Back Abort Next

From this dialog, you can configure the following:

- **Network Configuration.** Specify the network **Device Type** (Ethernet, Bluetooth, Wireless, etc.) and the device number.

- **Kernel Module.** If your network card is a PCMCIA or USB device, select the corresponding check boxes and confirm selecting **Next**.
- Otherwise, select **Select from List** and select your network card from the list. YaST automatically loads the appropriate driver for the selected card. Confirm by selecting **OK**.
- If you selected **Wireless** as Device Type for a WLAN card, **Next** brings you to a Network Address dialog. The default, DHCP, is usually the right choice. Selecting **Next** again opens a dialog where you can enter WLAN specific configuration parameters, like the **Operating Mode**, the **Network Name** (ESSID), the **Authentication Mode**, and the encryption key.

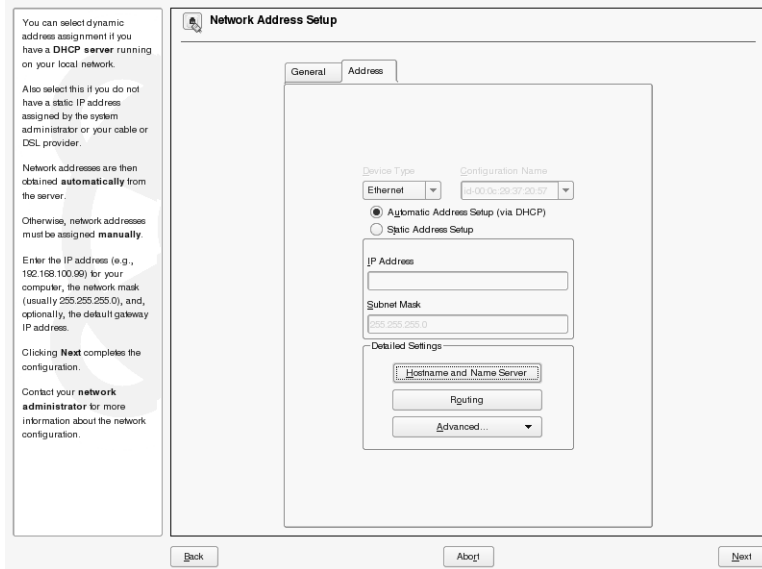
WEP keys are entered in a separate dialog after selecting WEP Keys. Expert settings concern parameters like the bit rate.

When you are finished with this dialog, select **Next**, which returns you to the Network Card Configuration Overview.

Edit an Existing Configuration

To edit a network card configuration, highlight its entry in the upper part of the **Network Card Configuration Overview** and select **Edit**.

Figure 1-19



The **Address** tab offers the following configuration options:

- **Automatic Address Setup (via DHCP).** If your network has a DHCP server, you can set up your network address automatically. You should also use this option if you are using a DSL line with no static IP address assigned by the ISP.

If you decide to use DHCP, you can configure the details after selecting **DHCP Options** from the **Advanced** drop-down list. Specify whether the DHCP server should always broadcast its responses (in this case select **Request Broadcast Response**) and any identifier to use.

By default, DHCP servers use the network card's hardware address to identify an interface. If you have a virtual host setup where different hosts communicate through the same interface, an identifier is necessary to distinguish them.

- **Static Address Setup.** If you have a static address, select the corresponding check box. Then enter the address and subnet mask for your network. The preset subnet mask should match the requirements of a typical home network.
- **Hostname and Name Server.** Select this option to set the host name and the name server manually.
- **Routing.** Select this option to configure routing manually.

The **General** tab offers the following configuration options:

- **Firewall Zone.** Decide whether this interface belongs to the Internal, External, or Demilitarized Zone, or if all traffic should be blocked (No Zone).
- **Device Activation.** Choose from At Boot Time, On Cable Connection, On Hotplug, Manually, or Never.
- **Detailed Network Interface Settings.** Specify the Maximum Transfer Unit (MTU), which sometimes improves the performance of certain DSL (Digital Subscriber Line) connections. For PPPoE (Point-to-Point over Ethernet) values between 1400 and 1492 are common; these values vary, depending on your ISP (Internet Service Provider).

Confirm the Network Address Setup and return to the Network Card Configuration Overview by selecting **Next**.

Delete an Existing Configuration

To delete an existing configuration, highlight it in the upper part of the Network Card Configuration Overview and select **Delete**.

When finished with adding, editing, or deleting network card configurations, save the network device configuration and return to the Network Configuration proposal by selecting **Next**.

After finishing the Network Configuration, select **Next**.

Test the Internet Connection

YaST then asks you to test your connection to the Internet. Select one of the following options:

- **Yes, Test Connection to the Internet.** YaST tries to test the Internet connection by downloading the latest release notes and checking for available updates.

If you select this option, the results are displayed on the next dialog.
- **No, Skip This Test.** Skip the connection test. If you skip the test, you can't update the system during installation.

Select one of the options and select **Next**.

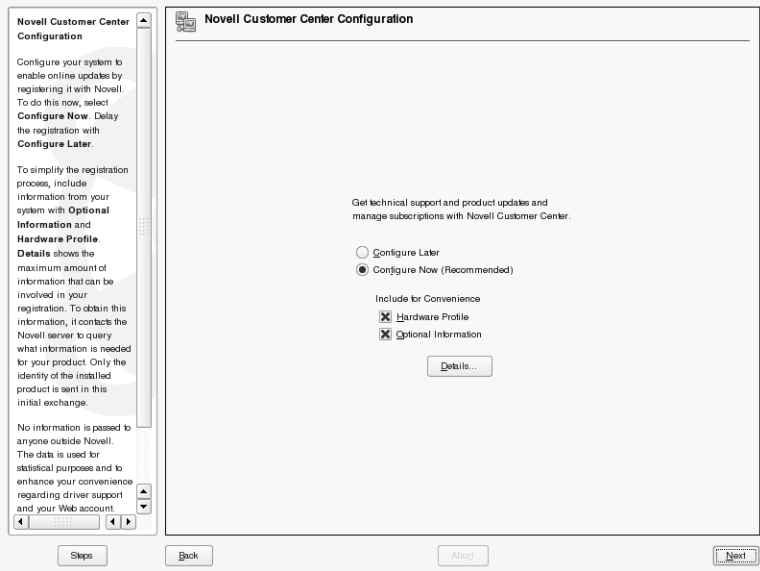
If the test fails, you can view the logs to find out what went wrong.

Select **Next** to continue.

Novell Customer Center Configuration and Online Update

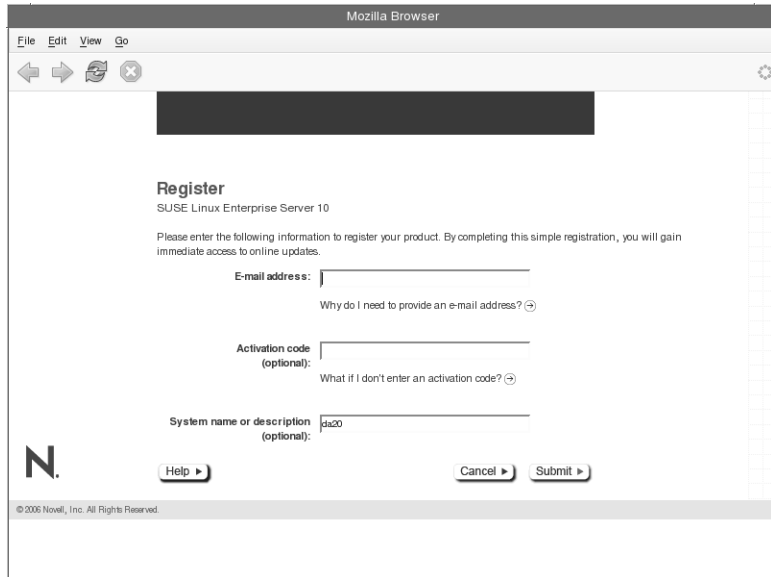
If the Internet connection test was successful, you can configure the Novell Customer Center, which is required to perform an online update. If there are any update packages available on the SUSE update servers, you can download and install them to fix known bugs or security issues.

Figure 1-20



Selecting **Next** starts a Browser and connects to the Novell web site, where all you have to enter is your e-mail address, and an activation code, if available.

Figure 1-21

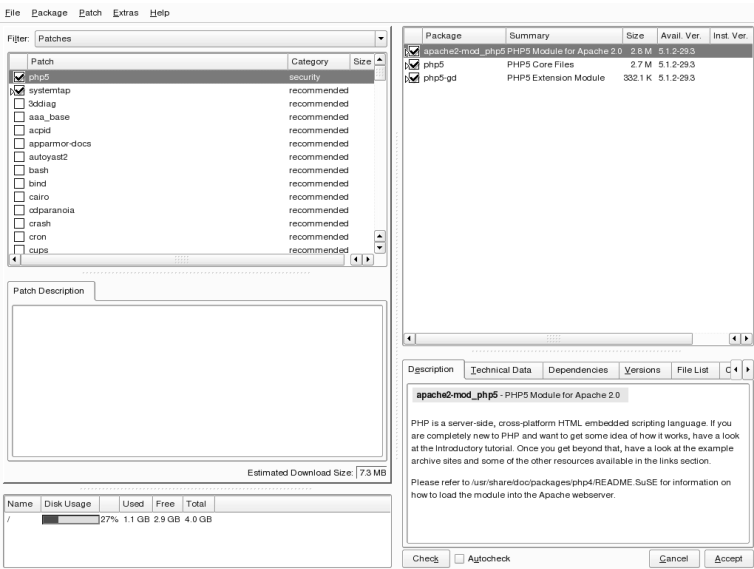


The screenshot shows a Mozilla Browser window displaying the registration page for SUSE Linux Enterprise Server 10. The page title is "Register" and the subtitle is "SUSE Linux Enterprise Server 10". The text on the page reads: "Please enter the following information to register your product. By completing this simple registration, you will gain immediate access to online updates." There are three input fields: "E-mail address:", "Activation code (optional):", and "System name or description (optional):". The "System name or description" field contains the text "ja20". Below the input fields are three buttons: "Help ►", "Cancel ►", and "Submit ►". The Novell logo is visible on the left side of the page. At the bottom, there is a copyright notice: "© 2006 Novell, Inc. All Rights Reserved."

After successful registration, the Online Update dialog opens. You can start the Online Update by selecting **Run Update** and **Next**. (You can also select **Skip Update** to perform the update later in the installed system.)

YaST's online update dialog opens up with a list of available patches (if any).

Figure 1-22



Select the patches you want to install, and then start the update process by selecting **Accept**.

Once the installation is complete, visit the Novell Customer Center at <http://www.novell.com/center/> to administer your Novell products and subscriptions.

Configure Network Services

In the next installation step, YaST displays the Service Installation Settings dialog.

In the top part of the dialog, you can choose one of the following options:

- **Skip Configuration.** Skip this configuration step. You can enable the services later in the installed system.
- **Use Following Configuration.** Use the automatically generated configuration displayed below this option or select one of the following headlines to change the configuration:

- **CA Management.** The purpose of a CA (certification authority or certificate authority) is to guarantee a trust relationship among all network services that communicate with each other.

If you decide that you do not want to establish a local CA, you must secure server communications using SSL (Secure Sockets Layer) and TLS (Transport Layer Security) with certificates from another CA.

By default, a CA is created and enabled during the installation.

To create proper certificates, the hostname has to be set correctly earlier in the Network Interface Configuration; otherwise the generated certificate will contain an incorrect hostname.

- **LDAP Server.** You can run an LDAP (Lightweight Directory Access Protocol) server on your host to have a central service managing a range of configuration settings. Typically, an LDAP server handles user account data, but since SLES 9 you can also use LDAP for mail, DHCP, and DNS related data on SUSE Linux Enterprise Server.

By default, an LDAP server is not set up during installation.

If you are not sure about the correct settings, keep the defaults generated by YaST. You can change the configuration later in the installed system.

When you are finished, select **Next**.

Manage Users

To manage users during this configuration step, do the following:

- Select the Authentication Method
- Configure the Authentication Method

Select the Authentication Method

The Authentication Method dialog offers four methods: You can selecting one of the following options:

- **Local (/etc/passwd).** Select this option to configure the system to use the traditional file-based authentication method.
- **LDAP.** If you have an LDAP server in your network, you can configure your system as an LDAP client.
- **NIS.** If you have a NIS server in your network, you can configure your system as a NIS client.
- **Windows Domain.** Choose this if you want to authenticate against a Windows Server.

If you are not sure which method to select, stay with Local, which is the default for SLES 10.

After selecting an authentication method, select **Next**.

The next dialog differs, depending on which authentication method you selected. We will cover here:


- Add Local Users
- Configure the System as an LDAP Client

The dialogs for NIS and Windows Domain are used in a similar fashion to obtain the necessary information to enable the respective authentication method.

Add Local Users

If you select **Local** as the authentication method, the following appears:

Figure 1-23



New Local User

Enter the **User's Full Name**, **Username**, and **Password** to assign to this user account.

When entering a password, distinguish between uppercase and lowercase. Passwords should not contain any special characters, such as accented characters.

With the current password encryption (Blowfish), the password length should be between 5 and 72 characters.

Valid password characters are letters, digits, blanks, and `! " # $ % & ' () * + , - . / : ; [\] ^ _ { | } ~`.

To ensure that the password was entered correctly, repeat it exactly in a second field. Do not forget your password.

Create the **User Login** from components of the full name by clicking **Suggestion**. It may be modified, but use only letters (no accented characters), digits, and `_`. Do not use uppercase letters in this

User's Full Name

Username

Suggestion

Password

Confirm Password

☐ Receive System Mail
 ☐ Automatic Login

User Management

Steps

Back

About

Next

You can use the following in this dialog to add local users to the system (account information is stored in the files `/etc/passwd` and `/etc/shadow`):

- **User Data.** Enter the full user name, the login name, and the password.

To provide effective security, a password should be 8 or more characters long. The maximum length for a password ranges from 8 to 128 characters, depending on the algorithm used to hash the password. While the Crypt algorithm commonly used in the past used only the first eight characters of the password, more recent algorithms allow longer passwords.

Passwords are case-sensitive. Special characters are allowed, but they might be hard to enter depending on the keyboard layout.

- **Password Settings.** Select this option to change advanced password settings (such as password expiration). The default settings are suitable in most cases.
- **Details.** Select this option to edit details of the user account. The default settings are suitable in most cases.
- **Receive System Mail.** Select this option to forward all emails addressed to root to this user.
- **Automatic Login.** Select this option to enable automatic login for this user. This option logs in the user automatically (without requesting a password) when the system starts.

You should not enable this feature on a production system.
- **User Management.** Select this option to add more users (with the YaST User Management module).



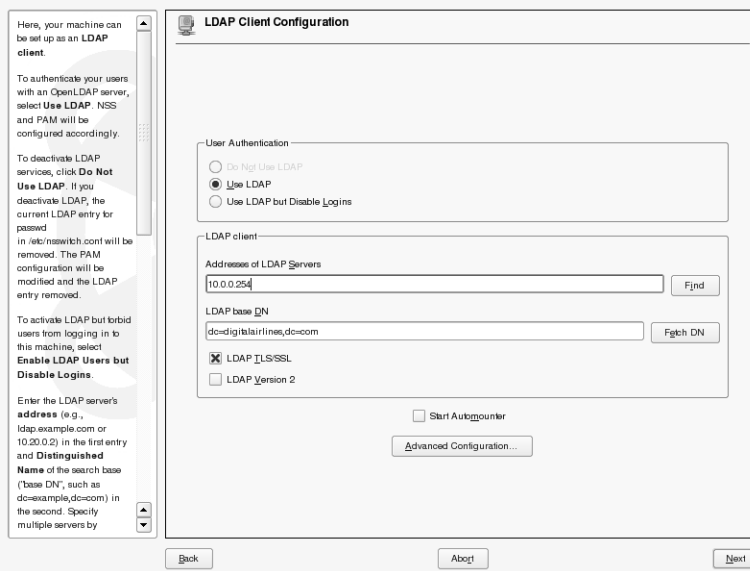
You can add other users later (after installation), but you have to create at least one user during installation so you don't have to work as the user root after the system has been set up.

After you enter all required information, select **Next**.

Configure the System as an LDAP Client

If you select LDAP as authentication method, the following appears:

Figure 1-24



From this dialog, you can configure your system as an LDAP client. The default configuration suggests to use a locally installed LDAP server.

You can change the configuration with the following options:

- **LDAP client.** You can configure the following:
 - **Addresses of LDAP Servers.** Enter the address of the LDAP server.
 - **LDAP base DN.** Enter the search base on the server.

- ❑ **LDAP TSL/SSL.** Select this option to encrypt the communication with the LDAP server.
- ❑ **LDAP Version2.** Select this option if your LDAP server only support LDAP version 2. By default, LDAP version 3 is used.
- **Start Automounter.** If your LDAP server provides information about the automatic mounting of file systems (such as home directories), you can start the automounter and use the automount information from the LDAP server.
- **Advanced Configuration.** Select this option to change advanced LDAP settings.

When finished with the LDAP configuration, select **Next**.

The Release notes are displayed. You should read them to make sure you are informed about the latest changes.

Configure Hardware

Selecting Next opens the Hardware Configuration dialog.

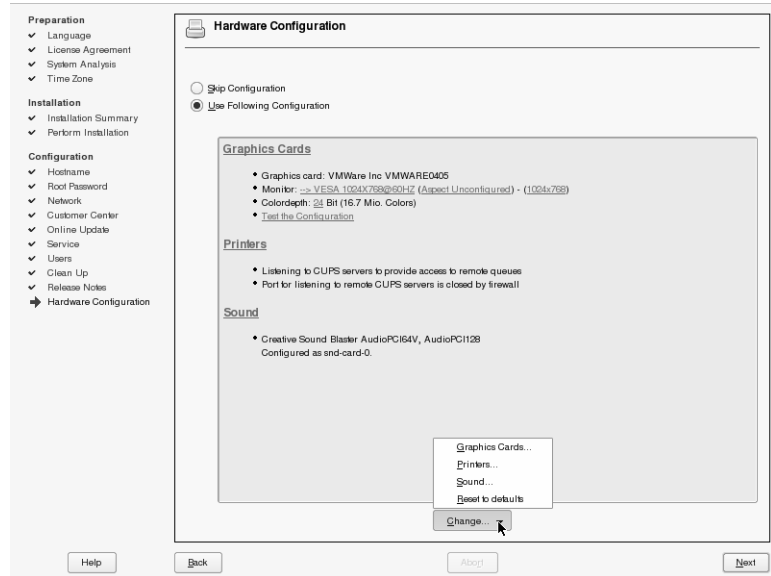
The configuration proposal contains the following items:

- **Graphics Cards.** Displays the graphic card and monitor setup.
- **Printers.** Displays the printer and printer server settings.
- **Sound.** Displays the configuration of the sound card.

To change the automatically generated configuration, select the headline of the item you want to change, or select the corresponding entry in the **Change** drop-down list.

You can also use the **Change** drop-down list to reset all settings to the automatically generated configuration proposal.

Figure 1-25



You can skip the hardware configuration at this time and configure your devices later in the installed system. However, if the settings of the graphics card in the configuration proposal are not correct, you should change them now to avoid problems during the first system start.

This is done by selecting the respective underlined entry and adjusting the values as needed.

Finalize the Installation Process

Confirm your hardware settings by selecting **Next**, and then select **Finish**. Unless you remove the check mark in front of **Clone This System for Autoyast**, an autoyast file is generated and saved as `/root/autoinst.xml`, which you can use to set up an identical system.

The system starts the graphical login screen, where you can log in with your previously created user account. SLES 10 is installed on your system.

Objective 3 Troubleshoot the Installation Process

SUSE Linux Enterprise Server 10 has been installed and tested on many different machines and hardware platforms. However, sometimes problems can occur.

The following table contains an overview of the most common installation problems, possible causes, and solutions:

Table 1-1

Problem	Possible Cause	Solution
The system does not start from the installation media.	The system is not configured to boot from the CD or DVD drive.	Enter the BIOS setup of the system and choose the CD or DVD drive as the first boot drive. Read the system manual for details about the BIOS setup.
	The CD or DVD drive is defective.	Try to boot a different system with <i>SLES 10 CD 1</i> . If it works, the CD or DVD drive of the actual system might be defective.
	The installation CD or DVD is defective.	If the installation CD does not boot on a different system, the CD or DVD itself could be defective. Contact your reseller to exchange the SLES 10 CD or DVD set.

Table 1-1 *(continued)*

Problem	Possible Cause	Solution
The installation program does not start.	Your system does not support newer hardware features correctly.	Select Installation – ACPI Disabled . If that doesn't fix the problem, select Installation – Save Settings from the Boot menu of the CD or DVD.
	Your system has less than 256 MB of main memory.	Install at least 256 MB of main memory and start the installation again.
The installation process stops.	Your system does not support newer hardware features correctly.	Select Installation – ACPI Disabled . If that doesn't fix the problem, select Installation – Save Settings from the Boot menu of the CD or DVD.
	The installation CD or DVD is defective.	If the installation process also stops on a different system, the CD or DVD could be defective. Contact your reseller to exchange the SLES 10 CD or DVD set.

Table 1-1 *(continued)*

Problem	Possible Cause	Solution
The network connection test or Online Update fails.	There is no DHCP server in the network.	If you configured your network card to use DHCP, assign a static IP address and configure routing and DNS settings manually.
	There is no route to the Internet.	Set the default gateway correctly.
	There is no direct Internet connection, but the system is using no or the wrong proxy settings.	Set the correct proxy configuration in the network configuration dialog. You can also skip the connection test and the Online Update and perform an Online Update in the installed system.
The graphical login does not appear after the installation is completed.	You are using the wrong X11 configuration.	Change to a text console and enter init 3 . Start sax2 from the command line and correct the X11 configuration. Enter init 5 to get a graphical login screen.

Exercise 1-1 Install SUSE Linux Enterprise Server 10

In this exercise, you install SUSE Linux Enterprise Server 10.

You will find this exercise in the workbook.

(End of Exercise)

Summary

Objective	Summary
1. Perform a SLES 10 Installation	<p>During the installation, the hard disks are prepared and the software packages are installed.</p> <p>The following tasks belong to the installation:</p> <ul style="list-style-type: none">■ Boot from the installation media.■ Select the language.■ Select the installation mode.■ Understand and change the installation proposal.■ Perform hard disk partitioning.■ Change the software selection.■ Launch the installation process.
2. Configure the SLES 10 Installation	<p>In the configuration step, you customize and configure the installed system.</p> <p>The following tasks belong to the configuration step:</p> <ul style="list-style-type: none">■ Set the root password.■ Configure the network.■ Configure Network Services.■ Manage Users.■ Configure Hardware.■ Finalize the Installation Process.

CNI USE ONLY-1 HARDCOPY PERMITTED

Objective	Summary
3. Troubleshoot the Installation Process	<p>SLES 10 has been installed and tested on many different machines and hardware platforms. However, sometimes installation problems can occur.</p> <p>Some issues to look for are:</p> <ul style="list-style-type: none">■ The system is not configured to boot from the CD or DVD drive.■ The CD or DVD drive is defective.■ The installation CD or DVD is defective.■ The system does not support newer hardware features (ACPI) correctly.■ There is no DHCP server in the network.■ There is no route to the Internet.■ You are using the wrong proxy settings.■ You are using the wrong X11 configuration.

CNI USE ONLY-1 HARDCOPY PERMITTED

SECTION 2 Administer the Linux File System

In this section, you learn how to manage your SUSE Linux Enterprise Server file system by implementing partitions, creating file systems, checking the file system for errors, setting up LVM and software RAID, and configuring disk quotas.

Objectives

1. Select a Linux File System
2. Configure Linux File System Partitions
3. Manage Linux File Systems
4. Configure Logical Volume Manager (LVM) and Software RAID
5. Set Up and Configure Disk Quotas

Objective 1 **Select a Linux File System**

One of the key roles performed by the Linux operating system is providing storage services through creating and managing a file system.

To successfully select a file system that meets your server requirements, you need to understand the following about file systems available for Linux:

- Linux File Systems
- Virtual Filesystem Switch
- Linux File System Internals
- File System Journaling
- Additional File System Documentation

It is very important to keep in mind that there might be no file system that best suits all kinds of applications. Each file system has its particular strengths and weaknesses, which must be taken into account.

Always bear in mind that even the most sophisticated file system cannot be a substitute for a reasonable backup strategy.



For additional details on specific file systems (such as ext3 and ReiserFS), see Section 18.2 in the ***SLES 10 Installation and Administration manual*** (/usr/share/doc/manual/sles-admin_en/, package sles-admin_en).

Also see “Additional File System Documentation” on page 2-14 at the end of this objective.

Linux File Systems

The type of file system you select depends on several factors (including speed and journaling). The following describes the file systems and formats available on Linux:

- Traditional File Systems
- Journaling File Systems

All of these file system types are included in the 2.6 Linux kernel (used in SUSE Linux Enterprise Server 10).

You can enter the following command to list the file system formats the kernel currently supports:

cat /proc/filesystems

Traditional File Systems

Traditional file systems supported by Linux do not journal data or metadata (permissions, file size, timestamps, etc.). These include the following:

- **ext2.** The ext2 file system is inode-based, designed for speed, is efficient, and does not fragment easily.

Because of these features, ext2 continues to be used by many administrators, even though it does not provide a journaling feature.

The ext2 file system has been available for many years, and is easily converted to an ext3 file system.

- **MS-DOS/VFAT.** FAT (File Allocation Table) is the primary file system for consumer versions of Microsoft Windows up to and including Windows Me.

VFAT is the 32-bit version of FAT that includes long filenames.

- **minix.** The minix file system is old and fairly limited, but is still sometimes used for floppy disks or RAM disks.

Journaling File Systems

A journaling file system is a file system that logs changes to a journal before actually writing them to the main file system. Depending on the file system and how it is mounted, the journal can include metadata, or also the data itself.

The following file systems available for Linux include a journaling feature:

- **ext3.** ext3 is the version of the ext2 file system that supports journaling.
- **ReiserFS.** Originally designed by Hans Reiser, ReiserFS treats the entire disk partition as if it were a single database table, storing not only the file metadata, but the file itself.

Directories, files, and file metadata are organized in an efficient data structure called a “balanced tree,” which offers significant speed improvements for many applications, especially those which use lots of small files.

- **XFS.** XFS is a high-performance journaling file system from SGI. It provides quick recovery after a crash, fast transactions, high scalability, and excellent bandwidth.

XFS combines advanced journaling technology with full 64-bit addressing and scalable structures and algorithms.



For details on XFS, see <http://oss.sgi.com/projects/xfs/>.

- **NTFS.** NTFS (New Technology File System), the file system used by Windows NT.

Currently only reading of the file system is supported under Linux. Support for creating, changing and deleting files is still experimental.

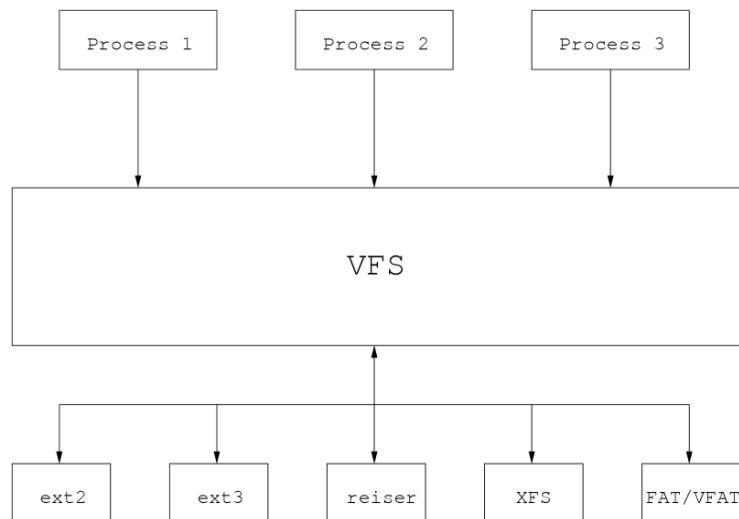
Virtual Filesystem Switch

For a user or program, it does not matter which file system format is used. The same interface to the data always appears. This is implemented by the Virtual Filesystem Switch (VFS) (also referred to as the *virtual file system*).

This is an abstract level in the kernel providing defined interfaces for processes. It includes functions such as open a file, write to a file, and read a file.

A program does not have to worry about how file access is implemented technically. The VFS forwards these requests to the corresponding driver for the file system format, as illustrated in the following:

Figure 2-1



One of the features of the VFS is to display file characteristics to the user as they are known from UNIX file system formats. This includes access permissions, even if they do not exist, as is the case with FAT/VFAT.

Linux File System Internals

File systems in Linux are characterized by the fact that data and administration information are kept separate. Each file is described by an *inode* (index node or information node).

Each of these inodes has a size of 128 bytes and contains all the information about this file except the filename. This includes details such as the owner, access permissions, the size, various time details (time of modification, last time of access, and time of modification of the inode), and the links to the data blocks of the file.

How data organization takes place differs from one file system format to the next. To understand the basics of file system data organization on Linux, you need to know the following:

- ext2fs File System Format
- ReiserFS Format
- Directories
- Network File System Formats

ext2fs File System Format

The ext2 file system format is, in many ways, identical to traditional UNIX file system formats. The concepts of inodes, blocks, and directories are the same.

When a file system is created (the equivalent of formatting in other operating systems), the maximum number of files that can be created is specified. The inode density (together with the capacity of the partition) determines how many inodes can be created.

Remember that it is not possible to generate additional inodes later. You can only specify the inode density when creating the file system.

An inode must exist for each file or directory on the partition. The number of inodes also determines the maximum possible number of files. Typically, an inode is generated for 4096 bytes of capacity.

On average, each file should be 4 KB in size for the capacity of the partition to be used optimally. If a large number of files are smaller than 4 KB, more inodes are used compared with the capacity.

This can result in the system being unable to create any more files, even if there is still space on the partition.

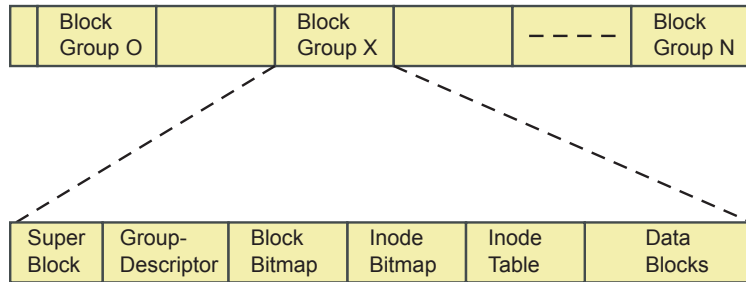
For applications that create a large number of very small files, the inode density should be increased by setting the corresponding capacity to a smaller value (such as 2048 or even 1024). However, the time needed for a file system check will increase substantially.

The space on a partition is divided into *blocks*. These have a fixed size of 1024, 2048, or 4096 bytes. You specify the block size when the file system is created; it cannot be changed later.

The block size determines how much space is reserved for a file. The larger this value is, the more space is consumed by the file, even if the actual amount of data is smaller.

In the classic file system formats (to which ext2 also belongs), data is stored in a linear chain of blocks of equal size. A specific number of blocks is grouped together in a block group (as illustrated in the following) and each block group consists of 32768 blocks:

Figure 2-2



The boot sector is located at the beginning of this chain and contains static information about the file system, including where the kernel to load can be found.

Each block group contains the following components:

- **Superblock.** The superblock is read when the file system is mounted and contains the following information about the file system:
 - The number of free and occupied blocks and inodes.
 - The number of blocks and inodes for each block.
 - Information about file system use, such as the time of the last mount, the last write access, and the number of mounts since the last file system check.
 - A valid bit, which is set to **0** when the file system is mounted and set to **1** again by umount.

When the computer is booted, the valid bit is checked. If it is set to 0 (power failure or reset), the automatic file system check is started.

The remains of files that can no longer be reconstructed are stored in the directory lost+found (in an ext2/ext3 file system).

For reasons of security, there are copies of the superblock. Because of this, the file system can be repaired, even if the first superblock has been destroyed.

- **Group Descriptor.** Information on the location of other areas (such as block bitmap and inode bitmap) is stored here. This information is stored at several locations within the file system for reasons of data security.
- **Block Bitmap.** Information is stored here indicating which blocks in this group are free or occupied.
- **Inode Bitmap.** Information is stored here indicating which inodes are free or occupied.
- **Inode Table.** File information is stored in this table that includes owners, access permissions, time stamps, and links to the data blocks in which the data is located.
- **Data Blocks.** This is where the actual data is located.

The ext2 file system format can process filenames with a length of up to 255 characters. With the path, a name can be a maximum of 4096 characters in length (slashes included).

A file can be up to 16 GB in size for a block size of 1024 bytes or two TB for a block size of 4096 bytes. The maximum file system size is two TB (with a block size of 1024 bytes) or 16 TB (with a block size of 4096 bytes).



The limitation on file size remains for the ext2 file system. However, the kernel can now handle files of almost any size.

ReiserFS Format

On a file system with ext2 and a block size of 1024 bytes, a file 8195 bytes in size occupies 8 blocks completely and a ninth block with three bytes.

Even though only three bytes are occupied, the block is no longer available. This means that approximately 11 percent of available space is wasted.

If the file is 1025 bytes in size, two blocks are required, one of which is almost completely empty. Almost 50 percent of the space is wasted.

A worst case occurs if the file is very small: even if the file is only 50 bytes in size, a whole block is used (95 percent wasted).

A solution to this problem is provided by the ReiserFS format, which organizes data in a different way. This file system format has currently a fixed block size of 4096 bytes.

However, small files are stored more efficiently. Only as much space is reserved as is actually required—not an entire block. Small files or the ends of files are stored together in the same block.

The inodes required are not generated when the file system is created, but only when they are actually needed. This allows a more flexible solution to storage requirements, increasing efficiency in the use of hard drive space.

Another advantage of the ReiserFS is that access to files is quicker. This is done through the use of balanced binary trees in the organization of data blocks.

However, balanced trees require considerably more processing power because after every file is written the entire tree must be rebalanced.

The current version of the ReiserFS (3.6) contained in the kernel since version 2.4.x allows a maximum partition size of 16 TB. A file also has a maximum size of 16 TB.

The same limitations exist for filenames as with the ext2 file system format.

Directories

Inodes contain all the administrative information for a file, but not the filename. The filename is stored in the directory.

Like a catalog, directories contain information on other files. This information includes the number of the inode for the file and its name.

Directories serve as a table in which inode numbers are assigned line-by-line to filenames. You can view the inode assigned to a filename by using the command **ls -li**, as in the following:

```
da10:~ # ls -li /
 2 .          104002 cdrom    80045 floppy    104081 mnt    103782 sbin
 2 ..         99068 dev      95657 home      81652 opt     80044 tmp
104005 bin    104004 dvd     102562 lib       1 proc       4 usr
 2 boot      95722 etc     95718 media     81598 root    80046 var
```

Each filename is preceded by the inode number.

On this particular SUSE Linux server there are 2 partitions: one holds the root directory **/**, and one holds the directory **/boot/**.

Because inodes are always uniquely defined on one partition only, the same inode numbers can exist on each partition.

In the example, the two entries “.” (a link to the current directory—here the root directory) and **boot** (the second partition is mounted on this directory) have the same inode number (2), but they are located on different partitions.

CNI USE ONLY-1 HARDCOPY PERMITTED

If you were to unmount the /boot partition, `ls -li` would show a different inode number, that of the directory /boot (the mountpoint) on the root partition. The same holds true for /proc.

The file “.”, which is actually a link to the previous layer in the direction of the root directory, also has an inode number of 2. Because you are already in the root directory, this link points to itself. It is another name entry for an inode number.

The table (the directory file) for the root directory can be represented as in the following example:

Table 2-1

Inode Number	Filename
2	.
2	..
4	usr
5	proc
18426	boot
80044	tmp
80045	floppy
80046	var
...	...

Network File System Formats

In addition to the already mentioned file system formats on the local computer, Linux also understands various network file system formats. The most significant of these is the Network File System (NFS), the standard in the UNIX world.

With NFS, it does not matter which file system format is used locally on individual partitions. As soon as a computer is functioning as an NFS server, it provides its file systems in a defined format NFS clients can access.

CNI USE ONLY-1 HARDCOPY PERMITTED

Using additional services included on SUSE Linux Enterprise Server, Linux can also work with the network file system formats of other operating systems.

These include the Server Message Block (SMB) format used in Windows and the Netware Core Protocol (NCP) from Novell.

SMB allows Linux to mount Windows 9x/NT/XP network shares.



File types, like directories, FIFOs, Sockets as well as the layout of the file system tree are covered in *SUSE Linux Enterprise Server 10 Fundamentals* (Course 3071).

File System Journaling

File systems are basically databases that store files and use file information such as the filename and timestamp (called *metadata*) to organize and locate the files on a disk.

When you modify a file, the file system performs the following transactions:

- It updates the file (the data)
- It updates the file metadata

Because there are two separate transactions, corruption can happen when only the file data is updated (but not the metadata) or vice versa, resulting in a difference between the data and metadata.

This can be caused, for instance by a power outage. The data might have been written already, but the metadata might not have been updated yet.

When there is a difference between the data and metadata, the state of the file system is inconsistent and requires a file system check and possibly repair. For ext2, this includes a walk through the entire file system, which is very time consuming on today's hard disks with hundreds of GB capacity.

In a journal-based file system, the journal keeps a record of all current transactions, and updates the journal as transactions are completed. Checking the file system, for instance after a power outage, consists mainly in replaying the journal and is much faster than checking the entire file system.

For example, when you first start copying a file from a network server to your workstation, the journaled file system submits an entry to the journal indicating that a new file on the workstation is being created.

After the file data and metadata are copied to the workstation, an entry is made indicating that the file was created successfully.

While recording entries in a journal requires extra time for creating files, it makes recovering an incomplete transaction easy as the journal can be used to repair the file system.

Additional File System Documentation

Each of the Linux file systems maintains its own home page on which to find mailing list information, further documentation, and FAQs. These include the following:

Table 2-2

File System	URL
ext2	http://e2fsprogs.sourceforge.net/
ReiserFS and Reiser4	http://www.namesys.com/
SGI's XFS	http://oss.sgi.com/projects/xfs/

A comprehensive multipart tutorial about Linux file systems can be found at IBM developerWorks at the following URL:

<http://www-106.ibm.com/developerworks/library/l-fs.html>

If you are interested in the limit of various file systems (file and file system sizes), visit http://www.novell.com/products/linuxenterpriseserver/kernel_limits.html.

The Linux Filesystem Hierarchy Standard (FHS) can be found at: <http://www.pathname.com/fhs/>

Objective 2 **Configure Linux File System Partitions**

A basic task of all system administrators is maintaining file system layouts. As a note of caution, you should always back up your data before working with tools that change the partition table or the file systems.

In most cases, YaST proposes a reasonable partitioning scheme during installation that can be accepted without change. However, you can also use YaST to customize partitioning after installation.

On the command line, you would first use `fdisk` to manage partitions, and then create a file system on that partition using `mkfs`.

To implement partitions on your SUSE Linux Enterprise Server, you need to know the following:

- Linux Device and Partition Names
- Design Guidelines for Implementing Partitions
- Manage Partitions with YaST
- Manage Partitions with `fdisk`

Linux Device and Partition Names

The different partition types available on x86 hardware have already been covered in “The Basics of Hard Drive Partitioning” on page 1-11

The following table shows the names of the Linux devices used for hard drives:

Table 2-3

Device	Linux Name
Primary master IDE hard disk	/dev/hda
Primary slave IDE hard disk	/dev/hdb

Table 2-3 *(continued)*

Device	Linux Name
Secondary master IDE hard disk	/dev/hdc
Secondary slave IDE hard disk	/dev/hdd
First SCSI hard disk	/dev/sda
Second SCSI hard disk	/dev/sdb

Partitions follow the naming convention of the device name and partition number.

For example, the first partition on the first IDE drive would be /dev/hda1 (/dev/hda + 1 as the first partition). The first logical partition defined on an IDE hard disk will always be number 5.

The following table shows the partition names corresponding to the device the partition is defined on:

Table 2-4

Partition	Linux Name
First partition on first IDE hard drive	/dev/hda1
Second partition on first IDE hard drive	/dev/hda2
First partition on third SCSI hard drive	/dev/sdc1
First logical partition on first IDE hard drive	/dev/hda5
Second logical partition on first IDE hard drive	/dev/hda6

For example, if you perform a new installation of SuSE Linux on a system with 2 IDE drives you might want the first drive to include a partition for swap and /. You might want to put all logs, mail, and home directories on the second hard drive.

The following is an example of how you might want to partition the disks (it assumes that the CD-ROM drive is the slave on the first IDE controller):

Table 2-5

Partition	Linux Name
Swap partition	/dev/hda1
/ partition	/dev/hda2
Extended partition on second disk	/dev/hdc1
/var as a logical partition on second disk	/dev/hdc5
/home as a logical partition on second disk	/dev/hdc6
/app1 as a logical partition on second disk	/dev/hdc7



On older installations you often find a small partition for /boot/. The reason for this is that the boot loader LILO needed the kernel within the first 1024 cylinders of the hard disk to boot the machine.

Design Guidelines for Implementing Partitions

YaST normally proposes a reasonable partitioning scheme with sufficient disk space. This is usually a swap partition (between 256 and 500 MB) and with the rest of the disk space reserved for a / partition.

In addition, if there is an existing partition on the hard drive, YaST attempts to maintain that partition.

If you want to implement your own partitioning scheme, consider the following recommendations:

Disk Space Distribution

Depending on the amount of space and how the computer will be used, adjust the distribution of the available disk space. The following are some basic guidelines:

Up to 4 GB

One partition for the swap space and one root partition (/). In this case, the root partition must allow for those directories that often reside on their own partitions if more space is available.

4 GB or More

A swap partition, a root partition (1 GB), and 1 partition each for the following directories as needed:

- **/boot/**. Depending on the hardware, it might also be useful to create a boot partition (/boot) to hold the boot mechanism and the Linux kernel.

This partition should be located at the start of the disk and should be at least 20 MB or 1 cylinder.

As a rule of thumb, always create such a partition if it was included in YaST's original proposal. If you are unsure about this, create a boot partition to be on the safe side.

- **/opt/**. Some (mostly commercial) programs install their data in /opt/. In this case, you might want to create a separate partition for /opt/ (4 GB or more). For instance KDE and GNOME are installed in /opt/.
- **/usr/**. Apart from directories holding user data, /usr/ is usually the biggest directory in the Linux installation. Putting it on a separate partition allows special mount options, like read only to prevent changes to programs. Software updates require to remount the partition read-write, though.

- **/var/**. If the computer is used as a mail server, it might be a good idea to put **/var/** on a separate partition. While too much mail might still render the mail service unusable, they would just fill the partition containing the **/var** directory, not the root file system. The administrator would still be able to administer the server and correct the issue.
- **/srv/**. When the machine acts as a web or ftp server, the data offered to users could be put on a separate partition.
- **/home/**. Putting **/home/** on a separate partition prevents users from using up all disk space and facilitates updates. If you have to reinstall the operating system you can preserve data in **/home** by leaving the partition untouched.
- **/tmp/**. Having **/tmp/** on a separate partition allows you to mount it with special options, like **noexec**, and also prevents processes from filling the disk with files in **/tmp/**.
- **Additional partitions**. If the partitioning is performed by YaST and other partitions are detected in the system, these partitions are also entered in the file **/etc/fstab** to enable easy access to this data.

The following is an example:

```
dev/sda8 /data2 auto noauto,user 0 0
```

Such partitions, whether they are Linux or FAT, are specified by YaST with the options **noauto** and **user**. This allows any user to mount or unmount these partitions as needed.

For security reasons, YaST does not automatically enter the **exec** option, which is needed for executing programs from the respective location. However, you can enter this option manually.

Entering the **exec** option is necessary if you encounter system messages such as “bad interpreter” or “Permission denied”.

Manage Partitions with YaST

You can use the YaST Expert Partitioner during or after installation to customize the default or existing partition configuration.

The interface of the Expert Partitioner after installation does not differ from the interface you used during installation (see “Verify Partitioning” on page 1-10).

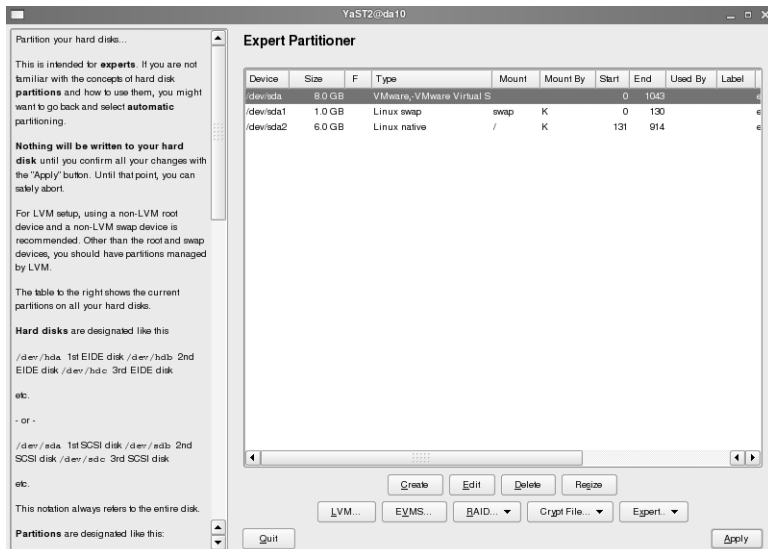
To start the Expert Partitioner, press **Alt+F2**, enter **yast2**, and enter the root password when prompted. Then select **System > Partitioner**. The following warning appears:

Figure 2-3



After selecting **Yes**, the expert partitioner appears:

Figure 2-4



The expert partitioner lets you modify the partitioning of your hard disk. You can manage the list of partitions by adding (**Create**), editing (**Edit**), deleting (**Delete**), or resizing (**Resize**) partitions.

Entire hard disks are listed as devices without numbers (such as /dev/hda or /dev/sda). Partitions are listed as parts of these devices (such as /dev/hda1 or /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 is mounted in the Linux file system tree.

Please refer to “Use the YaST Expert Partitioner ” on page 1-16 for details on how to create, edit, delete, and resize partitions.

Manage Partitions with fdisk

The program **fdisk** is used for partitioning hard disks from the command line.

To view the current partitioning scheme, use the option **-l**:

```
da10:~ # fdisk -l

Disk /dev/sda: 9139 MB, 9139200000 bytes
255 heads, 63 sectors/track, 1111 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device      Boot    Start      End    Blocks   Id  System
/dev/sda1                1        97     779121   82  Linux swap / Solaris
/dev/sda2    *        98       620     4200997+  83  Linux
/dev/sda3                621     1111     3943957+   f  W95 Ext'd (LBA)
/dev/sda5                621     751     1052226   83  Linux
```

To change the partition scheme, you enter the device of the hard disk as a parameter:

```
da10:~ # fdisk /dev/sda

The number of cylinders for this disk is set to 1111.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help):
```

fdisk is used with the keyboard only: a letter, followed by **Enter**, carries out an action. The following table lists the most frequently used keys:

Table 2-6

Letter	Action
d	Deletes a partition.

Table 2-6

Letter	Action
m	Gives a short summary of the fdisk commands.
n	Creates a new partition.
p	Shows a list of partitions that are currently available on the hard disk specified.
q	Ends the program fdisk without saving changes.
t	Changes a partition's system id.
w	Saves the changes made to the hard disk and ends fdisk.

The following shows the partitioning using fdisk. The example starts with a hard disk with no partitions configured so far.

Enter **fdisk *hard_disk***, for example, **fdisk /dev/hdb**. You can always enter **m** (help) to view the available commands. Enter **p** (print) to view the current partition table:

```
Command (m for help): p

Disk /dev/hdb: 32 heads, 63 sectors, 528 cylinders
Units = cylinders of 2016 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
Command (m for help):
```

To create a primary partition, enter **n** (new); then enter **p** (primary) as shown in the following:

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-528): 1
Last cylinder or +size or +sizeM or +sizeK (1-528, default 528): +128M

Command (m for help):
```

To display the partition table with the current settings, enter **p** (print). The following is displayed:

```
Command (m for help): p

Disk /dev/hdb: 32 heads, 63 sectors, 528 cylinders
Units = cylinders of 2016 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hdb1            1          131     132016+   83   Linux

Command (m for help):
```

This partition table contains all the relevant information on the partition created:

- This is the first partition of this hard disk (Device hdb1).
- It begins at cylinder 1 (Start) and ends at cylinder 131 (End).
- It consists of 132016 blocks (Blocks).
- Its Hex code (Id) is 83.
- Its type is Linux (System).

To set up an extended partition, enter **n** (new); then enter **e** (extended) as shown in the following:

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
e
Partition number (1-4): 2
First cylinder (132-528): 132
Last cylinder or +size or +sizeM or +sizeK (132-528, default 528): 528

Command (m for help):
```

To display the partition table with the current settings, again enter **p**. The following is displayed:

```
Command (m for help): p

Disk /dev/hdb: 32 heads, 63 sectors, 528 cylinders
Units = cylinders of 2016 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hdb1             1          131     132016+   83  Linux
/dev/hdb2           132          528     400176    5  Extended

Command (m for help):
```

After an extended partition has been created, you can now set up logical partitions by entering **n** (new) and then entering **l** (logical) as shown in the following:

```
Command (m for help): n
Command action
  l   logical (5 or over)
  p   primary partition (1-4)
l
First cylinder (132-528, default 132): 132
Last cylinder or +size or +sizeM or +sizeK (132-528, default 528): +128M

Command (m for help):
```

CNI USE ONLY-1 HARDCOPY PERMITTED

The current settings now look like this:

```
Command (m for help): p

Disk /dev/hda: 32 heads, 63 sectors, 528 cylinders
Units = cylinders of 2016 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hdb1           1          131     132016+    83   Linux
/dev/hdb2          132          528     400176     5   Extended
/dev/hdb5          132          262     132016+    83   Linux

Command (m for help):
```

The standard type for these partitions is Linux. To view the available types, enter l:

```
0 Empty                1b Hidden Win95 FA 63 GNU HURD or Sys b7 BSDI fs
1 FAT12                1c Hidden Win95 FA 64 Novell Netware b8 BSDI swap
2 XENIX root           1e Hidden Win95 FA 65 Novell Netware c1 DRDOS/sec (FAT-
3 XENIX usr            24 NEC DOS          70 DiskSecure Mult c4 DRDOS/sec (FAT-
4 FAT16 <32M          39 Plan 9           75 PC/IX           c6 DRDOS/sec (FAT-
5 Extended             3c PartitionMagic  80 Old Minix       c7 Syrix
6 FAT16                40 Venix 80286      81 Minix / old Lin da Non-FS data
7 HPFS/NTFS            41 PPC PReP Boot   82 Linux swap      db CP/M / CTOS / .
8 AIX                  42 SFS             83 Linux           de Dell Utility
9 AIX bootable         4d QNX4.x           84 OS/2 hidden C:  e1 DOS access
a OS/2 Boot Manag     4e QNX4.x 2nd part  85 Linux extended  e3 DOS R/O
b Win95 FAT32         4f QNX4.x 3rd part  86 NTFS volume set  e4 SpeedStor
c Win95 FAT32 (LB 50 OnTrack DM      87 NTFS volume set  eb BeOS fs
e Win95 FAT16 (LB 51 OnTrack DM6 Aux  8e Linux LVM       ee EFI GPT
f Win95 Extíð (LB 52 CP/M           93 Amoeba          ef EFI (FAT-12/16/
...
```

To change the partition type, for instance to create a swap partition, change the type by doing the following:

1. Enter **t**.
2. Enter the partition number.
3. Enter the hex code.

The following shows this procedure:

```
Command (m for help): t
Partition number (1-5): 5
Hex code (type L to list codes): 82
Changed system type of partition 5 to 82 (Linux swap)

Command (m for help):
```

The partition table now looks like this:

```
Command (m for help): p

Disk /dev/hdb: 32 heads, 63 sectors, 528 cylinders
Units = cylinders of 2016 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hdb1            1          131     132016+    83  Linux
/dev/hdb2           132          528     400176     5  Extended
/dev/hdb5           132          262     132016+    82  Linux swap

Command (m for help):
```

So far, nothing has been written to disk. If you want to discard your changes, enter **q** (quit). To actually write your changes to the partition table on the disk, enter **w** (write).

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or
resource busy.
The kernel still uses the old table.
The new table will be used at the next reboot.
Syncing disks.
```



When the new table is written, you are not asked for confirmation if you really want to do this.

As the output of `fdisk` says, you cannot directly use the new partition to create a file system on a new partition. You could now reboot as suggested, but you can also use the program

partprobe

to get the kernel to use the new partition table.

Objective 3 Manage Linux File Systems

To perform basic Linux file system management tasks in SUSE Linux Enterprise Server, you need to know how to do the following:

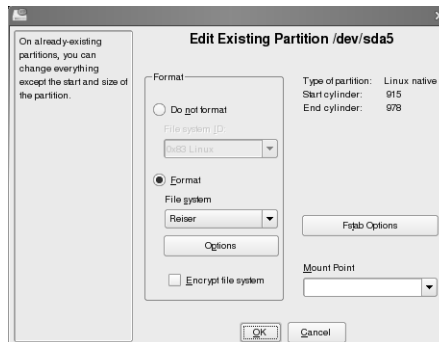
- Create a File System Using YaST
- Create a File System Using Command Line Tools
- Mount File Systems
- Monitor and Check a File System

Create a File System Using YaST

You can use YaST to create a file system (such as ext3 or ReiserFS) on a partition. This is done by starting the Expert Partitioner as root by entering in a console window **yast2 disk**. After acknowledging the warning message, the **Expert Partitioner** opens up.

To create a file system on a partition, select the partition and then select **Edit**; the following appears:

Figure 2-5



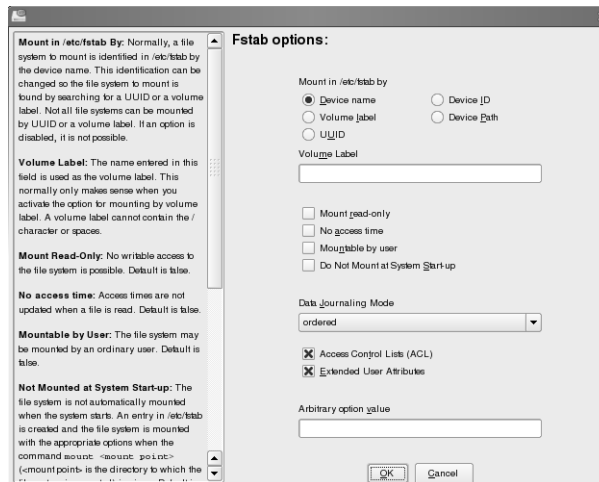
To format the partition with a file system, select **Format**. From the **File system** drop-down list, select a file system from the list of available file systems (such as **Reiser** or **Ext3**).

To view the available format options, select **Options**; the options shown depend on the file system you chose from the drop-down menu. We recommend keeping the default settings for most implementations. To return to the main format menu, select **OK**.

If you want to encrypt all data saved to the partition, select **Encrypt file system**. Encrypting a file system only prevents unauthorized mounting; once mounted the files are accessible like any other file on the system. You should only use this option for non-system partitions such as user home directories.

Select **Fstab Options** to edit the fstab entry for this partition.

Figure 2-6



These options are saved in **/etc/fstab** and are used when mounting the file system. In most cases the defaults offered don't need to be changed.

A description of each option is included in the left frame of the **Fstab options** dialog.

When you finish configuring the fstab options; select **Ok**.

In the **Mount Point** field enter the *directory* where the partition should be mounted in the file system tree. If the directory does not exist yet, it is automatically created by YaST.

When you finish configuring the file system and mounting parameters, select **OK**, and **Apply** in the Expert Partitioner dialog.

A warning message appears cautioning you about committing the changes you have made. Choosing **Apply** commits the changes to disk and returns you to the Expert Partitioner, whereas **Finish** commits them and closes the Expert Partitioner.

Create a File System Using Command Line Tools

There are various commands to create file systems, including `mke2fs`, `mkfs.ext3`, and `mkreiserfs`. You can use these to create file systems, such as `ext2`, `ext3`, and `ReiserFS`.

The alternative is the command **mkfs**, which is a frontend for the actual commands that create file systems (such as `mkfs.ext2`, `mkfs.ext3`, or `mkfs.msdos`).

When using **mkfs**, you need to use the option **-t** to indicate the file system type you want to create. If you do not indicate a file system type, `mkfs` automatically creates an `ext2` file system.

You need to know how to:

- Create an `ext2` or `ext3` File System
- Create a Reiser File System

Create an ext2 or ext3 File System

When you create an ext2 or ext3 file system with `mkfs`, you can use the following options:

Table 2-7

Option	Description
<code>-b <i>blocksize</i></code>	You can use this option to indicate the size of the data blocks in the file system. Values of 1024, 2048, . . . , 16384 are allowed for the block size.
<code>-i <i>bytes_per_inode</i></code>	You can use this option to indicate how many inodes are created on the file system. For <i>bytes_per_inode</i> you can use the same values available for the block size.
<code>-j</code>	You can use this option to create an ext3 Journal on the file system.

If you do not include options `-b` and `-i`, the data block sizes and the number of inodes is set by `mkfs`, depending on the size of the partition.

The following is an example of creating an ext3 file system on a partition. Please note that there is no confirmation required—the partition is formatted directly after pressing enter:

```
da10:~ # mkfs -t ext3 /dev/sda6
mke2fs 1.38 (30-Jun-2005)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
62248 inodes, 248976 blocks
12448 blocks (5.00%) reserved for the super user
First data block=1
31 block groups
8192 blocks per group, 8192 fragments per group
2008 inodes per group
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729, 204801, 221185

Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 32 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

This mkfs example creates ext3 file system on an existing partition with the following values:

- **Block size=1024 (log=0)**

The block size is 1 KB.

- **62248 inodes, 248976 blocks**

The maximum number of files and directories is 62248. The total number of blocks is 248976.

- **12448 blocks (5.00%) reserved for the super user**

5% of the entire space is reserved for the system administrator. If the hard disk is 95% full, then a normal user cannot use any more space.



You can also use the command `mke2fs` (`mkfs.ext2` and `mkfs.ext3` are hardlinks to the same file) to create an ext2 or ext3 file system (see **man mke2fs**).

Create a Reiser File System

You can create a Reiser file system by using the command **mkreiserfs** or **mkfs -t reiserfs**:

```
da10:~ # mkfs -t reiserfs /dev/sda6
mkfs.reiserfs 3.6.19 (2003 www.namesys.com)

A pair of credits:
Yury Umanets (aka Umka) developed libreiser4, userspace plugins,
...

Guessing about desired format.. Kernel 2.6.16.14-6-smp is running.
Format 3.6 with standard journal
Count of blocks on the device: 62240
Number of blocks consumed by mkreiserfs formatting process: 8213
Blocksize: 4096
Hash function used to sort names: "r5"
Journal Size 8193 blocks (first block 18)
Journal Max transaction length 1024
inode generation number: 0
UUID: 73abdf80-2b72-4844-9967-74e99813d056
ATTENTION: YOU SHOULD REBOOT AFTER FDISK!
      ALL DATA WILL BE LOST ON '/dev/sda6'!
Continue (y/n):y
Initializing journal - 0%....20%....40%....60%....80%....100%
Syncing..ok
ReiserFS is successfully created on /dev/sda6.
```

To find out about the available options, look at **man mkreiserfs**. Usually there is no need to use different values than those used by default.

Mount File Systems

In Windows systems separate drive letters represent different partitions. Linux does not use letters to designate partitions, it mounts partitions to a directory in the file system. Directories used for mounting are also called *mount points*.

For example, to add a new hard disk to a Linux system, first you partition and format the drive. You then use a directory (such as /data/) in the file system and mount the drive to that directory using the command **mount**.

To unmount (detach) a file system, you use the **umount** command (for details, enter **man umount**).



You can also mount remote file systems, shared via the Network File System (NFS), to directories you create in your file system.

The directory /mnt/ is used by default for temporarily mounting local and remote file systems. All removable devices are mounted by default to /media/, such as the following:

- A CD-ROM on /dev/cdrom is mounted by default to /media/cdrom.
- A floppy disk on /dev/floppy is mounted by default to /media/floppy.

When using SLES 10 from a desktop environment such as Gnome or KDE, media such as floppy disks and CDs are automatically mounted and unmounted. If the CD-ROM has a label, it is mounted to /media/*label*.

To manage mounting (and unmounting) file systems, you need to know the following:

- Configuration File for Mounting File Systems: /etc/fstab
- View Currently Mounted File Systems

- Mount a File System
- Unmount a File System

Configuration File for Mounting File Systems: `/etc/fstab`

The file systems and their mount points in the directory tree are configured in the file `/etc/fstab`. This file contains 1 line with 6 fields for each mounted file system.

The lines look similar to the following:

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
<code>/dev/hda2</code>	<code>/</code>	<code>reiserfs</code>	<code>acl,user_xattr</code>	<code>1</code>	<code>1</code>
<code>/dev/hda1</code>	<code>swap</code>	<code>swap</code>	<code>defaults</code>	<code>0</code>	<code>0</code>
<code>proc</code>	<code>/proc</code>	<code>proc</code>	<code>defaults</code>	<code>0</code>	<code>0</code>
<code>sysfs</code>	<code>/sys</code>	<code>sysfs</code>	<code>noauto</code>	<code>0</code>	<code>0</code>
<code>debugfs</code>	<code>/sys/kernel/debug</code>	<code>debugfs</code>	<code>noauto</code>	<code>0</code>	<code>0</code>
<code>usbfs</code>	<code>/proc/bus/usb</code>	<code>usbfs</code>	<code>noauto</code>	<code>0</code>	<code>0</code>
<code>devpts</code>	<code>/dev/pts</code>	<code>devpts</code>	<code>mode=0620,gid=5</code>	<code>0</code>	<code>0</code>
<code>/dev/fd0</code>	<code>/media/floppy</code>	<code>auto</code>	<code>noauto,user,sync</code>	<code>0</code>	<code>0</code>

Each field provides the following information for mounting the file system:

- **Field 1.** Lists the name of the device file, or the file system label, or the UUID (Universally Unique Identifier). Use of **LABEL=label** or **UUID=uuid** has the advantage that the partition is mounted correctly even if the device file used changes, for instance because you swapped hard disks on the IDE controller.
- **Field 2.** Lists the mount point—the directory to which the file system should be mounted. The directory specified here must already exist. You can access the content on the media by changing to the respective directory.
- **Field 3.** Lists the file system type (such as `ext2`, `reiserfs`).

- **Field 4.** Shows the mount options. Multiple mount options are separated by commas (such as **noauto,user,sync**).
- **Field 5.** Indicates whether to use the backup utility **dump** for the file system. **0** means no backup.
- **Field 6.** Indicates the sequence of the file system checks (with the **fsck** utility) when the system is booted:
 - **0:** file systems that are not to be checked
 - **1:** the root directory
 - **2:** all other modifiable file systems; file systems on different drives are checked in parallel

While `/etc/fstab` lists the file systems and where they should be mounted in the directory tree during startup, it does not contain information on the actual current mounts.

The `/etc/mtab` file lists the file systems currently mounted and their mountpoints. The `mount` and `umount` commands affect the state of mounted file systems and modify the `/etc/mtab` file.

The kernel also keeps information for `/proc/mounts`, which lists all currently mounted partitions.

For troubleshooting purposes, if there is a conflict between `/proc/mounts` and `/etc/mtab` information, the `/proc/mounts` data is always more current and reliable than `/etc/mtab`.

View Currently Mounted File Systems

You can view the file systems currently mounted by entering the command **mount**. Information similar to the following appears:

```
da10:~ # mount
/dev/sda2 on / type reiserfs (rw,acl,user_xattr)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
debugfs on /sys/kernel/debug type debugfs (rw)
udev on /dev type tmpfs (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
securityfs on /sys/kernel/security type securityfs (rw)
```

You can also view this information in the file `/proc/mounts`.

Mount a File System

You can use the command **mount** to manually mount a file system. The general syntax for mounting a file system with mount is

**mount [-t *file_system_type*] [-o *mount_options*] *device*
*mount_point_directory***

By using mount, you can override the default settings in `/etc/fstab`.

For example, entering the following mounts the partition `/dev/hda9` to the directory `/space/`:

```
mount /dev/hda9 /space
```

You do not usually specify the file system type because it is recognized automatically (using magic numbers in the superblock, or simply by trying different file system types; see **man mount** for details).

The following are some of the options you can use when mounting a file system with the command `mount` or by entering them in `/etc/fstab`:

- **remount.** This option causes file systems that are already mounted to be mounted again.

When you make a change to the options in `/etc/fstab`, you can use `remount` to incorporate the changes.
- **rw, ro.** These options indicate whether a file system should be writable (**rw**) or only readable (**ro**).
- **sync, async.** These options set synchronous (**sync**) or asynchronous (**async**) input and output in a file system. The default setting is `async`.
- **atime, noatime.** These options set whether the access time of a file is updated in the inode (**atime**) or not (**noatime**). The option `noatime` should improve the performance.
- **nodev, dev.** The **nodev** option prevents device files from being interpreted as such in the file system.
- **noexec, exec.** You can prohibit the execution of programs on a file system with the option **noexec**.
- **nosuid, suid.** The **nosuid** option ensures that the `suid` and `sgid` bits in the file system are ignored.

Some options only make sense in the file `/etc/fstab`. These options include the following:

- **auto, noauto.** File systems set with the option **noauto** in the file `/etc/fstab` are not mounted automatically when the system is booted.
- **user, nouser.** This option lets users mount the file system. Normally, this is a privilege of the user `root`.
- **defaults.** This option causes the default options `rw`, `suid`, `dev`, `exec`, `auto`, `nouser`, and `async` to be used.

The options **noauto** and **user** are usually combined for removable media such as floppy disk or CD-ROM drives.

Unmount a File System

Once a file system is mounted, you can use the **umount** command (without an “n”) to unmount the file system.

You can unmount the file system by using **umount** with the device or the mount point.

For example to unmount a CD file system mounted at `/media/cdrecorder`, you could enter one of the following:

- **umount /media/cdrecorder**

or

- **umount /dev/hdb**

In order to unmount the file system, no application or user may use the file system. If it is being used, Linux sees the file system as being “busy” and will refuse to unmount the file system.



To help determine the processes that are acting on a file or directory, you can use the **fuser** utility. For details on using the **fuser** utility, see “Identify Processes Using Files (**fuser**)” on page 2-45.

One way to make sure the file system is not busy is to enter **cd /** at the shell prompt before using the **umount** command. This command takes you to the root of the file system.

However, there might be times when the system (kernel) still sees the file system as busy, no matter what you try to do.

In these cases, you can enter **umount -f** to force the file system to unmount. However, we recommend using this only as a last resort, as there is probably a reason why the kernel thinks the file system is still mounted.

Exercise 2-1 *Configure Partitions on Your Hard Drive*

In this exercise, you practice creating partitions and file systems on them with YaST and **fdisk**. You also use command line tools to create file systems.

You will find this exercise in the workbook.

(End of Exercise)

Monitor and Check a File System

Once you set up and begin using your Linux file system, you can monitor the status and health of the system by doing the following from the command line:

- Check Partition and File Usage (`df` and `du`)
- Check Open Files (`lsof`)
- Identify Processes Using Files (`fuser`)
- Check lost+found (`ext2` and `ext3` only)
- Check and Repair File Systems (`fsck`)
- Check and Repair `ext2/ext3` and ReiserFS (`e2fsck` and `reiserfsck`)
- Use Additional tools to manage file systems

Check Partition and File Usage (`df` and `du`)

The following commands help you monitor usage by partitions, files, and directories:

- **df.** This command provides information on where hard drives and their partitions or other drives are mounted in the file system, and how much space they occupy.

If you use the `df` command without parameters, the space available on all currently-mounted file systems is displayed. If you provide a filename, `df` displays the space available on the file system this file resides in.

Some useful options include **-h** (human readable format—in MB or GB), **-i** (list inode information instead of block usage), and **-l** (limit listing to local file systems).

For example, to list information for all local file systems in human-readable format, you would enter **`df -lh`**.

- **du.** This command provides information on the space occupied by files and directories.

Some useful options include **-c** (display a grand total), **-h** (human-readable format), **-s** (display only a total for each argument), and **--exclude=*pattern*** (exclude files that match *pattern*).

For example, to display information for files in human-readable format except for files that end in “.o,” you would enter the following:

```
du -h --exclude='*.o'
```

Check Open Files (lsof)

The command **lsof** lists open files. Entering **lsof** without any options lists all open files belonging to all active processes.

An open file can be a regular file, a directory, a device file, a library, or a stream or a network file (Internet socket, NFS file, or UNIX domain socket.)

In addition to producing a single output list, **lsof** can run in repeat mode using the option **-r**. In repeat mode it outputs, delays, and then repeats the output operation until stopped with an interrupt or quit signal.

Some useful options include **-c *x*** (list only files starting with *x*), **-s** (display file sizes), and **-u *x*** (list only files for users who are *x*).

For example to list open files for the users root and geeko only and include the file sizes, you would enter **lsof -s -u root,geeko**.

Identify Processes Using Files (fuser)

The command **fuser** displays the PIDs of processes using the specified files or file systems.

In the default display mode, each filename is followed by a letter that describes the type of access:

- **c**: current directory
- **e**: executable being run
- **f**: open file (omitted in default display mode)
- **r**: root directory
- **m**: memory mapped file or shared library

A non-zero return code is displayed if none of the specified files is accessed or in case of a fatal error. If at least one access has been found, **fuser** returns zero.

Some useful options include **-a** (return information for all files, even if it they are not accessed by a process), **-v** (verbose mode), and **-u** (append the user name of the process owner to each PID).

Another useful option is **-m**. To check the PID information for processes accessing files on the partition that holds /home, you would enter **fuser -m /home**.

Check lost+found (ext2 and ext3 only)

The directory **lost+found** is a special feature of the ext2 and ext3 file system format. After a system crash, Linux automatically carries out a check of the complete file system. Files or file fragments to which a name can no longer be allocated are not simply deleted, but stored in this directory.

By reviewing the contents of this directory, you can try to reconstruct the original name and purpose of a file.

Check and Repair File Systems (fsck)

The command **fsck** lets you check and optionally repair one or more Linux file systems. Normally, fsck tries to run file systems on different physical disk drives in parallel to reduce the total amount of time to check all file systems.

If you do not specify a file system on the command line and do not specify the option **-A**, fsck defaults to checking filesystems in `/etc/fstab` serially.

fsck is a frontend for the various file system checkers (**fsck.fstype**) available on the system. The fsck utility looks for the system-specific checker in `/sbin/` first, then in `/etc/fs/` and `/etc/`, and finally in the directories listed in the `PATH` environment variable.

To check a specific file system, use the following syntax:

fsck device

For example if you wanted to check the file system on `/dev/hda2`, you would enter **fsck /dev/hda2**.

Some options that are available with fsck include **-A** (walk through the `/etc/fstab` file and try to check all the file systems in one pass), **-N** (don't execute, just show what would be done), and **-V** (verbose output).

Check and Repair ext2/ext3 and ReiserFS (e2fsck and reiserfsck)

Switching off the Linux system without unmounting partitions (for example, when a power outage occurs) can lead to errors in the file system.

The next time you boot the system, the fact that the computer was not shut down correctly is detected and a file system check is performed. If errors are found in the file system, they are corrected, if possible. If not, the computer does not start up properly and you are prompted to enter the root password, together with a hint on how to correct the issue. In cases of severe file system damage, you may even have to resort to the rescue system to repair the system.

Depending on the file system type, you use either `/sbin/e2fsck` or `/sbin/reiserfsck`. These tools check the file system for a correct superblock (the block at the beginning of the partition containing information on the structure of the file system), faulty data blocks, or faulty allocation of data blocks.

A possible problem in the ext2 (or ext3) file system is damage to the superblock. You can first view the location of all copies of the superblock in the file system using **dumpe2fs**.

Then, with **e2fsck**, you can use one of the backup copies, as in the following:

```
e2fsck -f -b 32768 /dev/hda1
```

In this example, the superblock located at data block 32768 in the ext2 file system of the partition `/dev/hda1` is used and the primary superblock is updated appropriately upon completion of the file system check.



With a block size of 4k, a backup copy of the superblock is stored every 32768 blocks.

With **reiserfsck**, the file system is subjected to a consistency check. The journal is checked to see if certain transactions need to be repeated. With the option **--fix-fixable**, errors such as wrong file sizes are fixed as soon as the file system is checked.

With an error in the binary tree, it is possible to have this rebuilt by entering **reiserfsck --rebuild-tree**.

Use Additional tools to manage file systems

There are additional tools to administer various aspects of file systems.

tune2fs is used to adjust tunable filesystem parameters on ext2/ext3 filesystems. Amongst these is the number of days or number of mounts a file system check is done. It is also used to add a label to the file system, or to add a journal to an ext2 file system, turning it into an ext3 file system.

reiserfstune is the corresponding tool for ReiserFS. See the reiserfstune manual page for options and uses for this tool.

resize2fs and **resize_reiserfs** are used to shrink or enlarge an ext2/3 and ReiserFS, respectively. **resize_reiserfs** can enlarge ReiserFS online. Shrinking file systems as well as enlarging ext2/3 can only be done while the file system is unmounted.



As stated before, when planning to manipulate partitions and file systems, back up your data first!

Exercise 2-2 Manage File Systems from the Command Line

In this exercise, you practice managing file systems from the command line.

You will find this exercise in the workbook.

(End of Exercise)

Objective 4 **Configure Logical Volume Manager (LVM) and Software RAID**

Logical volume manager (LVM) provides a higher-level view of the disk storage on a computer system than the traditional view of disks and partitions. This gives you much more flexibility in allocating storage space to applications and users.

After creating logical volumes with LVM, you can (within certain limits) resize and move logical volumes while they are still mounted and running.

You can also use LVM to manage logical volumes with names that make sense (such as “development” and “sales”) instead of physical disk names such as “sda” and “sdb.”

To configure a file system with LVM, you need to know the following:

- How to Use VM Components
- How to Use VM Features
- How to Configure Logical Volumes With YaST
- How to Configure LVM with Command Line Tools

The Linux Kernel is capable of combining hard disks to arrays with the RAID levels 0, 1, 5, and 6. Software RAID is covered in

- Manage Software RAID

How to Use VM Components

Conventional partitioning of hard disks on a Linux file system is basically inflexible. When a partition is full, you have to move the data to another medium before you can resize the partition, create a new file system, and copy the files back.

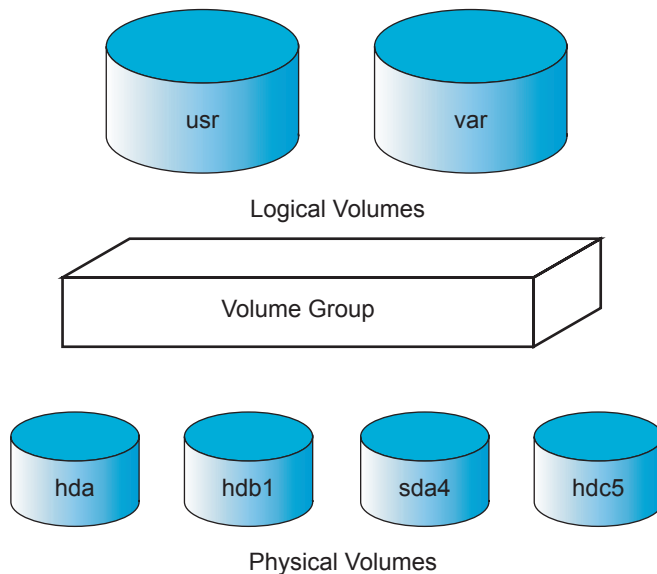
Normally, these changes cannot be implemented without changing adjacent partitions, whose contents also need to be backed up to other media and written to their original locations after the repartitioning.

Because it is difficult to modify partitions on a running system, LVM was developed. It provides a virtual pool of memory space (called a *volume group*) from which logical volumes can be generated if needed. The operating system accesses these logical volumes like conventional physical partitions.

This approach lets you resize the physical media during operation without affecting the applications.

The basic structure of LVM includes the following components:

Figure 2-7



- **Physical volume.** A physical volume can be a partition or an entire hard disk.

- **Volume group.** A volume group consists of one or several physical volumes grouped together. The physical partitions can be spread over different hard disks. You can add hard disks or partitions to the volume group during operation whenever necessary.

The volume group can also be reduced in size by removing physical volumes (hard disks or partitions).

- **Logical volume.** A logical volume is a part of a volume group. A logical volume can be formatted and mounted like a physical partition.

You can think of volume groups as hard disks and logical volumes as partitions on those hard disks. The volume group can be split into several logical volumes that can be addressed with their device names (such as `/dev/system/usr`) like conventional partitions with theirs (`dev/hda1`).



Just as with other direct manipulations of the file system, a data backup should be made before configuring LVM.

How to Use VM Features

LVM is useful for any computer, as it is very flexible when the need to adapt to changed needs for storage space arises.

The following are features of LVM that help you implement storage solutions:

- You can combine several hard disks or partitions into a large volume group.
- Provided the configuration is suitable, you can enlarge a logical volume when free space is exhausted. Resizing logical volumes is easier than resizing physical partitions.
- You can create extremely large logical volumes (Terabytes).

- You can add hard disks to the volume group in a running system, provided you have hot-swappable hardware capable of such actions.
- You can add logical volumes in a running system, provided there is free space in the volume group.
- You can use several hard disks with improved performance in the RAID 0 (striping) mode.
- There is no limit that is relevant in practice on the number of logical volumes (the limit in LVM version 1 was 256).
- The Snapshot feature enables consistent backups in the running system.

How to Configure Logical Volumes With YaST

The following are the basic steps for configuring logical volumes (LVM) with YaST:

- Define the LVM partitions (physical volumes) on the hard drive
- Create the volume group and logical volumes
- Access the YaST Module `lvm_config`

Define the LVM partitions (physical volumes) on the hard drive

During (or after) the installation of SUSE Linux Enterprise Server, you need to configure the LVM partition on the hard disk.

You can use YaST or `fdisk` to perform this task as described in “Configure Linux File System Partitions” on page 2-16.

For the File system ID, select **0x8E Linux LVM**. Do not create a file system on that partition.

Create the volume group and logical volumes

Select **LVM** in the YaST Expert Partitioner. The following appears:

Figure 2-8



You use this dialog to create a new logical volume group by entering the following:

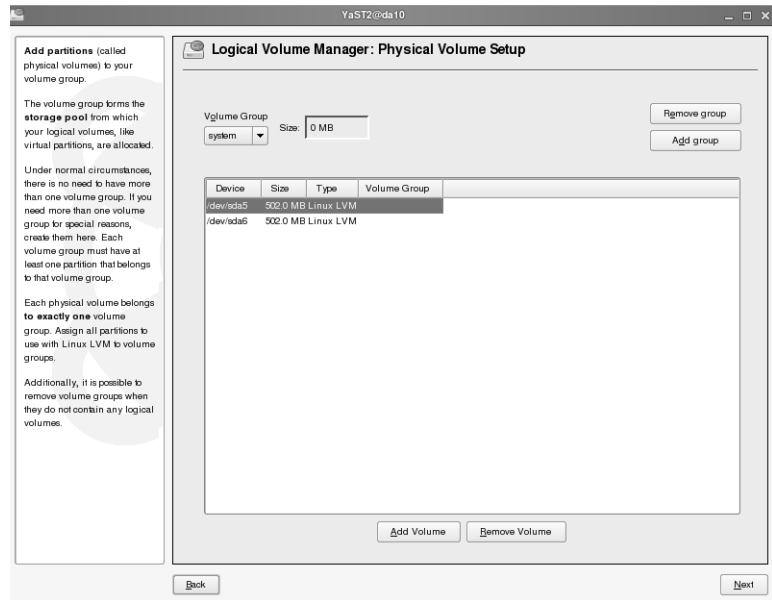
- **Volume Group Name.** Enter the name of your volume group.
- **Physical Extent Size.** The physical extent size defines the smallest unit of a logical volume group.

With LVM version 1, this also defined the maximum size of a logical volume. Entering a value 4 MB allowed logical volumes of 256 GB. With LVM2, this limitation does not exist anymore.

If you are not sure which values to enter, use the default settings.

After you select **OK**, the following appears:

Figure 2-9



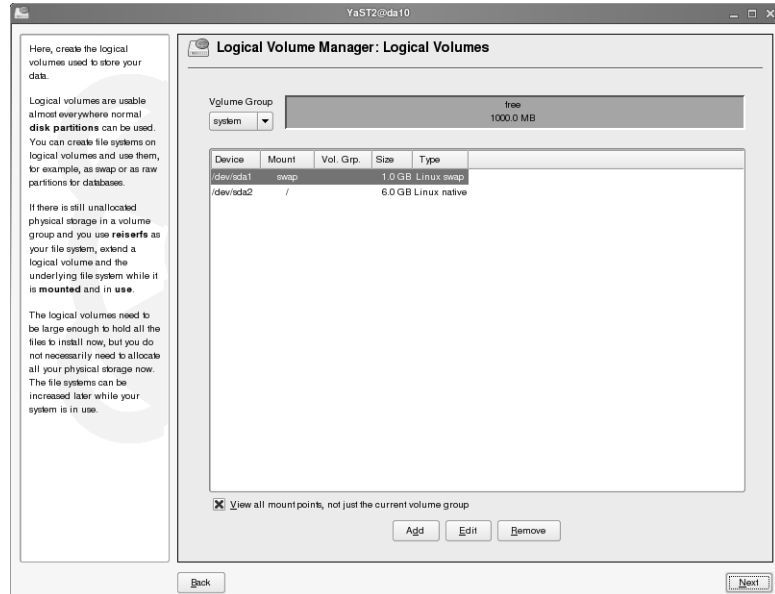
Here you can set up the physical volumes.

- **Volume Group.** Lets you select the volume group from the drop-down list that you want to add partitions to.
- **Size.** Displays the current size of the selected logical volume group.
- **Remove Group.** Deletes the currently selected volume group. You can delete empty groups only.
- **Add Group.** Adds a logical volume group.
- **Partition List.** Lets you select the partition you want to add to the volume group.
- **Add Volume.** Adds the selected partition to the volume group.

- **Remove Volume.** Removes the selected partition from the volume group.

Add physical volumes (these are usually partitions on a hard disk) to your volume group, and then select **Next** to continue. The following appears:

Figure 2-10

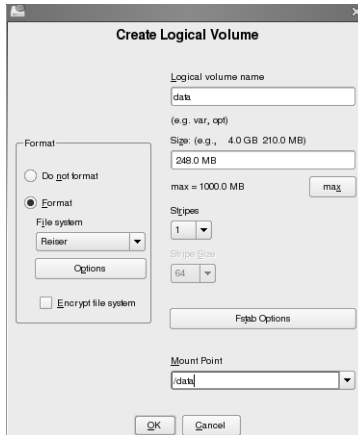


You can use the following to create logical volumes in your volume group:

- **Volume Group.** Allows you to select the volume group that you want to create partitions in.
- **Used/Available Space bar.** Displays the available space within the selected volume group.
- **Volume list.** Displays physical partitions and logical volumes in the system.

- **View all mount points, not just the current volume group.** When you select this option, all partitions and volumes that have entries in `/etc/fstab` are displayed. Otherwise, only the volumes in the selected volume group are displayed.
- **Add.** Adds a new logical volume to the volume group. When you select **Add**, the following appears:

Figure 2-11



This dialog lets you configure a logical volume using the same options available for creating a file system (see “Create a File System Using YaST” on page 2-30).

In addition, you can enter a logical volume name, the maximum amount of space available (by selecting **max**), the number of stripes (equal or less than the number of disks), and the stripe size (if you configure more than one stripe).

Striping is only useful if you have two or more disks. It can increase performance by allowing parallel file system read and writes, but it also increases the risk of data loss. One failed disk can lead to data corruption in the whole volume group.

- **Edit.** Allows you to change the parameters of a selected volume.

The dialog to edit a volume has the same options as the dialog to create volumes (already described). You can also edit logical volumes directly from the partition list in the Expert Partitioner.

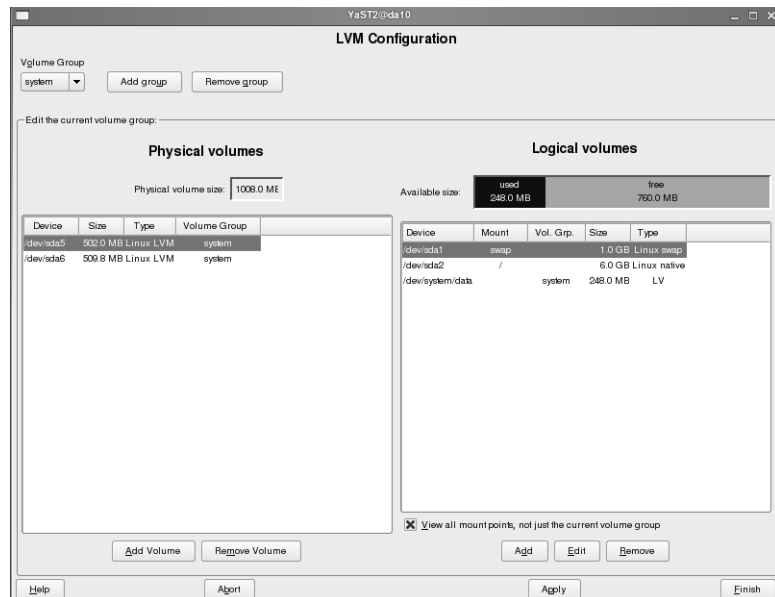
- **Remove.** Removes a selected volume. You can also remove logical volumes directly from the partition list in the Expert Partitioner.

When you are finished with the logical volume setup, select **Next** to save the settings and return to the Expert Partitioner.

Access the YaST Module `lvm_config`

To manage an existing LVM setup, you can access (as root) the YaST LVM configuration directly with **yast2 lvm_config**. It combines the configuration options for LVM in one dialog:

Figure 2-12



The configuration options are the same as those accessed by selecting LVM in the YaST Expert Partitioner.



For additional information on configuring LVM, see the LVM HOWTO at <http://tldp.org/HOWTO/LVM-HOWTO/>.

How to Configure LVM with Command Line Tools

Setting up LVM consists of several steps, with a dedicated tool for each:

- Tools to Administer Physical Volumes
- Tools to Administer Volume Groups
- Tools to Administer Logical Volumes

This is just a brief overview, not all available LVM tools are covered. To view the tools that come with LVM, enter **rpm -ql lvm2 | less** on a command line, and have a look at the corresponding manual pages for details on each of them.

Tools to Administer Physical Volumes

Partitions or entire disks can serve as physical volumes for LVM.

The ID of a partition used as part of LVM should be **Linux LVM, 0x8e**. However the ID **0x83, Linux**, works as well.

To use an entire disk as physical volume, it may not contain a partition table. Overwrite any existing partition table with **dd**:

```
da10:~ # dd if=/dev/zero of=/dev/hdd bs=512 count=1
```

The next step is to initialize the partition for LVM. The tool to use is **pvcreate**:

```
da10:~ # pvcreate /dev/hda9
Physical volume "/dev/hda9" successfully created
```

pvscan shows the physical volumes and their use:

```
da10:~ # pvscan
PV /dev/hda9    lvm2 [242,95 MB]
Total: 1 [242,95 MB] / in use: 0 [0    ] / in no VG: 1 [242,95 MB]
```

The tool **pvmove** is used to move data from one physical volume to another (providing there is enough space), in order to remove a physical volume from LVM.

Tools to Administer Volume Groups

The tool **vgcreate** is used to create a new volume group. To create the volume group system, and add the physical volume /dev/hda9 to it, enter:

```
da10:~ # vgcreate system /dev/hda9
Volume group "system" successfully created
da10:~ # pvscan
PV /dev/hda9    VG system    lvm2 [240,00 MB / 240,00 MB free]
Total: 1 [240,00 MB] / in use: 1 [240,00 MB] / in no VG: 0 [0    ]
```

pvscan shows the new situation.

To add further physical volumes to the group, use **vgexpand**. Removing unused physical volumes is done with **vgreduce** after shifting data from the physical volume scheduled for removal to other physical volumes using **pvmove**. **vgremove** removes a volume group, providing there are no logical volumes in the group.

Tools to Administer Logical Volumes

To create a logical volume, use **lvcreate**, specifying the size, the name for the logical volume, and the volume group:

```
da10:~ # lvcreate -L 100M -n data system
Logical volume "data" created
```

The next step is to create a file system within the logical volume and mount it:

```
da10:~ # lvscan
ACTIVE                '/dev/system/data' [100,00 MB] inherit
da10:~ # mkreiserfs /dev/system/data
mkreiserfs 3.6.19 (2003 www.namesys.com)
...
ReiserFS is successfully created on /dev/system/data.
da10:~ # mount /dev/system/data /data
```

As shown above, **lvscan** is used to view the logical volumes. It shows the device to use for the formatting and mounting.

lvextend is used to increase the size of a logical volume. After that you can increase the size of the file system on that logical volume to make use of the additional space.

Before you use **lvreduce** to reduce the size of a logical volume, you have to reduce the size of the file system. Only then reduce the size of the logical volume. If you cut off parts of the file system by simply reducing the size of the logical volume without shrinking the file system first, you will lose data.

Manage Software RAID

To manage software RAID (Redundant Array of Independent (or Inexpensive) Disks), select **RAID** in the YaST Expert Partitioner.

The purpose of RAID is to combine several hard disk partitions into one large virtual hard disk for optimizing performance and improving data security.

There are two types of RAID configurations:

- **Hardware RAID.** The hard disks are connected to a separate RAID controller. The operating system sees the combined hard disks as one device. No additional RAID configuration is necessary at the operating system level.
- **Software RAID.** Hard disks are combined by the operating system. The operating system sees every single disk and needs to be configured to use them as a RAID system.

In the past, hardware RAID provided better performance and data security than software RAID. However, with the current maturity of software RAID in the Linux kernel, software RAID provides comparable performance and data security.

In this section, you learn how to set up software RAID.

You combine hard disks according to RAID levels:

- **RAID 0.** This level improves the performance of your data access, however there is no redundancy in RAID 0. With RAID 0, two or more hard disks are pooled together (striping). Disk performance is very good, but the RAID system is vulnerable to a single point of failure. If one of the disks fails, all data is lost.
- **RAID 1.** This level provides enhanced security for your data because the data is copied to one or several hard disks. This is also known as *hard disk mirroring*. If one disk is destroyed, a copy of its contents is available on the other disk(s). Minimum number of disks (or partitions) required for RAID 1 is two.

- **RAID 5.** RAID 5 is an optimized compromise between RAID 0 and RAID 1 in terms of performance and redundancy. Data and a checksum are distributed across the hard disks. Minimum number of disks (or partitions) required for RAID 5 is three.

If one hard disk fails, it must be replaced as soon as possible to avoid the risk of losing data. The data on the failed disk is reconstructed on its replacement from the data on the remaining disks and the checksum. If more than one hard disk fails at the same time, the data on the disks is lost.

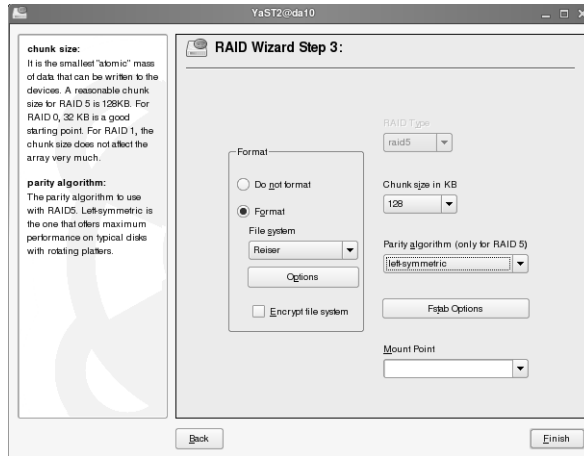
- **RAID 6.** RAID 6 is comparable to RAID 5, the difference being that 2 disks may fail without data loss. The minimum number of disks (or partitions) required for RAID 6 is four.

Using YaST you can set up RAID levels 0, 1, and 5 (RAID levels 2, 3, and 4 are not available with software RAID). To create software RAID with YaST, do the following:

- **Partition your hard disks.** For RAID 0 and RAID 1, at least 2 partitions on different disks are needed. RAID 5 requires at least 3 partitions. We recommend that you use only partitions of the same size.
- **Set up RAID.** Select **RAID** in the YaST Expert Partitioner to open a dialog to choose between the RAID levels 0, 1, and 5, and then add partitions to the new RAID.

Choose a file system and a mount point for your RAID. By changing the chunk size, which is explained in the help text in Figure 2-13, you can fine tune the RAID performance.

Figure 2-13



After finishing the configuration, the RAID partitions appear in the partition list of the Expert Partitioner.



For the purpose of testing, the partitions may reside on a single disk. However, this does not increase any performance or data security.



A RAID is no substitute for a data backup. A RAID does not, for instance, protect files from accidental deletion.

Exercise 2-3 Create Logical Volumes

In this exercise, you learn how to administer LVM using YaST.

You will find this exercise in the workbook.

(End of Exercise)

Objective 5 Set Up and Configure Disk Quotas

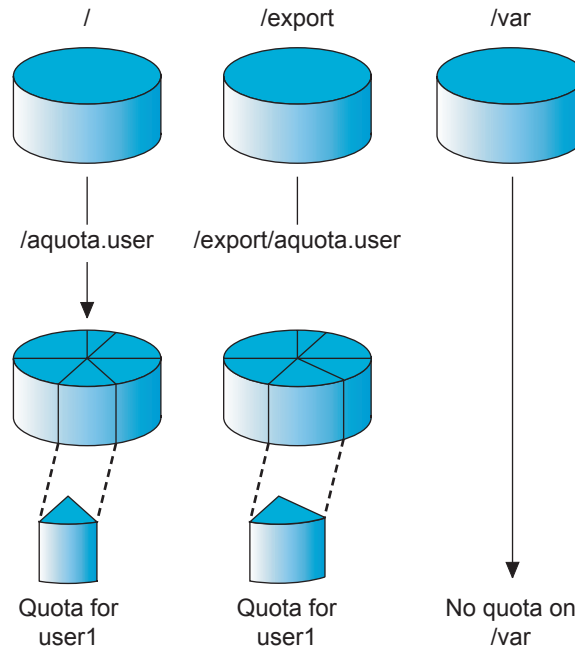
Drive space continues to be a problem. When no limits are imposed, a user can easily fill up hard drive space with all kinds of data.

Linux includes a quota system that allows you to specify a specific amount of storage space each user or group may use, and how many files users or groups may create.

In SUSE Linux Enterprise Server you can use the quota package to establish these limitations.

The following illustrates the quota architecture:

Figure 2-14



You can implement disk quotas for partitions configured with the `ext2`, `ext3`, or Reiser file systems.

Setting up and configuring the disk quota service on your server includes installing the package quota and the following tasks:

- Prepare the File System
- Initialize the Quota System
- Start and Activate the Quota Service
- Configure and Manage User and Group Quotas

Prepare the File System

When the system is started, the quotas for the file system must be activated. You can indicate for which file systems quotas are to be activated by configuring entries in the file `/etc/fstab`.

You enter the keyword **usrquota** for quotas on the user level and the keyword **grpquota** for group quotas, as in the following:

```
/dev/hda2 / reiserfs acl,user_xattr,usrquota,grpquota 1 1
/dev/hda1 swap swap defaults 0 0
proc /proc proc defaults 0 0
...
```

In this example, quotas are configured for the file system `/` (root). Quotas are always defined for file systems (partitions).

If you have configured `/etc/fstab` without rebooting your server, you need to remount the file systems for which quotas have been defined. In the case of quotas for the partition holding the root file system, you do this by using the mount option **(-o) remount** like in the following:

```
da10:~ # mount -o remount /
```

Initialize the Quota System

After remounting, you need to initialize the quota system. You can do this by using the command **quotacheck**. This command checks the partitions with quota keywords in `/etc/fstab` to determine the already occupied data blocks and inodes and stores the determined values in the files **aquota.user** (for user quotas) and **aquota.group** (for group quotas).



Up to kernel version 2.4 these files were called `quota.user` and `quota.group`, and had to be created before `quotacheck` was run.

If you enter the command **quotacheck -avug**, all file systems with the option `usrquota` or `grpquota` in `/etc/fstab` (**-a**) are checked for data blocks and inodes that are occupied by users (**-u**) and groups (**-g**). The option **-v** provides a detailed output.

When checking mounted file systems, you might need to use the option **-m** to force the check.

Assuming the quota entries exist for `/`, after running **quotacheck** the following files are created:

```
dal10:~ # ls -l /aquota* /export/aquota*
-rw----- 1 root root 9216 Aug 27 10:06 /aquota.group
-rw----- 1 root root 9216 Aug 27 10:06 /aquota.user
```

Start and Activate the Quota Service

In order for the quota system to be initialized when the system is booted, the appropriate links must be made in the runlevel directories by entering **insserv boot.quota** (**insserv quotad** for NFS). Runlevels and the command `insserv` are explained in detail in Section 7, “Manage System Initialization” on page 7-1.

You can then start the quota system by entering
/etc/init.d/boot.quota start.

You can also start or stop the quota system by entering one of the following:

/sbin/quotaon filesystem

/sbin/quotaoff filesystem

You can use the option **-a** to activate and deactivate all automatically mounted file systems (except NFS) with quotas.



For additional information on quotaon options, enter **man quotaon.**

Configure and Manage User and Group Quotas

To configure quotas for users and groups, you need to know how to do the following:

- Configure Soft and Hard Limits for Blocks and Inodes
- Configure Grace Periods for Blocks and Inodes
- Copy User Quotas
- Generate a Quota Report

Configure Soft and Hard Limits for Blocks and Inodes

With the command **edquota** and the following options, you can edit the current quota settings for a user or group:

- **edquota -u user:** for setting up user quotas.
- **edquota -g group:** for setting up group quotas. All members of the group together share this quota.

The current settings are displayed in the vi editor for you to edit. You can edit the **soft** and **hard** limits. The values under **blocks** and **inodes** show the currently used blocks and inodes and are for information only; changing them has no effect.

For example, you can enter the following to configure quotas for the user **geeko**:

edquota -u geeko

After entering the command, the following quota information appears in vi:

```
Disk quotas for user geeko (uid 1001):
Filesystem      blocks      soft      hard      inodes      soft      hard
/dev/sda2       7820       10000     20000     145         0         0
```

The following describes the settings:

- **Blocks.** Shows how much hard disk space is currently used, with soft and hard limits listed.

The values for blocks are given in blocks of 1 KB (independent of the block size of the file system).

For example, the value **7820** under Blocks indicates that the user **geeko** is currently using about 8 MB of hard drive space.

Notice that the soft limit is set to **10** MB and the hard limit is set to **20** MB.

- **Inodes.** Indicates how many files belong to the user on the file system, with soft and hard limits listed.

Notice that the soft and hard limits for **geeko** are set to **0**, which means that the user can create an unlimited number of files.

The soft limits indicate a quota that the user cannot permanently exceed. The hard limits indicate a boundary beyond which no more space or inodes can be used.

If users move beyond the soft limit, they have a fixed time available (a grace period) to free up space by deleting files or blocks.

If users exceed the grace period, they cannot create any new files until they delete enough files to get below the soft limit.

Configure Grace Periods for Blocks and Inodes

You can edit the grace periods in vi for blocks and inodes by entering **edquota -t**. A screen similar to the following appears:

```
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
Filesystem      Block grace period   Inode grace period
/dev/sda2        7days                7days
```

You can set the grace periods in days, hours, minutes, or seconds for a listed file system. However, you cannot specify a grace period for a specific user or group.

Copy User Quotas

You can copy user quotas from one user to another by using **edquota -p**.

For example, by entering **edquota -p tux geeko**, you can copy the user quotas for the user **tux** to the user **geeko**.

Generate a Quota Report

The quota system files contain information in binary format about the space occupied by users and groups, and which quotas are set up. You can display this information by using the command **repquota**.

For example, entering **repquota -aug** displays a report similar to the following for all users and groups:

```
*** Report for user quotas on device /dev/sda2
Block grace time: 7days; Inode grace time: 7days

      Block limits
User      used    soft    hard    grace    File limits
-----
root      -- 2646650      0      0      140161      0      0
geeko     +-  20000    10000    20000    7days      146      0      0
```

For additional details on using repquota, enter **man 8 repquota**.

Exercise 2-4 Set Up and Configure Disk Quotas

In this exercise, you learn how to administer quotas.

You will find this exercise in the workbook.

(End of Exercise)

Summary

Objective	Summary
1. Select a Linux File System	<p>Linux supports various file systems. Each file system has its particular strengths and weaknesses, which must be taken into account.</p> <p>File systems that keep a journal of transactions recover faster after a system crash or a power failure.</p>
2. Configure Linux File System Partitions	<p>A basic task of all system administrators is maintaining file system layouts. Under Linux, new partitions can be transparently grafted into existing file system structures using the mount command.</p> <p>In most cases, YaST proposes a reasonable partitioning scheme during installation. However, you can use YaST to customize partitioning during and after installation.</p> <p>To implement partitions on your SUSE Linux Enterprise Server, you learned about design guidelines for implementing partitions and how to administer partitions using YaST or command line tools.</p>

Objective	Summary
3. Manage Linux File Systems	<p>To perform basic Linux file system management tasks in SUSE Linux Enterprise Server, you learned how to use YaST and command line tools to create file systems on partitions.</p> <p>/etc/fstab is the configuration file that holds information about where each partition is to be mounted.</p> <p>mount is the command to attach file systems on partitions to the file system tree; umount detaches them.</p> <p>Various tools exist to monitor, repair and tune filesystems.</p>
4. Configure Logical Volume Manager (LVM) and Software RAID	<p>Logical volume management (LVM) provides a higher-level view of the disk storage on a computer system than the traditional view of disks and partitions.</p> <p>When you create logical volumes with LVM, you can resize and move logical volumes while partitions are still mounted and running.</p> <p>YaST can be used to create, edit or delete the components of LVM.</p> <p>Software RAID allows you to combine several disks to provide increased performance and redundancy.</p>

Objective**Summary**

5. Set Up and Configure Disk Quotas

Linux includes a quota system that lets you specify a specific amount of storage space for each user or group, and how many files that user or members of the group can create.

In this objective, you learned how to perform the following quota management tasks:

- Prepare the File System
 - Initialize the Quota System
 - Configure and Manage User and Group Quotas
 - Start and Activate the Quota Service
-

CNI USE ONLY-1 HARDCOPY PERMITTED

SECTION 3 Administer User Access and Security

In this section you learn how to perform basic user and group management tasks that provide users with a secure and accessible SUSE Linux Enterprise Server environment.

Objectives

1. Configure User Authentication with PAM
2. Manage and Secure the Linux User Environment
3. Use Access Control Lists (ACLs) for Advanced Access Control

Objective 1 **Configure User Authentication with PAM**

User authentication plays a central role in IT security. Linux uses PAM (Pluggable Authentication Modules) in the authentication process as a layer between users and applications. A Linux system administrator can use these modules to configure the way programs should authenticate users.

By providing system-wide access to applications through authentication modules, authentication does not have to be part of each application requiring authentication. The Pluggable Authentication Modules take care of that task for applications.

For example, when a user logs into a Linux system on a virtual terminal, a program called **login** is usually involved in this process.

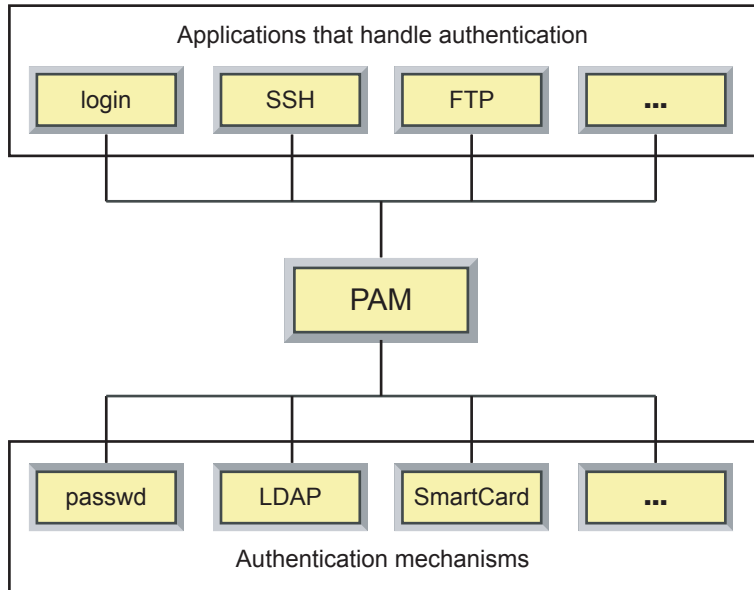
Login requires a user's login name and the password. The password is encrypted and then compared with the encrypted password stored in an authentication database. If the encrypted passwords are identical, login grants the user access to the system by starting the user's login shell.

If other authentication procedures are used, such as smart cards, all programs that perform user authentication must be able to work together with these smart cards. Before PAM was introduced all applications that handle authentication, like login, FTP, or SSH, had to be extended to support a smart card reader.

PAM makes things easier. PAM creates a software level with clearly defined interfaces between applications (such as login) and the current authentication mechanism. Instead of modifying every program, a new PAM module can to enable authentication with a smart card reader. After adjusting the PAM configuration for an application this application can make use of this new authentication method.

The following graphic illustrates the role of PAM:

Figure 3-1



Third party vendors can supply other PAM modules to enable specific authentication features for their products, such as the PAM modules that enable Novell's Linux User Management (LUM) authentication with eDirectory.

To understand how to configure PAM, you need to know the following:

- Location and Purpose of PAM Configuration Files
- PAM Configuration
- PAM Configuration File Examples
- Secure Password Guidelines
- PAM Documentation Resources

Location and Purpose of PAM Configuration Files

PAM provides a variety of modules—each one with a different purpose. For example, one module checks the password, another verifies the location from which the system is accessed, and another reads user-specific settings.

Every program that relies on the PAM modules has its own configuration file `/etc/pam.d/program_name`. For example, the configuration file for the program `passwd` is called `/etc/pam.d/passwd`.

There is one special configuration file with the name **other**. This file contains the default configuration if no application-specific file is found.

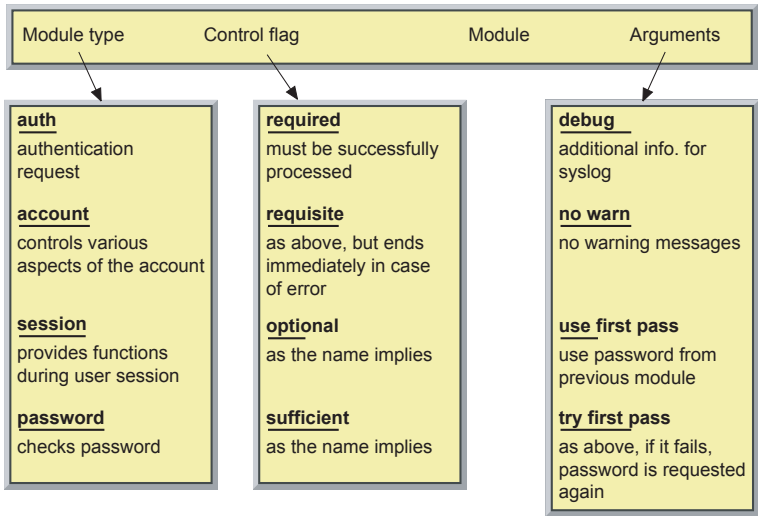
In addition, there are global configuration files for most PAM modules in `/etc/security/`, which define the exact behavior of these modules. These include files such as `pam_env.conf`, `pam_pwcheck.conf`, `pam_unix2.conf`, and `time.conf`.

Every application that uses a PAM module actually calls a set of PAM functions. These functions are implemented in modules which perform the authentication process according to the information in the various configuration files and return the result to the calling application.

PAM Configuration

Each line in a PAM configuration file contains 3 columns plus optional arguments:

Figure 3-2



The following describes the purpose of each column:

- **Module Type.** There are four types of PAM modules:
 - **auth.** These modules provide two ways of authenticating the user.

First, they establish that the user is who he claims to be by instructing the application to prompt the user for a password or other means of identification.

Second, the module can grant group membership or other privileges through its credential granting properties.
 - **account.** These modules perform nonauthentication based account management.

They are typically used to restrict or permit access to a service based on the time of day, currently available system resources (maximum number of users) or perhaps the location of the applicant user (for example, to limit 'root' login to the console).

- **session.** These modules are associated with performing tasks that need to be done for the user before she can be given access to a service or after a service is provided to her.

Such things include logging information concerning the user, mounting directories and the opening and closing of some data exchange with another user.

- **password.** This last module type is required for updating the authentication token associated with the user. Typically, there is one module for each challenge/response-based authentication (auth) module type.

- **Control Flag.** The control flag indicates how PAM will react to the success or failure of the module it is associated with.

Since modules can be stacked (modules of the same type execute in a series, one after another), the control-flags determine the relative importance of each module.

The Linux-PAM library interprets these keywords in the following manner:

- **required.** A module with this flag must be successfully processed before the authentication can proceed.

After the failure of a module with the required flag, all other modules with the same flag are processed before the user receives a message about the failure of the authentication attempt. This prevents the user from knowing at what stage their authentication failed.

- **requisite.** A module with this flag must also be processed successfully. In case of success, other modules are subsequently processed, just like modules with the required flag.

However, in case of failure the module gives immediate feedback to the user and no further modules are processed.

You can use the requisite flag as a basic filter, checking for the existence of certain conditions that are essential for a correct authentication.

- **optional.** The failure or success of a module with this flag does not have any direct consequences.

You can use this flag for modules that are only intended to display a message (such as telling a user that mail has arrived) without taking any further action.

- **sufficient.** After a module with this flag has been successfully processed, the calling application receives an immediate message about the success and no further modules are processed (provided there was no preceding failure of a module with the “required” flag).

The failure of a module with the sufficient flag has no direct consequences. In other words, any subsequent modules are processed in their respective order.

- **include.** This is not really a control flag but indicates that the keyword in the next column is to be interpreted as a file name relative to `/etc/pam.d/` that should be included at this point.

The file included has to have the same structure as any other PAM configuration file.

The purpose of **include** files is to simplify changes concerning several applications.

- **Module.** The PAM modules are located in the directory `/lib/security/`. Every filename of a module starts with the prefix `pam_`. You do not need to include the path, as long as the module is located in the default directory `/lib/security/`.



For all 64 bit platforms supported by SUSE Linux, the default directory is `/lib64/security/`.

Some PAM modules (such as `pam_unix2.so`) can be used for several module types (for instance type `auth` as well as type `password`).

- **Arguments (options).** You can include options in this column for the module, such as **`debug`** (enables debugging) or **`nullok`** (allows the use of empty passwords).

PAM Configuration File Examples

The following is the default configuration file for the login program on SLES 10, `/etc/pam.d/login`:

```
#%PAM-1.0
auth      required      pam_securetty.so
auth      include       common-auth
auth      required      pam_nologin.so
account   include       common-account
password  include       common-password
session   include       common-session
session   required      pam_lastlog.so nowtmp
session   required      pam_resmgr.so
session   optional      pam_mail.so standard
```

As an example of the files included in the above configuration, the file `/etc/pam.d/common-auth` looks like this::

```
#
# /etc/pam.d/common-auth - authentication settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.). The default is to use the
# traditional Unix authentication mechanisms.
#
auth      required      pam_env.so
auth      required      pam_unix2.so
```

CNI USE ONLY-1 HARDCOPY PERMITTED

The modules perform the following tasks (not all are included in the above configuration):

- **auth required pam_securetty.so**

This module checks the file `/etc/securetty` for a list of valid login terminals. If a terminal is not listed in that file, the login is denied from that terminal. This concerns only the root user.

- **auth required pam_env.so**

This module can be used to set additional environment variables. The variables can be configured in the file `/etc/security/pam_env.conf`.

- **auth required pam_unix2.so nullok**

The module `pam_unix2.so` is used during the authentication process to validate the login and password provided by the user.

- **auth required pam_nologin.so**

This module checks whether a file `/etc/nologin` exists. If such a file is found, its content is displayed when a user tries to log in. Login is denied for all but the root user.

- **account required pam_unix2.so**

In this entry the `pam_unix2.so` module is used again, but in this case it checks whether the password of the user is still valid or if the user needs to create a new one.

- **password required pam_pwcheck.so nullok**

This is an entry for a module of the type password. It is used when a user attempts to change the password. In this case, the module `pam_pwcheck.so` is used to check if a new password is secure enough.

The **nullok** argument allows users to change an empty password, otherwise empty passwords are treated as locked accounts.

- **password required pam_unix2.so nullok use_first_pass use_authtok**

The `pam_unix2.so` module is also necessary when changing a password. It encrypts (or hashes, to be more exact) the new password, and writes it to the authentication database.

nullok has the same significance as described above for `pam_pwcheck.so`. With the argument **use_first_pass**, `pam_unix2` uses the password from a previous module, for instance `pam_pwcheck.so`, and aborts with an error if no authentication token from a previous module is available. The argument **use_authtok** is used to force this module to set the new password to the one provided by the previously stacked password module.

- **session required pam_unix2.so**

Here the session component of the `pam_unix2.so` module is used. Without arguments this module has no effect, with the argument **trace** it uses the syslog daemon to log the user's login.

- **session required pam_limits.so**

The `pam_limits.so` sets resource limits for the users that can be configured in the file `/etc/security/limits.conf`.

- **session required pam_mail.so**

This module displays a message if any new mail is in the user's mail box. It also sets an environment variable pointing to the user's mail directory.

Secure Password Guidelines

Even the best security setup for a system can be defeated if users choose easy to guess passwords. With today's computing power, a simple password can be cracked within minutes.

These attacks are also called *dictionary attacks*, as the password cracking program just tries one word after another from a dictionary file, including some common variations of these words.

Therefore, a password should never be a word which could be found in a dictionary. A good, secure password should always contain numbers and uppercase characters.

To check whether user passwords fulfill this requirement, you can enable a special PAM module to test a password first before a user can set it. The PAM module is called `pam_pwcheck.so` and uses the `cracklib` library to test the security of passwords.

By default, this PAM module is enabled on SLES 10.

If a user enters a password that is not secure enough, the following message is displayed:

Bad password: too simple

and the user is prompted to enter a different one.

There are also dedicated password check programs available such as **John the Ripper** (<http://www.openwall.com/john/>).

PAM Documentation Resources

The following PAM documentation is available in the directory `/usr/share/doc/packages/pam/`:

- **READMEs.** In the top level of this directory, there are some general README files. The subdirectory `modules/` holds README files about the available PAM modules.
- **The Linux-PAM System Administrators' Guide.** This document includes everything that a system administrator should know about PAM.

The document discusses a range of topics, from the syntax of configuration files to the security aspects of PAM. The document is available as a PDF file, in HTML format, and as plain text.

- **The Linux-PAM Module Writers' Manual.** This document summarizes the topic from the developer's point of view, with information about how to write standard-compliant PAM modules. It is available as a PDF file, in HTML format, and as plain text.
- **The Linux-PAM Application Developers' Guide.** This document includes everything needed by an application developer who wants to use the PAM libraries. It is available as a PDF file, in HTML format, and as plain text.

There are also manual pages for some PAM modules, such as **`man pam_unix2`**.

Exercise 3-1 Configure PAM Authentication

In this exercise, you practice configuring PAM authentication.

You will find this exercise in the workbook.

(End of Exercise)

Objective 2 **Manage and Secure the Linux User Environment**

Besides managing individual user accounts, you also need to know how to do the following to manage and secure the Linux user environment:

- Perform Administrative Tasks as root
- Delegate Administrative Tasks With `sudo`
- Set Defaults for New User Accounts
- Configure Security Settings

Perform Administrative Tasks as root

As a system administrator, you are advised to log in as a normal user and only switch to root to perform tasks that require root permissions.

To switch between a normal user and root while performing administrative tasks, you can do the following:

- Switch to Another User With `su`
- Switch to Another Group With `newgrp`
- Start Programs as Another User From Gnome

Switch to Another User With `su`

You can use the command **`su`** (switch user) to assume the UID of root or of other users.

The following is the `su` syntax:

`su [options] ...[-] [user[argument]]`

For example, to change to the user **geeko**, enter **su geeko**; to change to the user **root**, enter **su root** or **su** (without a user name). If you want to start a login shell when changing to the user **root**, you can enter **su -**.



Root can change to any user ID without knowing the password of the user.

To return to your previous user ID, enter **exit**.

To change to the user **root** and execute a single command, use the option **-c**:

```
geeko@da10:~> su - -c "grep geeko /etc/shadow"
```



For additional information on the command **su**, enter **su --help**.

Switch to Another Group With **newgrp**

A user can be a member of many different groups, but only one GID is his *effective* (current) group at any one time. Normally this is the *primary group*, which is specified in the file `/etc/passwd`.

If a user creates directories or files, then they belong to the user and to the effective group.

You can change the effective group GID with the command **newgrp** or **sg** (such as **sg video**).

Only group members may perform this group change, unless a group password is defined. In this case, any user that knows the group password can make the change too.

You can undo the change (return to the original effective GID) by entering **exit** or by pressing **Ctrl+D**.

Start Programs as Another User From Gnome

In Gnome you can start any program with a different UID (as long as you know the password), using the program **gnomesu**.

From the Gnome desktop, open a command line dialog by pressing **Alt+F2**; then enter **gnomesu**. You are prompted for the root password, and after entering it a terminal window opens up. The path is still that of the user logged in to Gnome; if you need the standard environment for root, enter **su -** in the terminal window.

You can specify a different user than root and also start a program directly with the following syntax: **gnomesu -u user command**. If the command is not in the path of the user logged in to Gnome, you have to enter the full path, like **gnomesu /sbin/yast2**, which starts YaST after the root password is entered.



For some programs it is not necessary to use **gnomesu** after entering **Alt+F2**; for instance when you enter **yast2** directly you are automatically prompted for the root password.

Delegate Administrative Tasks With sudo

Sometimes it is necessary to allow a normal user access to a command which is usually reserved for root. For example, you might want a co-worker to take over tasks such as shutting down the computer and creating users while you are on vacation, without sharing the root password.

The default configuration of `sudo` in SLES 10 requires the knowledge of the root password. If you know the root password, you actually would not need to use `sudo` for administrative tasks. Its use has the advantage that the commands executed are logged to `/var/log/messages` and that you do not need to retype the password for each command (as with `su -c` command), because it is cached for some minutes by `sudo`.

```
geeko@da10:~ > sudo /sbin/shutdown -h now

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

root's password:
```

You can change the configuration of `sudo` so that it asks for the user password instead of the root password. In order to do this, put a comment sign (`#`) in front of the following two lines in `/etc/sudoers`, using the command `visudo`:

```
# In the default (unconfigured) configuration, sudo asks for the root
# password. This allows use of an ordinary user account for administration
# of a freshly installed system. When configuring sudo, delete the two
# following lines:
Defaults targetpw      # ask for the password of the target user i.e. root
ALL ALL=(ALL) ALL      # WARNING! Only use this together with 'Defaults
                        # targetpw'!
```

Using `visudo`, you can specify which commands a user can or cannot enter by configuring the file `/etc/sudoers`.

The following is the general syntax of an entry in the configuration file:

user/group host = command1, command2 ...

For example

```
geeko ALL = /sbin/shutdown
```

In this example, the user **geeko** is able to carry out the command **/sbin/shutdown** with the permissions of root on all computers (**ALL**). Being able to specify the computer in `/etc/sudoers` allows to copy the same file to different computers without having to grant the same permissions on all computers involved.

The following is a more complex example that illustrates the flexibility of sudo:

Figure 3-3

```
1 User_Alias  ADMINS      = tux, geeko
2 User_Alias  WEBMASTER  = john
3 User_Alias  SUBSTITUTE = olli, klaas
4
5 # Cmnd alias specification
6
7 Cmnd_Alias  PRINTING    = /usr/sbin/lpc, /usr/bin/lprm
8 Cmnd_Alias  SHUTDOWN    = /sbin/shutdown
9 Cmnd_Alias  APACHE      = /etc/init.d/apache
10
11
12 # User privilege specification
13 root      ALL=(ALL) ALL
14
15 ADMINS     ALL = NOPASSWD: !/usr/bin/passwd, /usr/bin/passwd [A-z]*,
16 !/usr/bin/passwd root
17 WEBMASTER ALL = APACHE
18 SUBSTITUTE ALL = SHUTDOWN, PRINTING
```

Lines 1 to 9 define aliases. You can do this for the following:

- Users with `User_Alias` (lines 1–3)
- Commands with `Cmnd_Alias` (lines 7–9)
- Hosts with `Host_Alias`

Lines 14–17 in this example show how these aliases can be used in the actual rules:

- **ADMINS**. This is the `User_Alias` for the users `tux` and `geeko` (see line 1).

The following are additional parameters:

- ❑ **!/usr/bin/passwd, /usr/bin/passwd [A-z]*.** This indicates that both users are allowed to run the command `passwd` with one single argument and change the passwords for user accounts.
- ❑ **!/usr/bin/passwd root.** This indicates that both users are not allowed to change the password for root. However, they can change the passwords of other users.



With this configuration, tux and geeko could still lock out root by entering **`sudo /usr/bin/passwd root -l`**.

- **WEBMASTER.** This is the User_Alias for the user account john (see line 2). This user can start and stop the web server (APACHE).
- **SUBSTITUTE.** This is the User_Alias for the user accounts olli and klaas (see line 3). These users can execute commands summarized in sections SHUTDOWN and PRINTING (see lines 7 and 8).



For additional documentation and configuration examples, enter **`man 5 sudoers`**.

Set Defaults for New User Accounts

You can use YaST to select default settings to be applied to new user accounts.

From the Gnome desktop, press **Alt+F2**, enter **yast2** and enter the root password when prompted. Select **Security and Users > User Management**. You can also start the User Management module directly from a terminal window as root by entering **yast2 users**.

Select **Expert Options > Defaults for New Users**. The following appears:

Figure 3-4

Here, set default values to use when creating new local or system users.

Default Group
The group name of a new user's primary group.

Secondary Groups
Names of additional groups to which to assign new users.

Default Login Shell
The name of the new user's login shell. Select one from the list or enter your own path to the shell.

Default Home
The initial path prefix for a new user's home directory. The username is added to the end of this value to create the default name of the home directory.

Skeleton Directory
The contents of this directory are copied to a user's home directory when a new user is added.

Expiration Date
The date on which the user account is disabled. The date must be in the format YYYY-MM-DD. Leave it empty if this account never expires.

Default Group
users

Secondary Groups
distout,video

Default Login Shell
/bin/bash

Path Prefix for Home Directory
/home/

Skeleton for Home Directory
/etc/skel

Default Expiration Date

Days after Password Expiration Login Is Usable
-1

You can enter or edit information in the following fields:

- **Default Group.** From the drop-down list select the primary (default) group.
- **Secondary Groups.** Enter a list of secondary groups (separated by commas) to assign to the user.
- **Default Login Shell.** From the drop-down list select the default login shell (command interpreter) from the shells installed on your system.

- **Default Home.** Enter or browse to the initial path prefix for a new user's home directory. The user's name will be appended to the end of this value to create the default name of the user's home directory.
- **Skeleton Directory.** Enter or browse to the skeleton directory. The contents of this directory will be copied to the user's home directory when you add a new user.
- **Default Expiration Date.** Enter the date on which the user account is disabled. The date must be in the format YYYY-MM-DD.

Leave the field empty if this account never expires.
- **Days after Password Expiration Login Is Usable.** This setting enables users to log in after passwords expire. Set how many days login is allowed after a password expired.

Enter **-1** for unlimited access.

Save the configuration settings by selecting **Next > Finish**. The values are written to the file **/etc/default/useradd**:

```
da10:~ # cat /etc/default/useradd
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
GROUPS=video,dialout
CREATE_MAIL_SPOOL=no
```

You can also use the command line program **useradd** to view or change the defaults. The option **--show-defaults** displays the same as the cat above. The option **--save-defaults** followed by an option with a value changes them:

```
da10:~ # useradd --save-defaults -d /export/home
da10:~ # useradd --show-defaults
GROUP=100
HOME=/export/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
GROUPS=video,dialout
CREATE_MAIL_SPOOL=no
```

The manual page for **useradd** lists the possible options.

Configure Security Settings

YaST provides a Local Security module that lets you configure the following local security settings for your SUSE Linux Enterprise Server:

- Password settings
- Boot configuration
- Login settings
- User creation settings
- File permissions

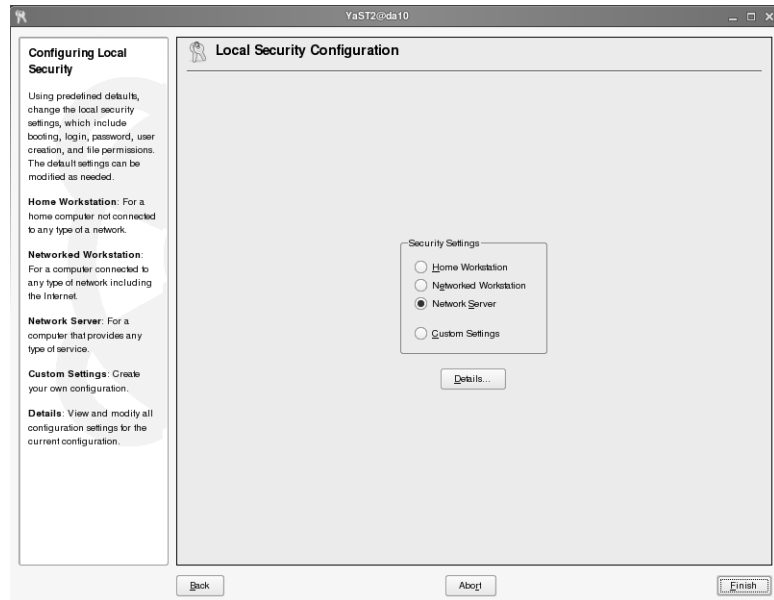
You can select from (or modify) three preset levels of security, or create your own customized security settings to meet the requirements of your enterprise security policies and procedures.

You can access the Security Settings module from the YaST Control Center by selecting **Security and Users > Local Security**, or by entering as root **yast2 security** in a terminal window.

CNI USE ONLY-1 HARDCOPY PERMITTED

The following appears:

Figure 3-5



From this dialog, you can select one of the following preset configurations:

- **Home Workstation.** Select for a home computer not connected to any type of a network. This option represents the lowest level of local security.
- **Networked Workstation.** Select for a computer connected to any type of a network or the Internet. This option provides an intermediate level of local security.
- **Network Server.** Select for a computer that provides any type of service (network or otherwise). This option enables a high level of local security.
- You can also select **Details** or **Custom Settings** to modify an existing security level or create your own configuration.

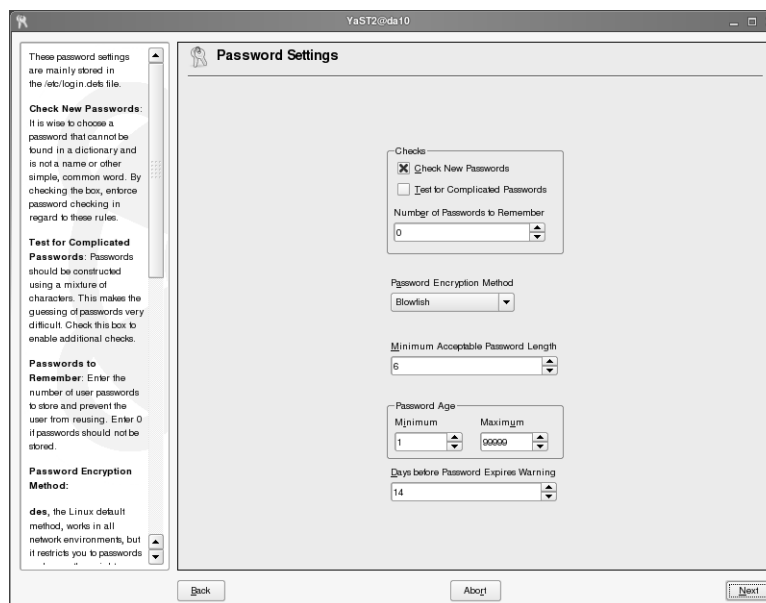
By selecting one of the three predefined security levels and selecting **Next**, the chosen security level is applied. By selecting **Details**, you can change the settings for the security level you have selected.

If you choose the **Customs Settings** and then select **Next**, you can directly change the details of the security configuration.

The dialogs for the detail settings look the same for every security level, but the preselected options are different. In the following dialogs, you see the settings for Level 3 (Network Server).

In the first dialog you can change the default password requirements that are accepted by the systems:

Figure 3-6



From this dialog, you can select or enter the following password settings (mainly stored in `/etc/login.defs`, some values also in `/etc/default/passwd` and `/etc/security/pam_pwcheck.conf`):

- **Check New Passwords.** It is important to choose a password that cannot be found in a dictionary and is not a name or other simple, common word. By selecting this option, you enforce password checking in regard to these rules.
- **Test for Complicated Passwords.** Passwords should be constructed using a mixture of uppercase and lower case characters as well as numbers. Special characters like ;(= etc. may be used too, but could be hard to enter on a different keyboard layout. This makes it very difficult to guess the password. Select this option to enable additional checks.
- **Password Encryption Method.** From the drop-down list, select one of the following encryption methods:
 - **DES.** This is the lowest common denominator. It works in all network environments, but it restricts you to passwords no longer than eight characters. If you need compatibility with other systems, select this method.
 - **MD5.** This encryption method allows longer passwords and is supported by all current Linux distributions, but not by other systems or older software.
 - **Blowfish.** This encryption method uses the blowfish algorithm to encrypt passwords. It is not yet supported by many systems. A lot of CPU power is needed to calculate the hash, which makes it difficult to crack passwords with the help of a dictionary. It is used as default encryption method on SLES 10
- **Minimum Acceptable Password Length.** Enter the minimum number of characters for an acceptable password. If a user enters fewer characters, the password is rejected.

Entering **0** disables this check.
- **Password Age.** Minimum refers to the number of days that have to elapse before a password can be changed again. Maximum is the number of days after which a password expires and must be changed.

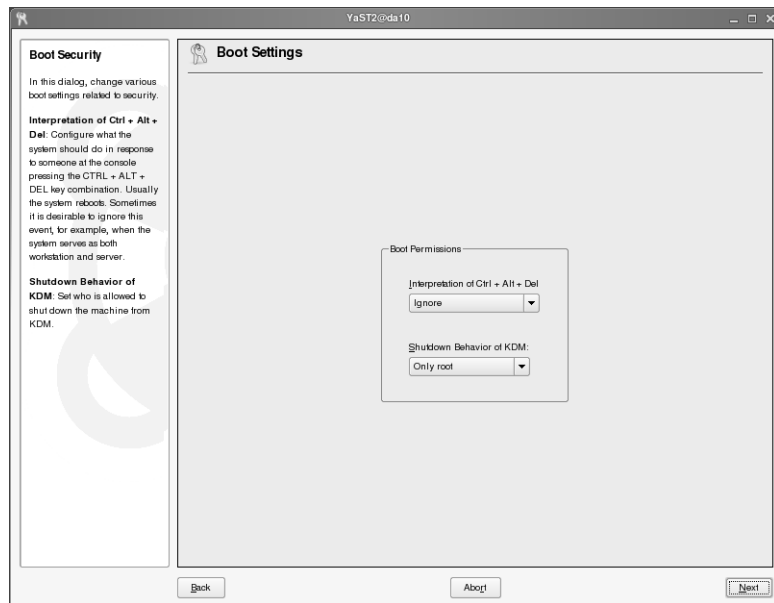
- **Days Before Password Expires Warning.** A warning is issued to the user this number of days before password expiration.



Although root receives a warning when setting a password, she can still enter a bad password despite the above settings.

When you finish configuring password settings, continue by selecting **Next**. The following appears:

Figure 3-7



From this dialog, you can select the following boot settings (which update the file `/etc/inittab`):

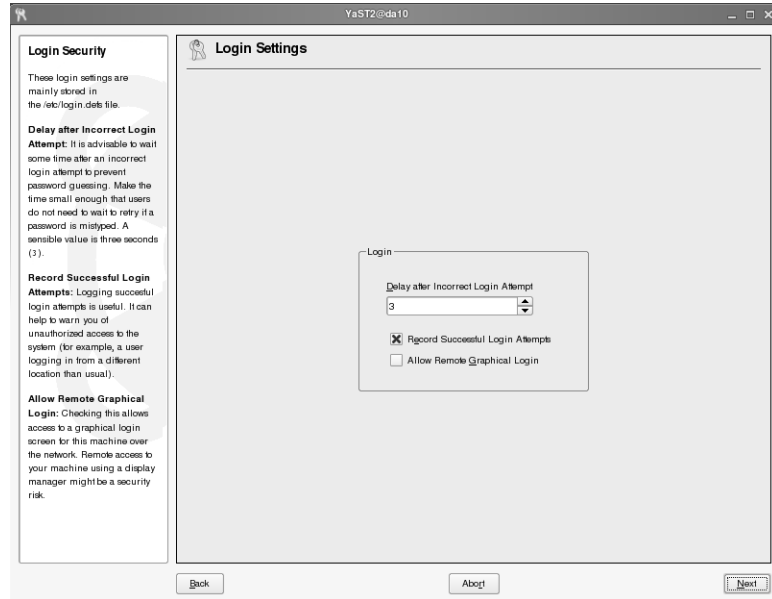
- **Interpretation of Ctrl + Alt + Del.** When someone at the console presses the **Ctrl+Alt+Del** keystroke combination, the system usually reboots.

- ❑ **Ignore.** Sometimes you want to have the system ignore this keystroke combination, especially when the system serves as both workstation and server. Nothing happens when the **Ctrl+Alt+Del** keystroke combination is pressed.
- ❑ **Reboot.** The system reboots when the **Ctrl+Alt+Del** keystroke combination is pressed.
- ❑ **Halt.** The system is shut down when the **Ctrl+Alt+Del** keystroke combination is pressed.
- **Shutdown Behavior of KDM.** You use this option to set who is allowed to shut down the computer from KDM.
 - ❑ **Only Root.** To halt the system, the root password has to be entered.
 - ❑ **All Users.** Everyone, even remotely connected users, can halt the system using KDM.
 - ❑ **Nobody.** Nobody can halt the system with KDM.
 - ❑ **Local Users.** Only locally connected users can halt the system with KDM.
 - ❑ **Automatic.** The system is halted automatically after log out.

For a server system you should use **Only Root** or **Nobody** to prevent normal or even remote users from halting the system

When you finish configuring boot settings, continue by selecting **Next**. The following appears:

Figure 3-8



From this dialog, you can enter and select the following login settings (mainly stored in `/etc/login.defs`):

- **Delay After Incorrect Login Attempt.** Following a failed login attempt, there is typically a waiting period of a few seconds before another login is possible. This makes it more difficult for password crackers to log in.

This option lets you adjust the time delay before another login attempt. Default is 3 seconds, which is a reasonable value.

- **Record Successful Login Attempts.** Recording successful login attempts can be useful, especially in warning you of unauthorized access to the system (such as a user logging in from a different location than normal).

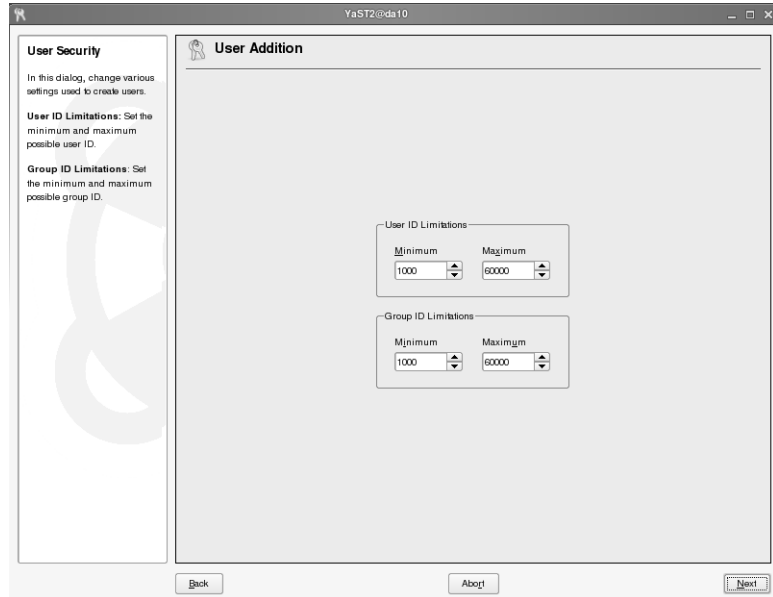
Select this option to record successful login attempts in the file **/var/log/wtmp**. You can use the command **last** to view who logged in at what time.

- **Allow Remote Graphical Login.** You can select this option to allow other users access to your graphical login screen via the network.

Because this type of access represents a potential security risk, it is inactive by default.

When you finish configuring login settings, continue by selecting **Next**. The following appears:

Figure 3-9

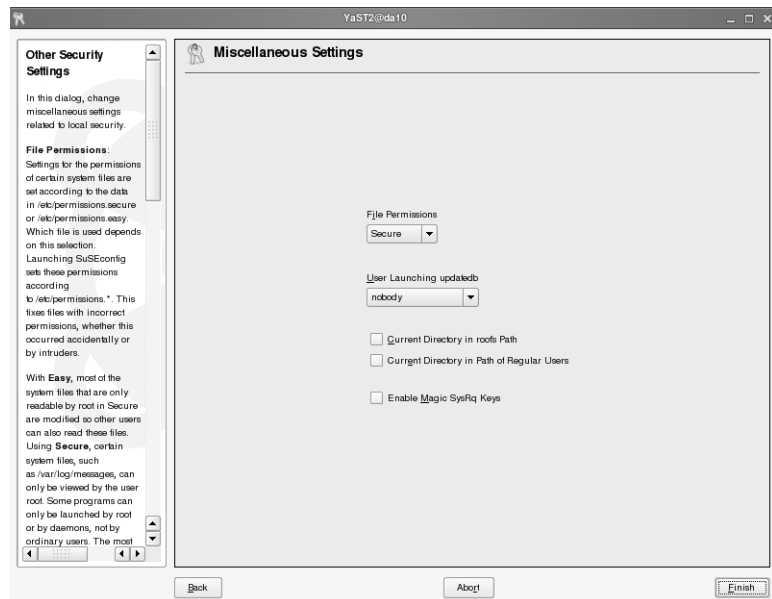


From this dialog, you can enter the following ID settings (stored in **/etc/login.defs**):

- **User ID Limitations.** Enter a minimum and maximum value to configure a range of possible user ID numbers. New users get a UID from within this range.
- **Group ID Limitations.** Enter a minimum and maximum value to configure a range of possible group ID numbers.

When you finish configuring user and group ID limitations, continue by selecting **Next**. The last page of the security configuration appears:

Figure 3-10



From this dialog, you can select the following miscellaneous global settings:

- **File Permissions.** Settings for the permissions of certain system files are configured in **/etc/permissions.easy**, **/etc/permissions.secure**, or **/etc/permissions.paranoid**. You can also add your own rules to the file **/etc/permissions.local**. Each file contains a description of the file syntax and purpose of the preset.

Settings in files in the directory **/etc/permissions.d/** are included as well. This directory is used by packages that bring their own permissions files.

From the drop-down list, select one of the following:

- **Easy.** Select this option to allow read access to most of the system files by users other than root.
- **Secure.** Select this option to make sure that certain configuration files (such as **/etc/ssh/ssh_config**) can only be viewed by the user root. Some programs can only be launched by root or by daemons, not by an ordinary user.
- **Paranoid.** Select this option for an extremely secure system. All SUID/SGID-Bits on programs have been cleared. Remember that some programs might not work correctly, because users no longer have the permissions to access certain files.

Running SuSEconfig sets these permissions according to the settings in the respective **/etc/permissions*** files. This fixes files with incorrect permissions, whether this occurred accidentally or by intruders.

- **User Launching updatedb.** If the program **updatedb** is installed, it automatically runs on a daily basis or after booting. It generates a database (**locatedb**) in which the location of each file on your computer is stored.

You can search this database with the utility **locate** (enter **man locate** for details).

From the drop-down list, select one of the following:

- ❑ **nobody.** Any user can find only the paths in the database that can be seen by any other (unprivileged) user.
- ❑ **root.** All files in the system are added into the database.
- **Current Directory in root's Path and Current Directory in the Path of Regular Users.**

If you deselect these options (the default), users must always launch programs in the current directory by adding “.” (such as **./configure**).

If you select these options, the dot (“.”) is appended to the end of the search path for root and users, allowing them to enter a command in the current directory without appending “.”.

Selecting these options can be very dangerous because users can accidentally launch unknown programs in the current directory instead of the usual system-wide files.

This configuration is written to **/etc/sysconfig/suseconfig**.

- **Enable Magic SysRq Keys.** Selecting this option gives you some control over the system even if it crashes (such as during kernel debugging). For details, see [/usr/src/linux/Documentation/sysrq.txt](#).

This configuration is written to **/etc/sysconfig/sysctl**.

When you finish configuring the miscellaneous settings, save the settings and run SuSEconfig by selecting **Finish**.

Exercise 3-2 Configure the Password Security Settings

In this exercise, you practice changing different security settings.

You will find this exercise in the workbook.

(End of Exercise)

Objective 3 **Use Access Control Lists (ACLs) for Advanced Access Control**

To use ACLs for advanced file system access control you need to know the following:

- The Basics of ACLs
- Basic ACL commands
- Important ACL Terms
- ACL Types
- How ACLs and Permission Bits Map to Each Other
- How to Use the ACL Command Line Tools
- How to Configure a Directory with an Access ACL
- How to Configure a Directory with a Default ACL
- The ACL Check Algorithm
- How Applications Handle ACLs

The Basics of ACLs

Traditionally, 3 sets of permissions are defined for each file object on a Linux system. These sets include the read (r), write (w), and execute (x) permissions for each of three types of users: the file owner, the group, and other users.

This concept is adequate for most practical cases. In the past however, for more complex scenarios or advanced applications, system administrators had to use a number of tricks to circumvent the limitations of the traditional permission concept.

ACLs (Access Control Lists) provide an extension of the traditional file permission concept. They allow you to assign permissions to individual users or groups even if these do not correspond to the original owner or the owning group.

ACLs are a feature of the Linux kernel and are supported by the ReiserFS, Ext2, Ext3, JFS, and XFS file systems. Using ACLs, you can create complex scenarios without implementing complex permission models on the application level.

The advantages of ACLs are clearly evident in situations like replacing a Windows server with a Linux server providing file and print services with Samba.

Since Samba supports ACLs, user permissions can be configured both on the Linux server and in Windows with a graphical user interface (only on Windows NT and later).

Basic ACL commands

There are two basic commands for ACLs: **setfacl** (set file ACLs) to set and **getfacl** (get file ACLs) to view the ACLs of a file or directory.

Allowing write access to a file to one single user besides the owning user is a simple scenario where ACLs come in handy. Using the conventional approach, you would have to create a new group, make the two users involved members of the group, change the owning group of the file to the new group and then grant write access to the file for the group. root access would be required to create the group and to make the two users members of that group.

With ACLs you can achieve the same by making the file writable for the owner plus the named user:

```
geeko@dal0:~> touch file
geeko@dal0:~> ls -l file
-rw-r--r-- 1 geeko users 0 2006-05-22 15:08 file
geeko@dal0:~> setfacl -m u:tux:rw file
geeko@dal0:~> ls -l file
-rw-rw-r--+ 1 geeko users 0 2006-05-22 15:08 file
geeko@dal0:~> getfacl file
# file: file
# owner: geeko
# group: users
user::rw-
user:tux:rw-
group::r--
mask::rw-
other::r--
```

Another advantage of this approach is that the system administrator does not have to get involved to create a group. The user can decide on his own whom he grants access to his files.

Note that the output of **ls** changes when ACLs are used (see the second output of **ls** above). A **+** is added to alert to the fact that ACLs are defined for this file, and the permissions displayed for the group have a different significance. They display the value of the ACL mask now, and no longer the permissions granted to the owning group.

Important ACL Terms

The following list defines terms concerning ACLs:

- **user class.** The conventional POSIX permission concept uses three classes of users for assigning permissions in the file system: the owning user, the owning group, and other users.

Three permission bits can be set for each user class, giving permission to read (r), write (w), and execute (x).

- **access ACL.** Determine access permissions for users and groups for all kinds of file system objects (files and directories).
- **default ACL.** Default ACLs can only be applied to directories. They determine the permissions a file system object inherits from its parent directory when it is created.
- **ACL entry.** Each ACL consists of a set of ACL entries. An ACL entry contains a type, a qualifier for the user or group to which the entry refers, and a set of permissions. For some entry types, the qualifier for the group or users is undefined.

ACL Types

There are two basic classes of ACLs:

- **Minimum ACL.** A minimum ACL includes the entries for the types: owning user, owning group, and other. These correspond to the conventional permission bits for files and directories.
- **Extended ACL.** An extended ACL goes beyond this. It contains a mask entry and can contain several entries of the named user and named group types.

ACLs extend the classic Linux file permission by the following permission types:

- **named user.** With this type, you can assign permissions to individual users.
- **named group.** With this type, you can assign permissions to individual groups.
- **mask.** With this type, you can limit the permissions of named users or groups.

The following is an overview of all possible ACL types:

Table 3-1

Type	Text Form
owner	user::rwx
named user	user:name:rwx
owning group	group::rwx
named group	group:name:rwx
mask	mask::rwx
other	other::rwx

The permissions defined in the entries owner and other are always effective. Except for the mask entry, all other entries (named user, owning group, and named group) can be either effective or masked.

If permissions exist in the named user, owning group, or named group entries as well as in the mask, they are effective (logical AND). Permissions contained only in the mask or only in the actual entry are not effective.

The following example determines the effective permissions for the user jane:

Table 3-2

Entry Type	Text Form	Permissions
named user	user:jane:r-x	r-x
mask	mask::rw-	rw
Effective permissions:		r--

The ACL contains two entries, one for the named user jane and one mask entry. Jane has permissions to read and execute the corresponding file, but the mask only contains permissions for reading and writing.

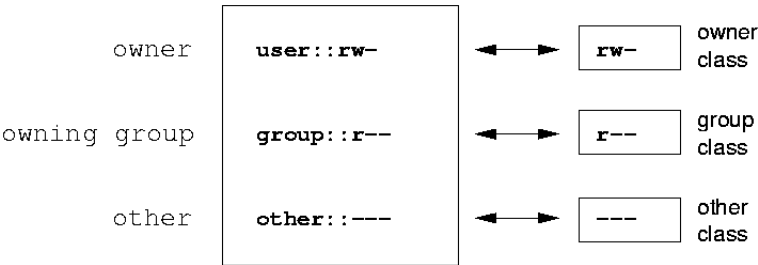
Because of the AND combination, the effective rights allow jane to only read the file.

How ACLs and Permission Bits Map to Each Other

When you assign an ACL to a file or directory, the permissions set in the ACL are mapped to the standard UNIX permissions.

The following figure illustrates the mapping of a minimum ACL:

Figure 3-11

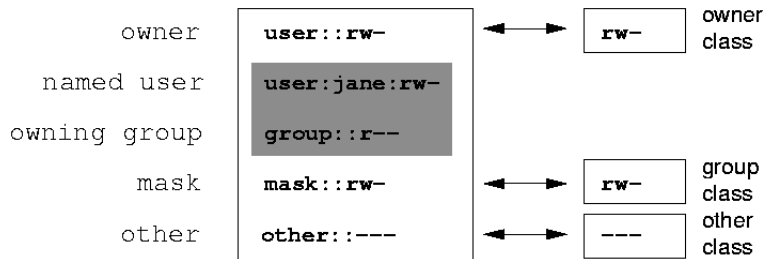


The figure is structured in three blocks:

- The left block shows the type specifications of the ACL entries.
- The center block displays an example ACL.
- The right block shows the respective permission bits according to the conventional permission concept as displayed by ls -l, for example.

The following is an example of an extended ACL:

Figure 3-12



In both cases (minimum ACL and extended ACL), the owner class permissions are mapped to the ACL entry owner. Other class permissions are mapped to their respective ACL entries. However, the mapping of the group class permissions is different in the second case.

In the case of a minimum ACL without a mask, the group class permissions are mapped to the ACL entry owning group. In the case of an extended ACL with a mask, the group class permissions are mapped to the mask entry.

This mapping approach ensures the smooth interaction of applications, regardless of whether they have ACL support or not.

The access permissions that were assigned by permission bits represent the upper limit for all other adjustments made by ACLs.

Any permissions not reflected here are either not in the ACL or are not effective. Changes made to the permission bits are reflected by the ACL and vice versa.

How to Use the ACL Command Line Tools

To manage the ACL settings, you can use the following command line tools:

- **getfacl.** The command `getfacl` can be used to display the ACL of a file.
- **setfacl.** The command `setfacl` can be used to change the ACL of a file.

The following are the most important options for the `setfacl` command:

Table 3-3

Option	Description
-m	Adds or modifies an ACL entry.
-x	Removes an ACL entry.
-d	Sets a default ACL.
-b	Removes all extended ACL entries.

The options `-m` and `-x` expect an ACL definition on the command line. The following are the definitions for the extended ACL types:

- **named user.** The following is an example entry for the user `tux`: **`setfacl -m u:tux:rx my_file`**

The user `tux` gets read and execute permissions for the file `my_file`.

- **named groups.** The following is an example entry for the group `accounting`: **`setfacl -m g:accounting:rw my_file`**

The group `accounting` gets read and write permissions for the file `my_file`.

- **mask.** Sets the ACL mask: **`setfacl -m m:rx`**

Sets the mask for the read and execute permissions.

How to Configure a Directory with an Access ACL

To configure a directory with ACL access, do the following:

1. Before you create the directory, use the `umask` command to define which access permissions should be masked each time a file object is created.

The command **`umask 027`** sets the default permissions by giving the owner the full range of permissions (0), denying the group write access (2), and giving other users no permissions at all (7).

`umask` actually masks the corresponding permission bits or turns them off.



For more information about `umask`, see the corresponding man page `man umask`.

The command **`mkdir mydir`** should create the `mydir` directory with the default permissions as set by `umask`. Enter the following command to check if all permissions were assigned correctly:

```
geeko@dal0:~> umask 027
geeko@dal0:~> mkdir mydir
geeko@dal0:~> ls -dl mydir
drwxr-x--- ... geeko project3 ... mydir
```

2. Check the initial state of the ACL by entering the following command:

```
geeko@dal0:~> getfacl mydir
# file: mydir
# owner: geeko
# group: project3
user::rwx
group::r-x
other::---
```

The output of `getfacl` precisely reflects the mapping of permission bits and ACL entries as described before. The first three output lines display the name, owner, and owning group of the directory.

The next three lines contain the three ACL. In fact, in the case of this minimum ACL, the `getfacl` command does not produce any information you could not have obtained with `ls`.

Your first modification of the ACL is the assignment of read, write, and execute permissions to an additional user `jane` and an additional group `jungle` by entering the following:

```
geeko@da10:~> setfacl -m user:jane:rwx,group:jungle:rwx mydir
```

The option `-m` prompts `setfacl` to modify the existing ACL. The following argument indicates the ACL entries to modify (several entries are separated by commas). The final part specifies the name of the directory to which these modifications should be applied.

Use the `getfacl` command to take a look at the resulting ACL:

```
geeko@da10:~> getfacl mydir
# file: mydir
# owner: geeko
# group: project3
user::rwx
user:jane:rwx
group::r-x
group:jungle:rwx
mask::rwx
other:---
```

In addition to the entries initiated for the user `jane` and the group `jungle`, a mask entry has been generated.

This mask entry is set automatically to reduce all entries in the group class to a common denominator. In addition, setfacl automatically adapts existing mask entries to the settings you modified, provided you do not deactivate this feature with `-n`.

The mask type defines the maximum effective access permissions for all entries in the group class. This includes named user, named group, and owning group.

The group class permission bits that would be displayed by `ls -dl mydir` now correspond to the mask entry:

```
geeko@dal10:~> ls -dl mydir
drwxrwx---+ ... geeko project3 ... mydir
```

The first column of the output now contains an additional `+` to indicate that there is an extended ACL for this item.

3. According to the output of the `ls` command, the permissions for the mask entry include write access. Traditionally, such permission bits would mean that the owning group (in this example `project3`) also has write access to the directory `mydir`.

However, the effective access permissions for the owning group correspond to the overlapping portion of the permissions defined for the owning group and for the mask, which is `r-x` in the example.

As far as the effective permissions of the owning group are concerned, nothing has changed even after adding the ACL entries.

In the following example, the write permission for the owning group is removed with the `chmod` command:

```
geeko@dal0:~> chmod g-w mydir
geeko@dal0:~> ls -dl mydir
drwxr-x---+ ... geeko project3 ... mydir
geeko@dal0:~> getfacl mydir
# file: mydir
# owner: geeko
# group: project3
user::rwx
user:jane:rwx    # effective: r-x
group::r-x
group:jungle:rwx # effective: r-x
mask::r-x
other:---
```

After executing the `chmod` command to remove the write permission from the group class bits, the output of the `ls` command is sufficient to see that the mask bits have changed accordingly: write permission is again limited to the owner of `mydir`.

The output of the `getfacl` confirms this. This output includes a comment for all those entries in which the effective permission bits do not correspond to the original permissions because they are filtered according to the mask entry.

The original permissions can be restored at any time with `chmod`:

```
geeko@dal0:~> chmod g+w mydir
geeko@dal0:~> ls -dl mydir
drwxrwx---+ ... geeko project3 ... mydir
geeko@dal0:~> getfacl mydir
# file: mydir
# owner: geeko
# group: project3
user::rwx
user:jane:rwx
group::r-x
group:jungle:rwx
mask::rwx
other:---
```

You can change the mask with `setfacl` as well, using **`setfacl -m m::rwx`**. The following removes write access from the mask using `setfacl`, with the same result as **`chown g-w`** above:

```
geeko@dal0:~> setfacl -m m::rx mydir
geeko@dal0:~> ls -dl mydir
drwxr-x---+ ... geeko project3 ... mydir
geeko@dal0:~> getfacl mydir
# file: mydir
# owner: geeko
# group: project3
user::rwx
user:jane:rwx      # effective: r-x
group::r-x
group:jungle:rwx  # effective: r-x
mask::r-x
other:---
```

How to Configure a Directory with a Default ACL

Directories can have a default ACL, which is a special kind of ACL that defines the access permissions that objects under the directory inherit when they are created. A default ACL affects subdirectories as well as files.

There are two different ways in which the permissions of a directory's default ACL are passed to the files and subdirectories in it:

- A subdirectory inherits the default ACL of the parent directory both as its own default ACL and as an access ACL.
- A file inherits the default ACL as its own access ACL.

All system functions that create file system objects use a mode parameter that defines the access permissions for the newly created file system object.

If the parent directory does not have a default ACL, the permission bits are set depending on the setting of umask.

If a default ACL exists for the parent directory, the permission bits assigned to the new object correspond to the overlapping portion of the permissions of the mode parameter and those that are defined in the default ACL. The umask command is disregarded in this case.

The following three examples show the main operations for directories and default ACLs:

- Add a default ACL to the existing directory mydir with the following command:

setfacl -d -m group:jungle:r-x mydir

The option -d of the setfacl command prompts setfacl to perform the following modifications (option -m) in the default ACL.

Take a closer look at the result of this command:

```
geeko@dal0:~> setfacl -d -m group:jungle:r-x mydir
geeko@dal0:~> getfacl mydir
# file: mydir
# owner: geeko
# group: project3
user::rwx
user:jane:rwx
group::r-x
group:jungle:rwx
mask::rwx
other:---
default:user::rwx
default:group::r-x
default:group:jungle:r-x
default:mask::r-x
default:other:---
```

getfacl returns both the access ACL and the default ACL. The default ACL is formed by all lines that start with default.

Although you merely executed the setfacl command with an entry for the jungle group for the default ACL, setfacl automatically copied all other entries from the access ACL to create a valid default ACL.

Default ACLs do not have an immediate effect on access permissions. They only come into play when file system objects are created. These new objects inherit permissions only from the default ACL of their parent directory.

- In the following example, `mkdir` is used to create a subdirectory in `mydir`, which inherits the default ACL:

```
geeko@dal0:~> mkdir mydir/mysubdir
geeko@dal0:~> getfacl mydir/mysubdir
# file: mydir/mysubdir
# owner: geeko
# group: project3
user::rwx
group::r-x
group:jungle:r-x
mask::r-x
other:---
default:user::rwx
default:group::r-x
default:group:jungle:r-x
default:mask::r-x
default:other:---
```

As expected, the newly-created subdirectory `mysubdir` has permissions from the default ACL of the parent directory.

The access ACL of `mysubdir` is an exact reflection of the default ACL of `mydir`, as is the default ACL that this directory hands down to its subordinate objects.

- In the following example, touch is used to create a file in the mydir directory:

```
geeko@dal0:~> touch mydir/myfile
geeko@dal0:~> ls -l mydir/myfile
-rw-r-----+ ... geeko project3 ... mydir/myfile
geeko@dal0:~> getfacl mydir/myfile
# file: mydir/myfile
# owner: geeko
# group: project3
user::rw
group::r-x          # effective:r--
group:jungle:r-x    # effective:r--
mask::r--
other::---
```

touch passes a mode with the value 0666, which means that new files are created with read and write permissions for all user classes, provided no other restrictions exist in umask or in the default ACL.

In effect, this means that all access permissions not contained in the mode value are removed from the respective ACL entries. Although no permissions were removed from the ACL entry of the group class, the mask entry was modified to mask permissions not set using mode.

This approach ensures the smooth interaction of applications, such as compilers, with ACLs. You can create files with restricted access permissions and subsequently assign them as executable. The mask mechanism guarantees that the right users and groups can execute them as desired.

Additional setfacl Options

Named user entries are deleted using the option **-x**:

```
geeko@dal0:~> setfacl -x g:jungle mydir/
```

The option **-b** is used to remove all ACL entries.

ACLs can be saved to a file and restored from a file. Simply use `getfacl` to write the ACLs to a file and `setfacl` with the option **-M** to restore them to a file, as in the following example:

```
geeko@dal0:~> touch fileA fileB
geeko@dal0:~> setfacl -m u:tux:rw fileA
geeko@dal0:~> getfacl fileA > ACL-backup
geeko@dal0:~> setfacl -M ACL-backup fileB
geeko@dal0:~> getfacl fileB
# file: fileB
# owner: geeko
# group: project3
user::rw-
user:tux:rw-
group::r--
mask::rw-
other::r--
```

The ACL Check Algorithm

A check algorithm is applied before any process or application is granted access to an ACL-protected file system object.

As a basic rule, the ACL entries are examined in the following sequence: owner, named user, owning group or named group, and other. The access is handled in accordance with the entry that best suits the process. Permissions do not accumulate.

Things are more complicated if a process belongs to more than one group and belongs to several group entries. An entry is randomly selected from the suitable entries with the required permissions.

It is irrelevant which of the entries triggers the final result, which is ***access granted***. Likewise, if none of the suitable group entries contains the correct permissions, a randomly selected entry triggers the final result, which is ***access denied***.

How Applications Handle ACLs

As described in the preceding sections, you can use ACLs to implement very complex permission scenarios that meet the requirements of applications. However, some important applications still lack ACL support. Except for the star archiver, there are currently no backup applications included with SLES 10 that guarantee the full preservation of ACLs.

The basic file commands (cp, mv, ls, and so on) support ACLs, but many editors and file managers (such as Konqueror or Nautilus) do not.

For example, when you copy files with Konqueror or Nautilus, the ACLs of these files are lost. When you modify files with an editor, the ACLs of files are sometimes preserved, sometimes not, depending on how the editor handles files.

If the editor writes the changes to the original file, the access ACL is preserved. If the editor saves the updated contents to a new file that is subsequently renamed to the old filename, the ACLs might be lost, unless the editor supports ACLs.

Exercise 3-3 Use ACLs

In this exercise, you practice using ACLs.

You will find this exercise in the workbook.

(End of Exercise)

Summary

Objective	Summary
1. Configure User Authentication with PAM	<p>Linux uses PAM (Pluggable Authentication Modules) in the authentication process as a layer that communicates between users and applications.</p> <p>Within the PAM framework there are four different module types: auth, account, session, and password. Control flags—required, requisite, sufficient, optional—govern what happens on success or failure of a module.</p> <p>Files in /etc/pam.d/ are used to configure PAM, with additional configuration options in files in /etc/security/ for certain modules.</p>
2. Manage and Secure the Linux User Environment	<p>You should only use the root account when absolutely necessary, using tools like sudo, su, or gnomesu as applicable.</p> <p>Defaults for user accounts and other security relevant settings can be configured using the YaST Local Security module.</p> <p>The configuration settings are written to various files, the most pertinent being files in /etc/default/, and /etc/login.defs.</p>

Objective	Summary
3. Use Access Control Lists (ACLs) for Advanced Access Control	<p>ACLs extend the classic Linux file system permissions.</p> <p>They let you assign permissions to named users and named groups.</p> <p>ACLs also provide a mask entry, which basically limits the permissions of named users and named groups.</p> <p>The ACL entries are managed with getfacl and setfacl.</p> <p>Directories can have a default ACL that is inherited by newly created files or subdirectories.</p>

CNI USE ONLY-1 HARDCOPY PERMITTED

SECTION 4 Configure the Network Manually

Although almost every step of a network configuration is done for you when you use YaST, it's sometimes useful to configure the network settings manually. For testing and troubleshooting, it can be much faster to change the network setup from the command line.

In this section, you learn how to configure network devices manually. You also learn how to configure routing with command line tools and how to save the network setup to configuration files.

Objectives

1. Understand Linux Network Terms
2. Set Up Network Interfaces with the ip Tool
3. Set Up Routing with the ip Tool
4. Test the Network Connection With Command Line Tools
5. Configure Host Name and Name Resolution
6. Use the NetworkManager to Configure the Network

Objective 1 Understand Linux Network Terms

Before you can configure the network manually with `ip`, you need to understand the following Linux networking terms:

- **Device.** The network adapter built into the system.
- **Interface.** To use a physical device, a software component creates an interface to the device. This interface can be used by other software applications.

The software component which creates the interface is also called a *driver*.

In Linux, network interfaces use a standard naming scheme. Interfaces to Ethernet adapters follow the naming scheme `eth0`, `eth1`, `eth2`, and so on. For every adapter installed in the system, an interface is created when the appropriate driver is loaded.

The command line tools for the network configuration use the term *device* when they actually mean an interface. The term *device* is used in this section for both physical devices and software interfaces.

- **Link.** The command line tool `ip` uses the term *link* to refer to the connection of a device to the network.
- **Address.** The IP address assigned to a device. The address can be either an IPv4 or an IPv6 address. To use a device in a network, you have to assign at least one address to it. However, you can assign more than one address to a device.
- **Broadcast.** The broadcast address of a network. By sending a network packet to the broadcast address, you can reach all hosts in the locally connected network at the same time. When you assign an IP address to a device, you can also set this broadcast address.
- **Route.** The path an IP packet takes from the source to the destination host. The term *route* also refers to an entry in the routing table of the Linux kernel.

CNI USE ONLY-1 HARDCOPY PERMITTED

Objective 2 Set Up Network Interfaces with the ip Tool

You normally configure a network card with YaST during or after installation. You can use the tool **ip** to change the network interface configuration quickly from the command line.

Changing the network interface configuration at the command line is especially useful for testing purposes; but if you want a configuration to be permanent, you must save it in a configuration file. These configuration files are generated automatically when you set up a network card with YaST.

You can use **ip** to perform the following tasks:

- Display the Current Network Configuration
- Change the Current Network Configuration



You can enter **/sbin/ip** as a normal user to display the current network setup only. To change the network setup, you have to be logged in as root.

As changes made with **ip** are lost with the next reboot, you also have to know how to:

- Save Device Settings to a Configuration File

Display the Current Network Configuration

With the **ip** tool, you can display the following information:

- IP Address Setup
- Device Attributes
- Device Statistics

IP Address Setup

To display the IP address setup of all interfaces, enter **ip address show**. Depending on your network setup, you see information similar to the following:

```
da2:~ # ip address show
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 127.255.255.255 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,NOTRAILERS,UP> mtu 1500 qdisc
pfifo_fast qlen 1000
    link/ether 00:30:05:4b:98:85 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.2/24 brd 10.0.0.255 scope global eth0
    inet6 fe80::230:5ff:fe4b:9885/64 scope link
        valid_lft forever preferred_lft forever
3: sit0: <NOARP> mtu 1480 qdisc noqueue
    link/sit 0.0.0.0 brd 0.0.0.0
```

The information is grouped by network interfaces. Every interface entry starts with a digit, called the *interface index*, with the interface name displayed after the interface index.

In the above example, there are 3 interfaces:

- **lo**. The loopback device, which is available on every Linux system, even when no network adapter is installed. (As stated above, “device” and “interface” are often used synonymously in the context of network configuration.) Using this virtual device, applications on the same machine can use the network to communicate with each other.

For example, you can use the IP address of the loopback device to access a locally installed web server by typing **http://127.0.0.1** in the address bar of your web browser.

- **eth0**. The first Ethernet adapter of the computer in this example. Ethernet devices are normally called eth0, eth1, eth2, and so on.

- **sit0.** This is a special virtual device which can be used to encapsulate IPv4 into IPv6 packets. It's not used in a normal IPv4 network.

You always have the entries for the loopback and sit devices. Depending on your hardware setup, you might have more Ethernet devices in the ip output.

Several lines of information are displayed for every network interface, such as eth0 in the preceding example:

```
2: eth0: <BROADCAST,MULTICAST,NOTRAILERS,UP> mtu 1500 qdisc
pfifo_fast qlen 1000
```

The most important information of the line in this example is the interface index (**2**) and the interface name (**eth0**).

The other information shows additional attributes set for this device, such as the hardware address of the Ethernet adapter (00:30:05:4b:98:85):

```
link/ether 00:30:05:4b:98:85 brd ff:ff:ff:ff:ff:ff
```

In the following line, the IPv4 setup of the device is displayed:

```
inet 10.0.0.2/24 brd 10.0.0.255 scope global eth0
```

The IP address (**10.0.0.2**) follows inet, and the broadcast address (**10.0.0.255**) after brd. The length of the network mask is displayed after the IP address, separated by a /. The length is displayed in bits (**24**).

The following lines show the IPv6 configuration of the device:

```
inet6 fe80::230:5ff:fe4b:9885/64 scope link
      valid_lft forever preferred_lft forever
```

The address shown here is automatically assigned, even though IPv6 is not used in the network that is connected with the device. The address is generated from the hardware address of the device.

Depending on the device type, the information can differ. However, the most important information (such as assigned IP addresses) is always shown.

Device Attributes

If you are only interested in the device attributes and not in the IP address setup, you can enter **ip link show**:

```
da2:~ # ip link show
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,NOTRAILERS,UP> mtu 1500 qdisc
pfifo_fast qlen 1000
    link/ether 00:30:05:4b:98:85 brd ff:ff:ff:ff:ff:ff
3: sit0: <NOARP> mtu 1480 qdisc noqueue
    link/sit 0.0.0.0 brd 0.0.0.0
```

The information is similar to what you have seen when entering **ip address show**, but the information about the address setup is missing. The device attributes are displayed in brackets right after the device name.

The following is a list of possible attributes and their meanings:

- **UP.** The device is turned on. It is ready to accept packets for transmission and it's ready to receive packets from the network.
- **LOOPBACK.** The device is a loopback device.
- **BROADCAST.** The device can send packets to all hosts sharing the same network.
- **POINTOPOINT.** The device is only connected to one other device. All packets are sent to and received from the other device.

- **MULTICAST.** The device can send packets to a group of other systems at the same time.
- **PROMISC.** The device listens to all packets on the network, not only to those sent to the device's hardware address. This is usually used for network monitoring.

Device Statistics

You can use the option **-s** with the command **ip** to display additional statistics information about the devices. The command looks like the following:

ip -s link show eth0

By giving the device name at the end of the command line, the output is limited to one specific device. This can also be used to display the address setup or the device attributes.

The following is an example of the information displayed for the device **eth0**:

```
da2:~ # ip -s link show eth0
2: eth0: <BROADCAST,MULTICAST,NOTRAILERS,UP> mtu 1500 qdisc
pfifo_fast qlen 1000
    link/ether 00:30:05:4b:98:85 brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped  overrun  mcast
    849172787  9304150  0       0        0        0
    TX: bytes  packets  errors  dropped  carrier  collsns
    875278145  1125639  0       0        0        0
```

Two additional sections with information are displayed for every device. Each of the sections has a headline with a description of the displayed information.

The section starting with **RX** displays information about received packets, and the section starting with **TX** displays information about sent packets.

The sections display the following information:

- **Bytes.** The total number of bytes received or transmitted by the device.
- **Packets.** The total number of packets received or transmitted by the device.
- **Errors.** The total number of receiver or transmitter errors.
- **Dropped.** The total number of packets dropped due to a lack of resources.
- **Overrun.** The total number of receiver overruns resulting in dropped packets.

As a rule, if a device is overrun, it means that there are serious problems in the Linux kernel or that your computer is too slow for the device.

- **Mcast.** The total number of received multicast packets. This option is supported by only a few devices.
- **Carrier.** The total number of link media failures, because of a lost carrier.
- **Collsns.** The total number of collision events on Ethernet-like media.
- **Compressed.** The total number of compressed packets.

Change the Current Network Configuration

You can also use the `ip` tool to change the network configuration by performing the following tasks:

- Assign an IP Address to a Device
- Delete the IP Address from a Device
- Change Device Attributes

Assign an IP Address to a Device

To assign an address to a device, use a command similar to the following:

```
da2:~ # ip address add 10.0.0.2/24 brd + dev eth0
```

In this example, the command assigns the IP address **10.0.0.2** to the device **eth0**. The network mask is **24** bits long, as determined by the **/24** after the IP address. The **brd +** option sets the broadcast address automatically as determined by the network mask.

You can enter **ip address show dev eth0** to verify the assigned IP address. The assigned IP address is displayed in the output of the command line.

You can assign more than one IP address to a device.

Delete the IP Address from a Device

To delete the IP address from a device, use a command similar to the following:

```
da2:~ # ip address del 10.0.0.2 dev eth0
```

In this example, the command deletes the IP address **10.0.0.2** from the device **eth0**.

Use **ip address show eth0** to verify that the address was deleted.

Change Device Attributes

You can also change device attributes with the ip tool. The following is the basic command to set device attributes:

ip link set *device attribute*

The possible attributes are described in “Device Attributes” on 4-6. The most important attributes are *up* and *down*. By setting these attributes, you can enable or disable a network device.

To enable a network device (such as eth0), enter the following command:

```
da2:~ # ip link set eth0 up
```

To disable a network device (such as eth0), enter the following command:

```
da2:~ # ip link set eth0 down
```

Save Device Settings to a Configuration File

All device configuration changes you make with ip are lost when the system is rebooted. To restore the device configuration automatically when the system is started, the settings need to be saved in configuration files.

The configuration files for network devices are located in the directory **/etc/sysconfig/network/**.

If the network devices are set up with YaST, one configuration file is created for every device.

For Ethernet devices, the filenames consist of ifcfg-eth-id- and the hardware address of the device. For a device with the hardware address 00:30:05:4b:98:85, the filename would be ifcfg-eth-id-00:30:05:4b:98:85.

We recommend that you set up a device with YaST first and make changes in the configuration file. Setting up a device from scratch is a complex task, because the hardware driver also needs to be configured manually.

If you have more than one network adapter in your system, it might be difficult to find the corresponding configuration file for a device.

You can use the command **ip link show** to display the hardware address for each Ethernet device. Because the hardware address is part of the file name, you can identify the right configuration file.

The content of the configuration files depends on the configuration of the device. To change the configuration file, you need to know how to do the following:

- Configure a Device Statically
- Configure a Device Dynamically with DHCP
- Start and Stop Configured Interfaces

Configure a Device Statically

The content of a configuration file of a statically configured device is similar to the following:

```
BOOTPROTO='static'
BROADCAST=''
ETHTOOL_OPTIONS=''
IPADDR='10.0.0.2'
MTU=''
NAME='Digital DECchip 21142/43'
NETMASK='255.255.255.0'
NETWORK=''
REMOTE_IPADDR=''
STARTMODE='auto'
UNIQUE='rBUF.+xOL8ZCSAQC'
USERCONTROL='no'
_nm_name='bus-pci-0000:00:0b.0'
ETHTOOL_OPTIONS=''
```

CNI USE ONLY-1 HARDCOPY PERMITTED

The configuration file includes several lines. Each line has an option and a value assigned to that option, as explained below:

■ **BOOTPROTO='static'**

The option **BOOTPROTO** determines the way the device is configured. There are 2 possible values:

- **Static.** The device is configured with a static IP address.
- **DHCP.** The device is configured automatically with a DHCP server.

■ **REMOTE_IPADDR=''**

You need to set the value for the **REMOTE_IPADDR** option only if you are setting up a point-to-point connection.

■ **STARTMODE='onboot'**

The **STARTMODE** option determines how the device is started. The option can include the following values:

- **auto.** The device is started at boot time or when initialized at runtime.
- **manual.** The device must be started manually with **ifup**.
- **ifplugd.** The interface is controlled by **ifplugd**. If you want to use interfaces mutually exclusive, also set **IFPLUGD_PRIORITY**

■ **UNIQUE='rBUF.+xOL8ZCSAQC'**
_nm_name='bus-pci-0000:00:0b.0'

These 2 lines contain options added by YaST when the device is configured. They don't affect the network configuration itself.

■ **BROADCAST=''**
IPADDR='10.0.0.2'
NETMASK='255.255.255.0'
NETWORK=''

These 4 lines contain the options for the network address configuration. The options have the following meanings:

- ❑ **BROADCAST.** The broadcast address of the network. If empty, the broadcast address is derived from the IP address and the netmask, according to the configuration in **/etc/sysconfig/network/config**.
- ❑ **IPADDR.** The IP address of the device.
- ❑ **NETMASK.** The network mask.
- ❑ **NETWORK.** The address of the network itself.
- **MTU=**"

You can use the MTU option to specify a value for the MTU (Maximum Transmission Unit). If you don't specify a value, the default value is used. For an Ethernet device, the default value is 1500 bytes.

- **ETHTOOL_OPTIONS=**"

ethtool is used for querying settings of an Ethernet device and changing them, for instance setting the speed or half/full duplex mode. The manual page for ethtool lists the available options.

If you want ethtool to modify any settings, list the options here; if no options are listed, ethtool is not called.

The file **/etc/sysconfig/network/ifcfg.template** contains a template that you can use as a base for device configuration files. It also has comments explaining the various options.

Configure a Device Dynamically with DHCP

If you want to configure a device by using a DHCP server, you set the BOOTPROTO option to **dhcp** as shown in the following:

BOOTPROTO='dhcp'

When the device is configured by using DHCP, you don't need to set any options for the network address configuration in the file. If there are any settings, they are overwritten by the settings of the DHCP server.

Start and Stop Configured Interfaces

To apply changes to a configuration file, you need to stop and restart the corresponding interface. You can do this with the commands `ifdown` and `ifup`.

For example, entering **`ifdown eth0`** disables the device `eth0`. **`ifup eth0`** enables `eth0` again.

When the device is restarted, the new configuration is read from the configuration file.



Configuring the interfaces with IP addresses, routes, etc. with the `ip` tool requires an existing device setup, including a correctly loaded kernel module. This is usually done at boot time by `/sbin/hwup`, using the configuration contained in files in the directory `/etc/sysconfig/hardware/`. Information is available in the manual page for `hwup`.



Under certain circumstances physical network devices can change the interface name, for instance the interface that used to be called `eth0` now becomes `eth1` and vice versa. Sometimes this happens from one boot to the next, even without any physical changes on the hardware. Information on how to achieve persistent interface names is contained in the file `/usr/share/doc/packages/sysconfig/README.Persistent_Interface_Names`.

Objective 3 Set Up Routing with the ip Tool

You can use the `ip` tool to configure the routing table of the Linux kernel. The routing table determines the path IP packets use to reach the destination system.



Because routing is a very complex topic, this objective only covers the most common routing scenarios.

You can use the `ip` tool to perform the following tasks:

- View the Routing Table
- Add Routes to the Routing Table
- Delete Routes from the Routing Table

As changes made with `ip` are lost with the next reboot, you also have to know how to:

- Save Routing Settings to a Configuration File

View the Routing Table

To view the current routing table, enter **`ip route show`**. For most systems, the output looks similar to the following:

```
da2:~ # ip route show
10.0.0.0/24 dev eth0 proto kernel scope link src 10.0.0.2
169.254.0.0/16 dev eth0 scope link
127.0.0.0/8 dev lo scope link
default via 10.0.0.254 dev eth0
```

Every line represents an entry in the routing table. Each line in the example is shown and explained below:

- **`10.0.0.0/24 dev eth0 proto kernel scope link src 10.0.0.2`**

This line represents the route for the local network. All network packets to a system in the same network are sent directly through the device **eth0**.

- **169.254.0.0/16 dev eth0 scope link**

This line shows a network route for the **169.254.0.0** network. Hosts can use this network for address auto configuration.

SLES 10 automatically assigns a free IP address from this network when no other device configuration is present. The route to this network is always set, especially when the system itself has no assigned IP address from that network

- **127.0.0.0/8 dev lo scope link**

This is the route for the loopback device.

- **default via 10.0.0.254 dev eth0**

This line is the entry for the **default** route. All network packets that cannot be sent according to the previous entries of the routing table are sent through the gateway defined in this entry.

Depending on the setup of your machine, the content of the routing table varies. In most cases, you have at least 2 entries in the routing table:

- One route to the local network the system is connected to
- One route to the default gateway for all other packets

Add Routes to the Routing Table

The following are the most common tasks you do when adding a route:

- Set a Route to the Locally Connected Network
- Set a Route to a Different Network
- Set a Default Route



Remember to substitute your own network and gateway addresses when using the following examples in a production environment.

Set a Route to the Locally Connected Network

The following command sets a route to the locally connected network:

```
da2:~ # ip route add 10.0.0.0/24 dev eth0
```

This system in this example is in the **10.0.0.0** network. The network mask is **24** bits long (255.255.255.0). All packets to the local network are sent directly through the device **eth0**.

Set a Route to a Different Network

The following command sets a route to different network:

```
da2:~ # ip route add 192.168.1.0/24 via 10.0.0.100
```

All packets for the network **192.168.1.0** are sent through the gateway **10.0.0.100**.

Set a Default Route

The following command sets a default route:

```
da2:~ # ip route add default via 10.0.0.254
```

Packets that cannot be sent according to previous entries in the routing table are sent through the gateway **10.0.0.254**.

Delete Routes from the Routing Table

To delete an entry from the routing table, use a command similar to the following:

```
da2:~ # ip route delete 192.168.1.0/24 dev eth0
```

This command deletes the route to the network **192.168.1.0** assigned to the device **eth0**.

Save Routing Settings to a Configuration File

Routing settings made with the `ip` tool are lost when you reboot your system. Settings have to be written to configuration files to be restored at boot time.

Routes to the directly connected network are automatically set up when a device is started. All other routes are saved in the configuration file `/etc/sysconfig/network/routes`.

The following shows the content of a typical configuration file:

```
192.168.1.0 10.0.0.100 255.255.255.0 eth-id-00:30:05:4b:98:85
default 10.0.0.254 - -
```

Each line of the configuration file represents an entry in the routing table. Each line is shown and explained below:

- **192.168.1.0 10.0.0.100 255.255.255.0
eth-id-00:30:05:4b:98:85**

All packets sent to the network **192.168.1.0** with the network mask **255.255.255.0** are sent to the gateway **10.0.0.100** through the device with the id **eth-id-00:30:05:4b:98:85**. The id is the same as used for the device configuration file.

- **Default 10.0.0.254 - -**

This entry represents a default route. All packets that are not affected by the previous entries of the routing table are sent to the gateway **10.0.0.254**. It's not necessary to fill out the last 2 columns of the line for a default route.

To apply changes to the routing configuration file, you need to restart the affected network device with the commands **ifdown** and **ifup**.

Objective 4 Test the Network Connection With Command Line Tools

After the network is configured, you might want to test the network connection by doing the following:

- Test Network Connections with ping
- Trace Network Packets with traceroute

Test Network Connections with ping

The tool ping lets you check network connections in a simple way between two hosts. If the ping command works, then both the physical and logical connections are correctly set up between the two hosts.

The ping command sends special network packets to the target system and waits for a reply. In the simplest scenario, you enter ping with an IP address:

ping 10.0.0.10

You can also use the host name of the target system instead of an IP address. The output of ping looks similar to the following:

```
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data.  
64 bytes from 10.0.0.10: icmp_seq=1 ttl=60 time=2.95 ms  
64 bytes from 10.0.0.10: icmp_seq=2 ttl=60 time=2.16 ms  
64 bytes from 10.0.0.10: icmp_seq=3 ttl=60 time=2.18 ms  
64 bytes from 10.0.0.10: icmp_seq=4 ttl=60 time=2.08 ms
```

Each line of the output represents a packet sent by ping. Ping keeps sending packets until it's terminated by pressing **Ctrl+C**.

The output displays the following information:

- The size of an ICMP datagram (64 bytes).

- The IP address of the target system (from 10.0.0.10).
- The sequence number of each datagram (seq=1).
- The TTL (TTL, time to live) of the datagram (ttl=60).
- The amount of time that passes between the transmission of a packet and the time a corresponding answer is received (time=2.95 ms). This time is also called the Round Trip Time.

If you get an answer from the target system, you can be sure that the basic network device setup and routing to the target host works.

The following table provides some options for ping you can use for advanced troubleshooting:

Table 4-1

Option	Description
<i>-c count</i>	The number of packets to be sent. After this number has been reached, ping is terminated.
<i>-I interface</i>	Specifies the network interface to be used on a computer with several network interfaces.
<i>-i seconds</i>	Specifies the number of seconds to wait between individual packet shipments. The default setting is 1 second.
<i>-f</i>	(Flood ping) Packets are sent one after another at the same rate as the respective replies arrive. Only root can use this option. For normal users the minimum time is 200 milliseconds.
<i>-l preload</i>	(Lowercase L) sends packets without waiting for a reply.
<i>-n</i>	The numerical output of the IP address. Address resolutions to host names are not carried out.
<i>-t ttl</i>	Sets the Time To Live for packets to be sent.

Table 4-1 *(continued)*

Option	Description
<code>-w <i>maxwait</i></code>	Specifies a timeout in seconds, before ping exits regardless of how many packets have been sent or received.
<code>-b</code>	Sends packets to the broadcast address of the network.

Trace Network Packets with traceroute

The diagnosis tool `traceroute` is primarily used to check the routing between different networks. To achieve this task, `traceroute` sends packets with an increasing TTL value to the destination host, whereby three packets of each value are sent.

Traceroute also uses UDP packets, which are called *datagrams*.

First, three datagrams with a TTL=1 are sent to the host, then three packets with a TTL=2, and so on. The TTL of a datagram is reduced by one, every time it passes through a router.

When the TTL reaches zero, the datagram is discarded and a message is sent to the sender. Because the TTL is increased by one every three packets, `traceroute` can collect information about every router on the way to the destination host.

You normally include a host name with the `traceroute` command, as in the following:

`traceroute pluto.example.com`

It's also possible to use an IP address instead of the host name. The output of traceroute looks similar to the following:

```
traceroute to pluto.example.com (192.168.2.1), 30 hops max,
40 byte packets
1 da1.digitalairlines.com (10.0.0.254) 0 ms 0 ms 0 ms
2 antares.example.com (192.168.1.254) 14 ms 18 ms 14 ms
3 pluto.example.com (192.168.2.1) 19 ms * 26 ms
```

The first line of the output displays general information about the traceroute call. Each of the lines that follow represents a router on the way to the destination host. Every router is displayed with the host name and IP address.

Traceroute also displays information about the round trip times of the 3 datagrams returned by every router. An asterisk indicates that no response was received from the router. The last line of the output represents the destination host itself.

Exercise 4-1 Configure the Network Connection Manually

In this exercise, you learn how to configure the network manually.

You will find this exercise in the workbook.

(End of Exercise)

Objective 5 **Configure Host Name and Name Resolution**

The host name and the name resolution can also be set up manually. In this objective, you learn how to do the following:

- Set the Host and Domain Name
- Configure Name Resolution

Set the Host and Domain Name

The host name is configured in the file `/etc/HOSTNAME`.

The content of the file is similar to the following:

```
da2.digitalairlines.com
```

The file contains the fully qualified domain name of the system, in this case, **da2.digitalairlines.com**.

Configure Name Resolution

The name resolution is configured in the file `/etc/resolv.conf`.

The content of the file is similar to the following:

```
search digitalairlines.com
nameserver 10.0.0.254
nameserver 10.10.0.1
nameserver 10.0.10.1
```

The file contains 2 types of entries:

- **search.** The domain name in this option is used to complete incomplete host names. For example, if you look up the host name da3, the name is automatically completed to the fully qualified domain name da3.digitalairlines.com.
- **nameserver.** Every entry starting with nameserver is followed by an IP address of a name server. You can configure up to 3 name servers. If the first name server fails, the next one is used.

Objective 6 **Use the NetworkManager to Configure the Network**

In case you are using SUSE Linux Enterprise Server 10 on a laptop, you will most likely use different kinds of Internet access, depending on where you are—maybe a LAN in your office and a wireless connection at a customer site.

The conventional network setup requires you to switch to the root account to change the network configuration. The purpose of the **NetworkManager** (package `NetworkManager`) is to allow the user to change the network configuration according to his needs, without switching to the root account.

NetworkManager runs as a root-user system level daemon, since root privileges are needed to manipulate hardware directly. The programs used for this purpose are `/usr/sbin/NetworkManager` and `/usr/sbin/NetworkManagerDispatcher`. **nm-tools** can be used to list information about NetworkManager, devices, and wireless networks.

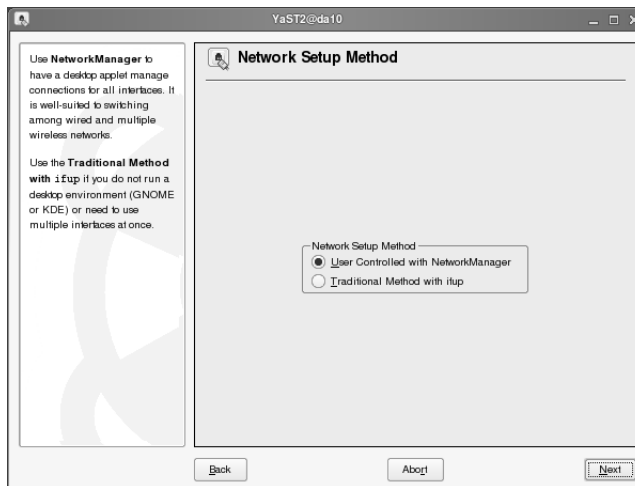
From a list of all adapters currently installed on the system, NetworkManager will first try a wired and then a wireless adapter. Wireless adapters that support wireless scanning are preferred over ones that cannot. NetworkManager does not try to keep a connection up as long as possible, meaning that plugging into a wired network will switch the connection to the wired network, away from the wireless one.

For wireless networking support, NetworkManager keeps two lists of wireless networks: a ***Trusted list***, and a ***Preferred list***. The trusted list contains networks the user specifically adds to it, while the preferred list contains networks the user forces NetworkManager to connect to.

Since trusted and preferred networks are user-specific, there must be some mechanism of getting and storing this information per user. This is achieved with a desktop-level per-user process, **nm-applet**, or KNetworkManager in KDE. NetworkManager communicates over DBUS with these user level processes.

Switching to NetworkManager is done by starting YaST and selecting **Network Devices > Network Cards**. In the **Network Setup Method** dialog, you select **User Controlled with NetworkManager**:

Figure 4-1

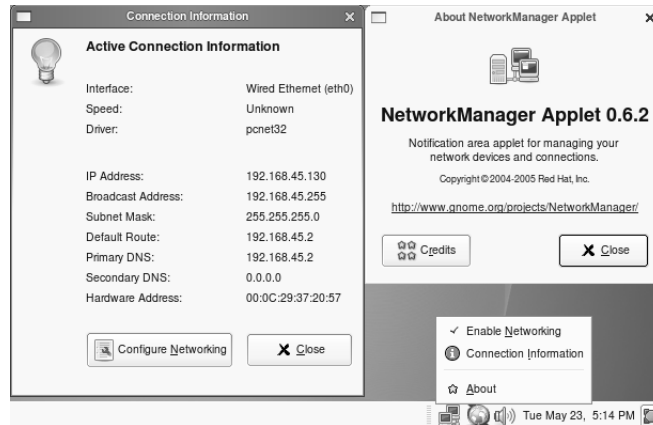


The following dialogs of this module are the same for both setup methods.

When selecting **User Controlled with NetworkManager**, YaST sets the variable **NETWORKMANAGER=** in `/etc/sysconfig/network/config` to “yes”.

Choosing the NetworkManager in YaST will also automatically start the Network Applet when a user logs in. Using the desktop applet, the user can easily change the network configuration:

Figure 4-2



Note: As there was no wireless card built into the computer on which the above screenshot was taken, there is no option for switching networks in this screenshot.

Summary

Objective	Summary
1. Understand Linux Network Terms	<p>The following terms are used for the Linux network configuration:</p> <ul style="list-style-type: none">■ Device■ Interface■ Link■ Address■ Broadcast■ Route
2. Set Up Network Interfaces with the ip Tool	<p>You can perform the following tasks with the ip tool:</p> <ul style="list-style-type: none">■ Display the IP address setup: ip address show■ Display device attributes: ip link show■ Display device statistics: ip -s link show■ Assign an IP address: ip address add <i>IP_address/netmask</i> brd + dev <i>device_name</i>■ Delete an IP address: ip address del <i>IP_address</i> dev <i>device_name</i> <p>The configuration files for network devices are located in <code>/etc/sysconfig/network</code>.</p> <p>Configured devices can be enabled with ifup <i>device_name</i> and disabled with ifdown <i>device_name</i>.</p>

Objective	Summary
3. Set Up Routing with the ip Tool	<p>You can perform the following tasks with the ip tool:</p> <ul style="list-style-type: none">■ View the routing table: ip route show■ Add routes to the routing table ip route add <i>network/netmask</i> dev <i>device_name</i>■ Delete routes from the routing table ip route del <i>network/netmask</i> dev <i>device_name</i> <p>The configuration for the routing table is located in the file <code>/etc/sysconfig/network/routes</code>.</p>
4. Test the Network Connection With Command Line Tools	<p>Two frequently used command line tools are available to test the network connection:</p> <ul style="list-style-type: none">■ ping ping <i>hostname</i> <p>With ping you can test whether another host is reachable in the network.</p>■ traceroute traceroute <i>hostname</i> <p>With traceroute you can test the routing in the network.</p>

Objective	Summary
5. Configure Host Name and Name Resolution	<p>The host name is configured in the file <code>/etc/HOSTNAME</code>.</p> <p>The name resolution is configured in the file <code>/etc/resolv.conf</code>.</p> <p>One line specifies the search domain; the others list up to three available name servers.</p>
6. Use the NetworkManager to Configure the Network	<p>NetworkManager allows the user to change the network configuration without having to assume root privileges.</p> <p>NetworkManager is mainly useful for use on laptops.</p>

SECTION 5 Administer Linux Processes and Services

In this section you learn how to view and manage processes, and how to schedule jobs.

Objectives

1. View and Manage Processes
2. Schedule Jobs

Objective 1 View and Manage Processes

To manage processes on your SUSE Linux Enterprise Server, you need to know the following:

- Understand Process Definitions
- Learn Jobs and Processes
- Manage Foreground and Background Processes
- View and Prioritize Processes
- End a Process
- Understand Services (Daemons)
- Manage a Daemon Process

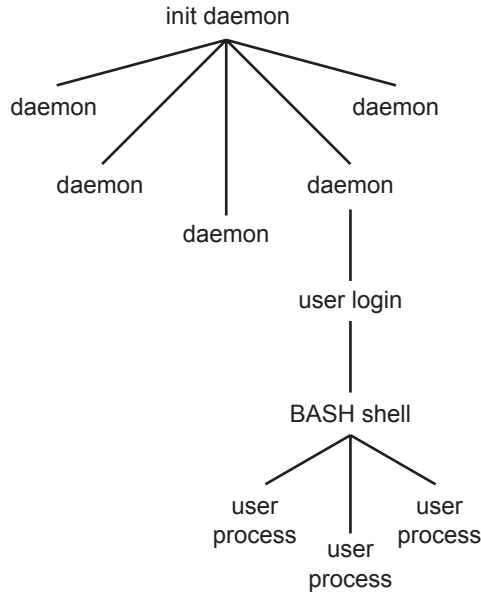
Understand Process Definitions

The following terms are used to describe Linux processes:

- **Program.** A structured set of commands stored in an executable file on a Linux file system. A program can be executed to create a process.
- **Process.** A program that is loaded into memory and executed by the CPU.
- **User Process.** A process launched by a user that is started from a terminal or within the graphical environment.
- **Daemon Process.** A system process that is not associated with a terminal or a graphical environment. It is a process or collection of processes that wait for an event to trigger an action on the part of the program. In network-based services, this event is a network connection. Other services, like cron and atd, are time-based and perform certain tasks at certain points in time.

The following illustrates the relationship between daemon processes and user processes:

Figure 5-1



In this example, during the boot process of a Linux system, the `init` process launches several daemons (*daemon processes*), including a daemon for user login.

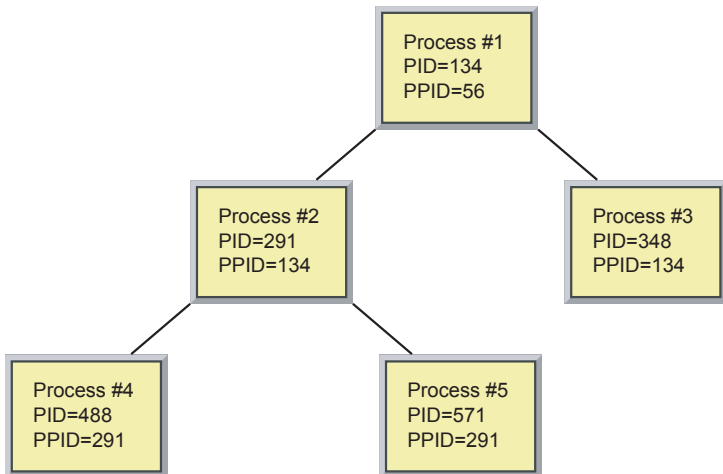
After the user logs in on a text console, a shell is started that lets him start processes manually (*user processes*). Within a graphical environment he can open a terminal window from which he can start user processes, or he starts processes by clicking on icons or choosing from menus.

- **Process ID (PID).** A unique identifier assigned to every process as it begins.
- **Child Process.** A process that is started by another process (the parent process).

- **Parent Process.** A process that starts other processes (child processes).
- **Parent Process ID (PPID).** The PID of the parent process that created the current process.

The following illustrates the relationship between parent and child process ID numbers:

Figure 5-2



For example, Process #1 is assigned a PID of 134. This process launches Process #2 with a PID of 291 and Process #3 with a PID of 348.

Because Process #1 launched Process #2 and Process #3, the second and third processes are considered *child processes* of Process #1 (the *parent process*). The PPID of Processes #2 and #3 is the PID of process #1—134.

Learn Jobs and Processes

In Linux, you use a *job identifier* (commonly called a *job ID*) to refer to processes when launching processes at the command line. The job identifier is a shell-specific numeric value that identifies the running program uniquely within that shell.

Independent of the shell each process is identified using a *process ID* (commonly called a *PID*) that is unique across the entire system. All jobs have a PID, but not all processes have a usable job identifier.

PID 1 always belongs to the init process. This is the first process started on the system and it creates a number of other processes, which in turn can generate additional processes.

If the highest possible PID within that system has been reached, the next process is allocated the lowest available number (such as PID 17494). Processes run for different lengths of time. After one process has ended, its number again becomes available.

When performing tasks such as changing the priority level of a running program, you use the PID instead of the job ID.

When you want to switch a process from the background to the foreground (and the process was started from a terminal), you use the job ID.

Manage Foreground and Background Processes

The Linux shell environment allows processes to run in either the *foreground* or the *background*.

Processes executed in the foreground are started in a terminal window and run until the process completes; the terminal window does not return to a prompt until the program's execution is complete.

Background process execution occurs when a process is started and the terminal window returns to a prompt before the process finishes executing.

Existing processes can be switched from foreground to background execution under the following circumstances:

- The process must be started in a terminal window or console shell.
- The process does not require input from the terminal window.

If the process meets this criteria, it can be moved to the background.



Processes that require input within the terminal can be moved to the background as well, but when input is requested, the process will be suspended until it is brought to the foreground and the requested input is provided.

Commands in a shell can be started in the foreground or in the background. Processes in the foreground can directly receive transmitted signals.

For example, if you enter **xeyes** to start the XEYES program, it is running in the foreground. If you press **Ctrl+Z**, the process stops:

```
[1]+  Stopped                  xeyes
geeko@da10:~>
```

You can continue running a stopped process in the background by entering **bg**, as in the following:

```
geeko@da10:~> bg
[1]+  xeyes &
geeko@da10:~>
```

The ampersand (**&**) displayed in the output means that the process is now running in the background.

Appending an ampersand to a command starts the process in the background (instead of the foreground), as in the following:

```
geeko@da10:~> xeyes &
[2] 4351
geeko@da10:~>
```

With this the shell from which you started the program is available again for user input immediately.

In the above example, both the job ID ([2]) and the process ID of the program (**4351**) are returned.

Each process started from the shell is assigned a *job ID* by the job control of the shell. The command **jobs** lists the contents of job control, as in the following:

```
geeko@da10:~> jobs
[1]+  Stopped                  xeyes
[2]   Running                  xeyes &
[4]-  Running                  sleep 99 &
geeko@da10:~>
```

In this example, the process with job ID 3 is already terminated. The processes 2 and 4 are running in the background (notice the ampersand), and process 1 is stopped.

The + sign indicates the process that will respond to **fg** without options, and the - sign indicates the process that inherits the + sign once the process with the + sign ends.

The next background process will be assigned the job ID of 5 (highest number + 1).

Not only can you continue running a stopped process in the background by using the command `bg`, you can also switch a process to the foreground by entering **`fg job_ID`**, as in the following:

```
geeko@dal10:~> fg 1
xeyes
```

The shell also informs you about the termination of a process running in the background:

```
[4]-  Done                  sleep 99
```

The job ID is displayed in square brackets. **Done** means the process terminated properly. If you see **Terminated** instead, it means that the process was requested to terminate. **Killed** indicates a forceful termination of the process.

View and Prioritize Processes

In addition to running jobs in the foreground or the background, you can view information about the processes and assign priorities by using the following tools:

- `ps`
- `pstree`
- `nice` and `renice`
- `top`

ps

You can view running processes with the command **ps** (process status). :

```

geeko@da10:~> ps
  PID TTY          TIME CMD
 3103 pts/0        00:00:00 bash
 3129 pts/0        00:00:00 sleep
 3130 pts/0        00:00:00 ps
geeko@da10:~>

```

With the option **x**, you can also view terminal-independent processes, as in the following:

```

geeko@da10:~> ps x
  PID TTY          STAT TIME COMMAND
 3102 ?            S      0:00 sshd: geeko@pts/0
 3103 pts/0        Ss     0:00 -bash
 3129 pts/0        S      0:00 sleep 99
 3133 pts/0        R+     0:00 ps x
geeko@da10:~>

```

In the above example, the process with PID 3102 is a terminal-independent process.

The following are some commonly-used options with **ps**:

Table 5-1

Option	Description
a	Show all processes that have controlling terminals, including those of other users.
x	Show processes with and without controlling terminals.
-w, w	Provide detailed, wide output.
u	Display user-oriented format.
f	List processes hierarchically (in a tree format).
-l, l	long format

Table 5-1 *(continued)*

Option	Description
--------	-------------

U userlist	Select by effective user ID (EUID) or name
------------	--

For example, the output of entering **ps axl** is similar to the following:

```
geeko@da10:~> ps axl
F  UID  PID  PPID  PRI NI  VSZ  RSS   WCHAN  STAT  TTY    TIME COMMAND
...
0  1013  4170  4169  15  0   3840 1760  wait4   Ss    pts/0  0:00 -bash
0  1013  4332  4170  15  0   4452 1812  finish  T     pts/0  0:00 xeyes
0  1013  4351  4170  15  0   4452 1812  schedu  S     pts/0  0:01 xeyes
0  1013  4356  4170  17  0   2156  652  -       R+    pts/0  0:00 ps axl
```

However, the output of entering **ps aux** looks like the following:

```
geeko@da10:~> ps aux
USER  PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
geeko 4170  0.0  0.3   3840 1760 pts/0    Ss   12:10   0:00 -bash
geeko 4332  0.0  0.3   4452 1812 pts/0    T    12:59   0:00 xeyes
geeko 4351  0.3  0.3   4452 1812 pts/0    S    13:01   0:03 xeyes
geeko 4375  0.0  0.1   2156  680 pts/0    R+   13:19   0:00 ps aux
```

The basic difference is that with the option **l**, you see the process ID of the parent process (**PPID**), the process priority (**PRI**), and the nice value (**NI**) of the individual processes.

With the **u** option, the load percentage is shown (**%CPU**, **%MEM**).

The following is a description of some fields (columns) in the process list:

Table 5-2

Field	Description
-------	-------------

UID	User ID
-----	---------

PID	Process ID
-----	------------

PPID	Parent process ID
------	-------------------

Table 5-2 *(continued)*

Field	Description
TTY	Number of the controlling terminal
PRI	Priority number (the lower it is, the more computer time is allocated to the process)
NI (nice)	Influences the dynamic priority adjustment
STAT	Current process status (see Table 5-3)
TIME	CPU time used
COMMAND	Name of the command

These and other fields are explained in the manual page of `ps`.

The process state **STAT** can be one of the following:

Table 5-3

Code	Description
R (Runnable)	Process can be run
S (Sleeping)	Process is waiting for an external event (such as data arriving)
D (Uninterruptable sleep)	Process cannot be terminated at the moment
T (Traced or Stopped)	Process is suspended
X	Process is dead
Z (Zombie)	Process has terminated itself, but its return value has not yet been requested

You can format the output of `ps` to present the information you need:

```
geeko@dal0:~ > ps ax --format 'cputime %C, nice %n, name %c'
cputime %CPU, nice NI, name COMMAND
cputime 0.0, nice 0, name bash
cputime 0.0, nice 0, name xeyes
cputime 0.3, nice 0, name xeyes
cputime 0.0, nice 0, name ps
```

For detailed information about using the command `ps`, enter **man ps**.

pstree

With the command **pstree**, you can view a list of processes in the form of a tree structure. This gives you an overview of the hierarchy of a process.

To end a series of processes, find the appropriate parent process and end that instead. The option `-p` displays the PID of the processes. The option `-u` displays the user ID if the owner has changed.

Because the list of processes is often long, you can enter **pstree -up | less** to view part of the processes at a time.

nice and renice

Linux always tries to distribute the available computing time equitably to all processes. However, you might want to assign a process more or less computing time.

You can do this with the command **nice**, as in the following:

```
geeko@dal0:~ > nice -n +5 sleep 99
```


This command assigns a process a specific nice value that affects the calculation of the process priority (which is increased or decreased). If you do not enter a nice value, the process is started with the value +10.

The NI column in the top list (see Figure 5-3) contains the nice value of the process. The default value 0 is regarded as neutral. You can assign the nice level using a numeric value of -20 to 19.

The lower the value of the nice level, the higher the priority of the process. A process with a nice level of -20 runs at the highest priority; a process with a nice level of 19 runs at the lowest priority.

The nice level is used by the scheduler to determine how frequently to service a running process.

Only root is permitted to start a process with a negative nice value (such as **nice -n -3 sleep 99**). If a normal user attempts to do this, an error message is returned.

You can use the command `renice` to change the nice value of a running process, such as

```
geeko@dal10:~ > renice 5 1712
```

In this example, the command assigns the process with the PID 1712 the new nice value 5.

Only root can reduce the nice value of a running process (such as from 10 to 9 or from 3 to -2). All other users can only increase the nice value (such as from 10 to 11).

For example, if the user `geeko` attempts to assign the process 28056 that currently has a nice value of 3 to a nice value of 1, a “Permission denied” message is returned.

top

The command **top** allows to watch processes continuously in a list that is updated in short intervals, thus providing a real-time view of a running system. **top** can also be used to assign a new nice value to running processes or to end processes.

The information displayed in **top** can be filtered by a specific user, and can be sorted on any displayed field. By typing **r**, you can adjust the priority of a process, provided you have sufficient privileges to do so.



As with the command **renice**, the same restrictions apply when changing process nice levels using **top**. Non-root users can increase the nice level, but they cannot lower it.

When you enter **top**, a list similar to the following appears:

Figure 5-3

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3255	root	15	0	37148	12m	6264	S	3.7	5.1	1:11.34	X
7804	geeko	15	0	121m	15m	10m	S	0.7	6.3	0:01.37	gnome-terminal
7523	geeko	15	0	113m	16m	12m	S	0.3	6.4	0:01.98	nautilus
7552	geeko	15	0	108m	13m	11m	S	0.3	5.3	0:02.41	nm-applet
7553	geeko	15	0	18796	5548	4152	S	0.3	2.2	0:00.21	gnome-power-man
1	root	16	0	720	168	132	S	0.0	0.1	0:00.75	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	0:00.02	events/0
4	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	khelper
5	root	11	-5	0	0	0	S	0.0	0.0	0:00.00	kthread
7	root	10	-5	0	0	0	S	0.0	0.0	0:00.16	kblockd/0
8	root	20	-5	0	0	0	S	0.0	0.0	0:00.00	kacpid
96	root	15	0	0	0	0	S	0.0	0.0	0:00.37	pdfflush
97	root	15	0	0	0	0	S	0.0	0.0	0:00.54	pdfflush
99	root	18	-5	0	0	0	S	0.0	0.0	0:00.00	aio/0
98	root	15	0	0	0	0	S	0.0	0.0	0:00.22	kswapd0
305	root	11	-5	0	0	0	S	0.0	0.0	0:00.00	cqueue/0
306	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kseriod
346	root	11	-5	0	0	0	S	0.0	0.0	0:00.00	kpsmouse
721	root	11	-5	0	0	0	S	0.0	0.0	0:00.00	scsi_ah_0
816	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	reiserfs/0
879	root	12	-4	1836	536	348	S	0.0	0.2	0:00.64	udev
1406	root	11	-5	0	0	0	S	0.0	0.0	0:00.00	kgameportd

CNI USE ONLY-1 HARDCOPY PERMITTED

The displayed list is sorted by computing time and is updated every 3 seconds. You can terminate the display by typing **q**.

The following describes the default columns:

Table 5-4

Column	Description
PID	Process ID
USER	User name
PR	Priority
NI	Nice value
VIRT	Virtual Image (in KB)
RES	Resident Size (in KB)
SHR	Shared Mem Size (in KB)
S	Process Status
%CPU	CPU Usage
%MEM	Memory Usage (RES)
TIME+	CPU time
COMMAND	Command name/line

You can view the process management commands available in top by entering **? or h**. The following are some of the more commonly used commands:

Table 5-5

Command	Description
r	Assign a new nice value to a running process
k	Send a running process the termination signal (same as kill or killall)
N	Sort by process ID

Table 5-5 *(continued)*

Command	Description
P	Sort by CPU load
i	Show non-idle processes only

Command line options can be used to change the default behaviour of top.

top -d 5 (delay) changes the default delay (3 seconds) before refresh to 5 seconds.

top -b (batch mode) is useful when you want to write the output of top to a file or pass it to another process.

top -n 3 (iterations) causes top to quit after the third refresh. This is especially useful in combination with -b, for instance **top -b -n 1**.

End a Process

You can use the following to end the process:

- kill and killall
- Gnome System Monitor



You can also send a signal to end the process in top using the command k.

kill and killall

You can use the commands **kill** and **killall** to terminate a process. The command **killall** kills all processes with an indicated command name; the command **kill** kills only the indicated process.

The command `kill` requires the PID of the process (use `ps` or `top` to find the PID). The command `killall` needs the command name of the process.

For example, if you enter `xeyes` at the command line to start the **xeyes** program (and the PID is 18734), you can enter **`kill 18734`** or **`killall xeyes`** to end the process.

A process can do one of the following when receiving a signal:

- Capture the signal and react to it (if it has a corresponding function available). For example, an editor can close a file properly before it is terminated.

or

- Ignore the signal if no function exists for handling that signal.

However, the process does not have control over the following 2 signals as they are handled by the kernel:

- **`kill -SIGKILL`** or **`kill -9`**
- **`kill -STOP`** or **`kill -19`**

These signals cause the process to be ended immediately (**SIGKILL**) or to be stopped (**STOP**).

You should use **SIGKILL** with caution. Although the operating system closes all files that are still open, data in buffers is no longer processed. This means that some processes might leave the service in an undefined state, so it cannot easily be started again.



For a complete list of signals generated by `kill` and what their numbers stand for, enter **`kill -l`** or **`man 7 signal`**.

The following are the more commonly-used signals:

Table 5-6

Number	Name	Description
1	SIGHUP	Reload configuration file
2	SIGINT	Interrupt from keyboard (Ctrl+C)
9	SIGKILL	Kill process
15	SIGTERM	End process immediately (terminate process in a controlled manner so clean-up is possible)
18	SIGCONT	Continue process stopped with STOP
19	STOP	Stop process

For the kernel to forward the signal to the process, it must be sent by the owner of the process or by root. By default (without options), `kill` and `killall` send signal 15 (SIGTERM).

The following is the recommended way of ending an unwanted process:

1. Send SIGTERM by entering one of the following:

- ❑ **`kill PID`**

This is equivalent to **`kill -SIGTERM PID`** or **`kill -15 PID`**. You can use `killall` instead of `kill` and the command name of the process instead of the PID.

2. Wait a few moments for the process to be cleaned up.
3. If the process is still there, send a SIGKILL signal by entering one of the following:

- ❑ **`kill -SIGKILL PID`**

or

- ❑ **`kill -9 PID`**

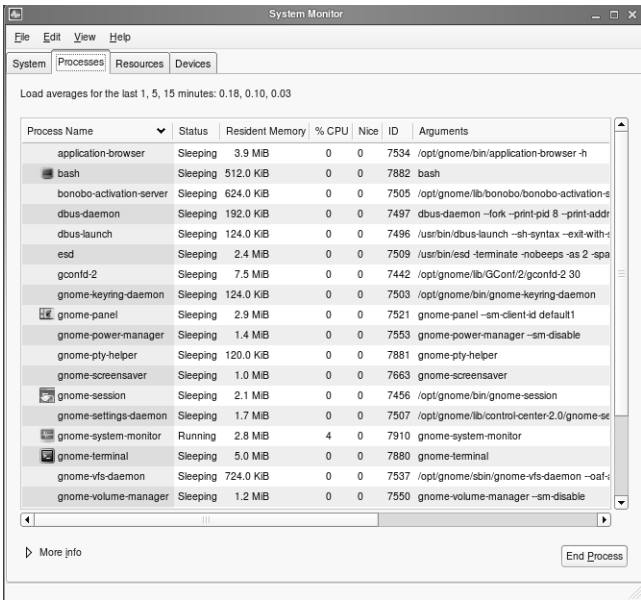
You can use `killall` instead of `kill` and the command name of the process instead of the PID.

If a process has been started from the bash shell, you can also use the job ID instead of the process number (such as `kill %4`).

Gnome System Monitor

From the Gnome desktop, you can start the utility Gnome System Monitor (**Computer > More Applications > GNOME System Monitor**) to view and kill processes:

Figure 5-4



If you encounter a misbehaving or hung process, you can kill it with Gnome System Monitor by selecting the *process* from the Process Table and selecting **End Process**.

The following information is displayed by default in columns in the Processes tab:

Table 5-7

Column	Description
Process Name	Name of the process
Status	Status of the process (running, sleeping, etc.)
Resident Memory	Actual memory occupied
CPU%	Processor load caused by system processes required for the process
Nice	Priority of the process when allocated computer time by the kernel
ID	Number of the process (Process ID)
Arguments	The start command for this process and the arguments used

You can customize what information is displayed by editing the preferences (**Edit > Preferences**).

Understand Services (Daemons)

A *service* is also called a *daemon* and is a process or collection of processes that wait for an event to trigger an action on the part of the program. In network based services, this event is a network connection. Other services, like cron and atd, are time-based and perform certain tasks at certain points in time.

These network-based services create a *listener* on a TCP or UDP port when they are started, usually during system startup. This listener waits for network traffic to appear on the designated port, and when traffic is detected, the program processes the traffic as input and generates output that is sent back to the requester.

For example, when a web browser connects to a web server, it sends a request to the web server, the web server processes the request and sends back its response. This response is then handled by the web browser, which would make the page human-readable.

Most network-based services work in a similar way, although the data is not always clear text data as in the web server example.

Manage a Daemon Process

Daemons run in the background and are usually started when the system is booted. Daemons make a number of services available.

For this reason, daemons are terminal-independent processes, and are indicated in the **ps x** TTY column by a “?”:

```
da10:~ # ps x
PID  TTY      STAT   TIME COMMAND
...
2767  ?        Ssl    0:00 /usr/sbin/nscd
...
```

In most cases, you can recognize a daemon by the ending “d” (such as syslogd or sshd). However, there are also a number of services where this is not the case (such as cron or portmap).

There are 2 types of daemons available:

- **Signal-controlled daemons.** These are always activated when a corresponding task exists (such as cupsd).
- **Interval-controlled daemons.** These are always activated at certain intervals (such as cron or atd).

For each daemon, there is a script in `/etc/init.d/`. Each script can be controlled and run with the following parameters:

Table 5-8

Parameter	Description
start	Starts the service
stop	Stops the service
reload (or restart)	Reloads the configuration file of the service, or stops the service and starts it again

For many scripts, there is a symbolic link in the directory `/usr/sbin/` or in the directory `/sbin/`, such as the following:

```
da10:~ # ls -l /usr/sbin/rcsshd
lrwxrwxrwx 1 root root 16 Jul 16 17:26 /usr/sbin/rcsshd ->
/etc/init.d/sshd
```

You can start the service from the directory `/etc/init.d/` (such as **`/etc/init.d/sshd start`**). If a link exists in the `/usr/sbin/` or `/sbin/`, you can also use **`rcservice`** (such as **`rcsshd start`**).

You can find configuration files for daemons in the directory `/etc/` or in a subdirectory of `/etc/`.

The executable programs (the actual daemons) are located either in the directory `/sbin/` or in the directory `/usr/sbin/`.



For documentation on most daemons, see `/usr/share/doc/packages/`.

Some important daemons include the following:

- **cron** Starts other processes at specified times.
- **cupsd** The printing daemon.
- **httpd** The daemon of the Apache2 web server

- **sshd** Enables secure communication by way of insecure networks (secure shell).
- **syslog-ng** Logs system messages in the directory `/var/log/`.

Exercise 5-1 Manage Linux Processes

In this exercise, you start and stop processes and change their priorities.

You will find this exercise in the workbook.

(End of Exercise)

Objective 2 **Schedule Jobs**

Most SUSE Linux Enterprise Server administrators and users find that they need to carry out certain tasks regularly on a running system (such as updating a database or backing up data).

You can automate these jobs in Linux by doing the following:

- Schedule a Job (cron)
- Run a Job One Time Only (at)

Schedule a Job (cron)

You can schedule jobs to be carried out on a regular basis by using the service cron (/usr/sbin/cron).

The service runs as a daemon and checks once a minute to see if jobs have been defined for the current time. By default, the service should be activated.

The file that contains the list of jobs is called a *crontab*. A crontab exists for the entire system as well as for each user defined on the system.

The file /etc/sysconfig/cron contains variables for the configuration of some scripts started by cron.

There are 2 types of jobs that can be defined with cron:

- System Jobs
- User Jobs

System Jobs

You control system jobs with the file `/etc/crontab`. After installation there is only one job defined that runs the scripts contained in the following directories in the intervals indicated:

Table 5-9

Directory	Interval
<code>/etc/cron.hourly</code>	Jobs are run on an hourly basis.
<code>/etc/cron.daily</code>	Jobs are run on a daily basis.
<code>/etc/cron.weekly</code>	Jobs are run on a weekly basis.
<code>/etc/cron.monthly</code>	Jobs are run on a monthly basis.

You can add lines to `/etc/crontab`, but you should not delete the lines added at installation.



For a detailed description of the syntax for `/etc/crontab`, enter **man 5 crontab**.

The scripts called from the file `/etc/crontab` not only ensure that the scripts are run at the prescribed intervals (handled by the script `/usr/lib/cron/run-crons`), but also that jobs are run later if they could not be run at the specified time.

For example, if a script could not be run at the specified time because the computer was turned off overnight, the script is automatically run later using the settings in `/etc/crontab`.

This is only valid for jobs defined in a script in `cron.hourly`, `cron.daily`, `cron.weekly`, or `cron.monthly`.

Information about the last time the jobs were run is kept in the directory `/var/spool/cron/lastrun/` in a file such as `cron.daily`.

The time stamp of the file is evaluated by the script `/usr/lib/cron/run-crons` to determine if scripts have to be run or not.

In a standard installation, only the directory `/etc/cron.daily/` contains scripts, as visible in the following:

```
da10:~ # ls -l /etc/cron*
-rw----- 1 root root  11 2006-05-08 20:47 /etc/cron.deny
-rw-r--r-- 1 root root 255 2006-05-08 20:47 /etc/crontab

/etc/cron.d:
insgesamt 0

/etc/cron.daily:
insgesamt 32
-rwxr-xr-x 1 root root  393 ... logrotate
-rwxr--r-- 1 root root  948 ... suse-clean_catman
-rwxr-xr-x 1 root root 1875 ... suse.de-backup-rc.config
-rwxr-xr-x 1 root root 2059 ... suse.de-backup-rpmdb
-rwxr-xr-x 1 root root  566 ... suse.de-check-battery
-rwxr-xr-x 1 root root 1320 ... suse.de-clean-tmp
-rwxr-xr-x 1 root root  371 ... suse.de-cron-local
-rwxr--r-- 1 root root 1196 ... suse-do_mandb

/etc/cron.hourly:
insgesamt 0

/etc/cron.monthly:
insgesamt 0

/etc/cron.weekly:
insgesamt 0
```

These shell scripts are overwritten when you update your system. Any modifications you made to these files get lost when these files are updated.

For this reason, it is advisable to write your own additions and modifications to `/root/bin/cron.daily.local` (see `/etc/cron.daily/suse.de-cron-local`), because this script is not overwritten when you update your system.

Other files for system jobs can be stored in the directory `/etc/cron.d/`. These files must have the same format as `/etc/crontab`. Jobs defined in `/etc/cron.d` are not run automatically at a later time.

User Jobs

The jobs of individual users are stored in the directory `/var/spool/cron/tabs/` in files matching the user names. These files always belong to the user root. Users create their own jobs using the command **crontab**.

The following are options for the command **crontab**:

Table 5-10

Option	Description
<code>crontab -e</code>	Creates or edits jobs. The vi editor is used.
<code>crontab file</code>	The specified <i>file</i> contains a list of jobs in the proper format and replaces any existing crontab file for that user.
<code>crontab -l</code>	Displays current jobs.
<code>crontab -r</code>	Deletes all jobs.

Each line in a file defines a job. There are 6 fields in a line.

The first 5 fields define the time, the final field contains the command to run. This can be any type of command or shell script. However, no user interaction is available when the command or shell script is run.

The first 5 fields have the following format:

Table 5-11

Field	Range
Minutes	0–59
Hours	0–23
Day of the Month	1–31
Month	0–12
Weekday	0–7

The following are guidelines for configuring these fields:

- If you want a job to run on every date, enter an asterisk (*) in the corresponding field.
- You can include several entries in a field, separated by commas.
- You can specify a range with start and end values separated by a hyphen.
- You can configure time steps with */n* (where *n* stands for the size of the step).
- You can specify months and weekdays by their first three letters (not case-sensitive). However, when you use letters, you cannot use ranges or lists.
- Numbers representing the weekdays start at 0 for Sunday and run through the entire week consecutively, with 7 representing Sunday again.

For example, 3 is Wednesday and 6 is Saturday.

The following is an example of a cron job entry:

```
*/10 8-17 * * 1-5 fetchmail mailserver
```

In this example, from Monday to Friday (**1-5**) every 10 minutes (***/10**) between 8.00 and 17.00 (**8-17**), the command **fetchmail** is run to fetch incoming emails from the computer **mailserver**.

For system jobs, the user who has the permissions to run the command must be specified in the file `/etc/crontab`, by entering the user name between the time details (the first 5 fields) and the name of the command (which now becomes the seventh field).

Run a Job One Time Only (at)

If you want to run a job one time only (instead of scheduling it on a regular basis with cron) you can use the command **at**. To use **at**, you must make sure the service **atd** is started.

There are 2 files that determine which users can run this command:

- **/etc/at.allow** (users entered here can define jobs)
- **/etc/at.deny** (users who are not listed in this file can define jobs)

These files are text files you can modify or create.

By default, the file **/etc/at.deny** already exists with its own entries, such as the following:

```
alias
backup
bin
daemon
ftp
games...
```

If the file **/etc/at.allow** exists, only this file is evaluated. If neither of these files exist, only the user **root** can define jobs with **at**.

You define a job from a command prompt by entering **at *launch_time*** (where ***launch_time*** is when you want the job to begin, for instance 12:34).

At this point you are placed in a special environment where you enter commands 1 line at a time. When you finish entering commands, you save the job by pressing **Ctrl+D**.

The following is an example of creating a job with the command `at`:

```
geeko@dal0:~> at 21:00
warning: commands will be executed using /bin/sh
at> /home/geeko/bin/doit
at> mail -s "Results file of geeko" geeko@dal0 < /home/geeko/results
at> <EOT>
job 4 at 2004-08-27 21:00
```

If the commands you want executed are contained in a text file, you need to enter **`at -f file launch_time`** (where *file* is the pathname of the file).

The following are some other commonly-used commands and options for `at`:

Table 5-12

Command	Description
<code>atq</code>	Display defined jobs (including job numbers, which are needed to delete a job)
<code>atrm job_number</code>	Delete a job (using the job number)

Exercise 5-2 Schedule Jobs with cron and at

In this exercise, you practice scheduling jobs with at and cron.

You will find this exercise in the workbook.

(End of Exercise)

Summary

Objective	Summary
1. View and Manage Processes	<p>To manage processes on your SUSE Linux Enterprise Server, you learned about the following:</p> <ul style="list-style-type: none">■ Understand Process Definitions■ Learn Jobs and Processes■ Manage Foreground and Background Processes■ View and Prioritize Processes■ End a Process■ Understand Services (Daemons)■ Manage a Daemon Process
2. Schedule Jobs	<p>Most SUSE Linux Enterprise Server administrators and regular users find that they need to carry out certain tasks regularly on a running system (such as updating a database or backing up data).</p> <p>You learned how to automate these jobs in Linux by doing the following:</p> <ul style="list-style-type: none">■ Schedule a Job (cron)■ Run a Job One Time Only (at)

CNI USE ONLY-1 HARDCOPY PERMITTED

SECTION 6 Monitor SUSE Linux Enterprise Server 10

In this section you learn how to monitor your SUSE Linux Enterprise Server 10 system, how to configure system logging, and how to monitor logins.

Objectives

1. Monitor a SUSE Linux Enterprise Server 10 System
2. Use System Logging Services
3. Monitor Login Activity

Objective 1 **Monitor a SUSE Linux Enterprise Server 10 System**

As a system administrator, you sometimes have questions similar to the following;

- Did the system boot normally?
- What is the kernel version?
- What services are running?
- What is the load on the system?

In this objective, you are introduced to tools that help you discover information about your hardware and Linux system:

- Boot Log Information
- Hardware Information (/proc/)
- Hardware Information (Command Line Utilities)
- System and Process Information (Command Line Utilities)
- Monitor Hard Drive Space

Boot Log Information

When SUSE Linux Enterprise Server 10 starts, some lines scroll by too quickly for you to read easily. If there is an error message, it might be nearly impossible to read it.

These messages are kept in the kernel ring buffer. As the capacity of this buffer is limited, older entries in the ring buffer are deleted when new entries are added to it.

To have the boot messages available even when they have been deleted from the buffer, they are written to the file **/var/log/boot.msg** in a slightly modified format after booting the machine. For each line displayed at the console during startup, there is one or several lines in the file **/var/log/boot.msg**.

dmesg is the command used to view the current content of the kernel ring buffer. **dmesg | less** allows you to scroll up and down in the output, which looks similar to the following:

```
Linux version 2.6.16.14-6-smp (geeko@buildhost) (gcc version 4.1.0 (SUSE
Linux)) #1 SMP Tue May 9 12:09:06 UTC 2006
BIOS-provided physical RAM map:
  BIOS-e820: 0000000000000000 - 000000000009fc00 (usable)
  BIOS-e820: 000000000009fc00 - 00000000000a0000 (reserved)
  BIOS-e820: 00000000000f0000 - 0000000000100000 (reserved)
  BIOS-e820: 0000000000100000 - 000000001ffffd000 (usable)
  BIOS-e820: 000000001ffffd000 - 000000001ffff000 (ACPI data)
  BIOS-e820: 000000001ffff000 - 0000000020000000 (ACPI NVS)
  BIOS-e820: 00000000fec00000 - 00000000fec01000 (reserved)
  BIOS-e820: 00000000fee00000 - 00000000fee01000 (reserved)
  BIOS-e820: 00000000ffff0000 - 0000000100000000 (reserved)
0MB HIGHMEM available.
511MB LOWMEM available.
found SMP MP-table at 000f6e60
On node 0 totalpages: 131069
  DMA zone: 4096 pages, LIFO batch:0
  DMA32 zone: 0 pages, LIFO batch:0
  Normal zone: 126973 pages, LIFO batch:31
  HighMem zone: 0 pages, LIFO batch:0
DMI 2.0 present.
Using APIC driver default
ASUS P2B-DS detected: force use of acpi=ht
lines 1-22
```

The output of **dmesg** shows messages generated during the initialization of the hardware by the kernel or kernel modules.

The file **/var/log/boot.msg** contains additional information beyond what you can display with **dmesg**.

This information includes data such as the messages the various scripts generated at boot time and exit status codes, as in the following:

```
...
System Boot Control: The system has been set up
Skipped features: boot.cycle
System Boot Control: Running /etc/init.d/boot.local
done<notice>killproc: kill(874,3)

INIT: Entering runlevel: 5

Boot logging started on /dev/tty1(/dev/console) at Wed May 24 10:31:51
2006

Master Resource Control: previous runlevel: N, switching to runlevel: 5
Loading AppArmor profiles - AppArmor already loaded with profiles. Not
loading profiles. warning
Initializing random number generatordone
<notice>startproc: execve (/usr/bin/dbus-daemon) [ /usr/bin/dbus-daemon
--system ], [ CONSOLE=/dev/console ROOTFS_FSTYPE=reiserfs TERM=linux
SHELL=/bin/sh ROOTFS_FSCK=0 LC_ALL=POSIX INIT_VERSION=sysvinit-2.86
REDIRECT=/dev/tty1 COLUMNS=123 PATH=/bin:/usr/bin:/sbin:/usr/sbin
vga=0x317 RUNLEVEL=5 PWD=/
SPLASHCFG=/etc/bootsplash/themes/SuSE-SLES/config/bootsplash-1024x768.cfg
PREVLEVEL=N LINES=44 SHLVL=2 HOME=/ splash=silent SPLASH=yes
ROOTFS_BLKDEV=/dev/sda2 _=/sbin/startproc DAEMON=/usr/bin/dbus-daemon ]
acpid: no ACPI support in kernelskipped
Starting D-BUS daemondone
...
```

These additional messages can be useful when troubleshooting.

You can also use YaST to view the file contents by starting YaST and then selecting **Miscellaneous > View Start-up Log**. Or you start the module directly with **yast2 view_anymsg** in a terminal window as root.

Hardware Information (/proc/)

The directory `/proc/` contains a lot of information on the running system, including hardware information stored in the kernel memory space.

For example, if you enter `cat /proc/cpuinfo`, output is generated from data stored in kernel memory that gives you information such as the CPU model name and cache size.

You can view the available information by using commands such as `cat`, `more`, or `less` with a file name (such as **`cat /proc/cpuinfo`**).

The following are some of the commonly-used filenames to generate information:

- **`/proc/devices`**. View the devices used on your Linux system.
- **`/proc/cpuinfo`**. View processor information.
- **`/proc/ioproports`**. View the I/O ports on your server. The I/O ports are the addresses of various hardware devices.
- **`/proc/interrupts`**. View the IRQ (hardware interrupt signal) assignments for your Linux system.
- **`/proc/dma`**. View the DMA (Direct Memory Access) channels used on your Linux system.
- **`/proc/bus/pci/devices`**. View the PCI (Peripheral Component Interconnect) information on your Linux system.
- **`/proc/scsi/scsi`**. View a summary of the SCSI (Small Computer System Interface) information on your Linux system.

Hardware Information (Command Line Utilities)

The following are utilities you can use from the command line to view information about the hardware on your Linux system:

- **hwinfo.** Entering this command generates and displays a list of specific information about the devices installed on your Linux system. If you want to be able to scroll up and down the list, enter **hwinfo | less**.

For a summary listing, enter **hwinfo --short**. **hwinfo --log *filename*** writes the information to a log file.

- **hdparm.** Entering this command with various options lets you view information about your hard drive and manage certain hard drive parameters.

For example, the option **-i** displays hard drive identification information available at boot time. The option **-l** requests information directly from the hard drive.

For a summary list of available options, enter **hdparm** or **hdparm -h**.

- **fdisk.** While this command is primarily used for managing the partition table on a Linux system, you can also use options such as **-l** (list partition tables), **-s** (size of partition) to view hard drive information.

- **iostat.** Entering this command displays CPU and input/output (I/O) statistics for devices and partitions. The program **iostat** is part of the package **sysstat**.

This command generates reports that can be used to change system configuration to better balance the input/output load between physical disks.

The first report generated provides statistics concerning the time since the system was booted. Each subsequent report covers the time since the previous report.

You can generate 2 types of reports with the command—the CPU usage report and the device usage report.

The option **-c** generates only the CPU usage report; the option **-d** generates only the device usage report.

- **lspci.** Entering this command displays information about all PCI buses in your Linux system and all devices connected to them.

The options `-v` and `-vv` generate verbose reports. The option `-b` gives you a bus-centric view of all the IRQ numbers and addresses as seen by the cards (instead of the kernel) on the PCI bus.

- **sig.** System Information GAttering. It collects various information on your system and outputs it in HTML or ASCII format.
- **sitar.** System InformaTion At Runtime. Prepare system information using Perl, reading the `/proc` filesystem. Output is written to `/tmp` in HTML, LaTeX, and simplified doc-book-xml.

System and Process Information (Command Line Utilities)

Besides `ps` and `top`, which have been covered in the Section “Administer Linux Processes and Services” on page 5-1, there are several more commonly-used command line tools for viewing system information:

- `uptime`
- `netstat`
- `uname`

uptime

Although the command `top` gives you system information in the header, there might be times when you only want specific information without starting a utility.

For example, you can use the command **uptime** to display the current time, the length of time the system has been running, the number of users on the system, and the average number of jobs in the run queue over the last 1, 5, and 15 minutes.

The following is an example of entering the command **uptime**:

```
geeko@dal0:~> uptime
1:13pm up 2:42, 1 user, load average: 0.02, 0.12, 0.09
```

For additional information on the **uptime** command, enter **man uptime**.

netstat

While the command **ps** provides information on a process level, you can use **netstat** to find out which network ports are offering services and what connections are established, as in the following:

```
dal0:~ # netstat -patune
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
User      Inode      PID/Program name
tcp        0      0 149.44.87.34:427        0.0.0.0:*               LISTEN
0          6812       2723/slpd
tcp        0      0 127.0.0.1:427           0.0.0.0:*               LISTEN
0          6811       2723/slpd
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN
0          6774       2689/portmap
tcp        0      0 127.0.0.1:2544          0.0.0.0:*               LISTEN
0          6862       2714/zmd
tcp        0      0 0.0.0.0:631             0.0.0.0:*               LISTEN
0          8614       2858/cupsd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN
0          7529       3055/master
...
```

CNI USE ONLY-1 HARDCOPY PERMITTED

The following are some useful options for customizing the output of netstat:

Table 6-1

Option	Description
-p	Show processes (as root)
-a	Show listening and non listening sockets (all)
-t	Show tcp information
-u	Show udp information
-n	Do not resolve hostnames
-e	Display additional information (extend)
-r	Display routing information

uname

You can use the command **uname** to find out about the current kernel version, as in the following:

```
da10:~ # uname -a
Linux da10 2.6.16.14-6-smp #1 SMP Tue May 9 12:09:06 UTC 2006 i686 i686
i386 GNU/Linux
```

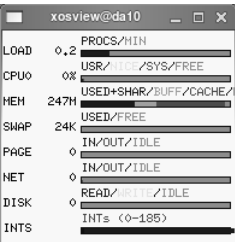
xosview

From the graphical desktop, you can use the utility **xosview** (package xosview) to display the status of several system-based parameters such as CPU usage, load average, memory usage, swap space usage, network usage, interrupts, and serial port status.

To start xosview, open a terminal window and enter **xosview &**.

A window similar to the following appears:

Figure 6-1



Each parameter status is displayed as a horizontal bar separated into color-coded regions. Each region represents a percentage of the resource that is being put to a particular use.

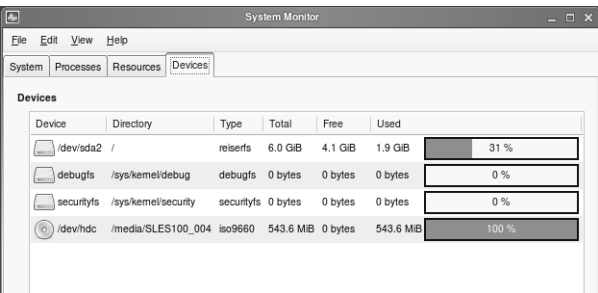
When you finish viewing the information, you can quit by closing the window or by typing **q**.

Monitor Hard Drive Space

The command line tools **df** and **du** have already been mentioned in “Check Partition and File Usage (**df** and **du**)” on page 2-44.

As a graphical tool equivalent to **df**, you can use the Gnome System Monitor (**Computer > More Applications > System**), selecting the **Devices** tab:

Figure 6-2



Exercise 6-1 *Gather Information About Your SUSE Linux Enterprise Server 10 Server*

In this exercise, you practice using the tools covered in this objective, “Monitor a SUSE Linux Enterprise Server 10 System” on page 6-2, to get information on the computer you are using.

You will find this exercise in the workbook.

(End of Exercise)

Objective 2 Use System Logging Services

In a Linux system, there are many logs that track various aspects of system operation. Many services log their activities to their own log files, and the level of detail can be set on a per-service basis. In addition, system logs in **/var/log/** track system-level events.

The information logged in these log files is typically used to assist in troubleshooting and for security purposes. Especially the latter mandates that the log files are reviewed regularly.

To use system logging services, you need to understand the following:

- The Syslog Daemon **syslog-ng**
- Important Log Files
- Archive Log Files (logrotate)

The Syslog Daemon syslog-ng

The syslog daemon **syslog-ng** is used by many services to log system events. The advantage in using a single service for logging is that all logging can be managed from one configuration file.

Up to SUSE Linux Enterprise Server 9, **syslogd** was used to log system events. With SUSE Linux Enterprise Server 10 these events are logged by **syslog-ng**, the new generation **syslogd**.

The main advantage of **syslog-ng** over **syslogd** is its capability to filter messages not only based on facilities and priorities, but also based on the content of each message.

The syslog daemon accepts messages from system services, and, depending on its configuration, other hosts, and logs them based on settings in the configuration files **/etc/sysconfig/syslog** and **/etc/syslog-ng/syslog-ng.conf**. The file **/etc/syslog-ng/syslog-ng.conf** is generated by SuSEconfig from **/etc/syslog-ng/syslog-ng.conf.in**. Both files share the same syntax.

The configuration of syslog-ng is distributed across three files:

- **/etc/sysconfig/syslog**
- **/etc/syslog-ng/syslog-ng.conf.in**
- **/etc/syslog-ng/syslog-ng.conf**

/etc/sysconfig/syslog

The file **/etc/sysconfig/syslog** contains general parameters applicable to syslog-ng as well as syslogd.

Parameters set in this file include switches passed to syslogd or syslog-ng, kernel log level, parameters for klogd, and which syslog daemon is to be used.

```
...
## Type:                string
## Default:             " "
## Config:              " "
## ServiceRestart:     syslog
#
# if not empty: parameters for syslogd
# for example SYSLOGD_PARAMS="-r -s my.dom.ain"
#
SYSLOGD_PARAMS=" "

## Type:                string
## Default:             -x
## Config:              " "
## ServiceRestart:     syslog
#
# if not empty: parameters for klogd
# for example KLOGD_PARAMS="-x" to avoid (duplicate) symbol
resolution
#
KLOGD_PARAMS="-x"

## Type:                list(syslogd,syslog-ng)
## Default:             syslogd
## Config:              syslog-ng
## Command:             /sbin/rcsyslog restart
## PreSaveCommand:     /sbin/rcsyslog status &&
/sbin/rcsyslog stop
#
# The name of the syslog daemon used as
# syslog service: "syslogd", "syslog-ng"
#
SYSLOG_DAEMON="syslog-ng"
...
```

Parameters set in `/etc/sysconfig/syslog` are evaluated by the start script `/etc/init.d/syslog`. Furthermore, `SuSEconfig` uses `/etc/sysconfig/syslog` to add log sockets to the file `/etc/syslog-ng/syslog-ng.conf` when generating this file from `/etc/syslog-ng/syslog-ng.conf.in`.

`/etc/syslog-ng/syslog-ng.conf.in`

`/etc/syslog-ng/syslog-ng.conf.in` is the template used to create the configuration file **`/etc/syslog-ng/syslog-ng.conf`**, which is the configuration file actually used by `syslog-ng`. Both files have the same syntax.

However, unless you turn off generation of `/etc/syslog-ng/syslog-ng.conf` in `/etc/sysconfig/syslog`, any manual changes to this file will be overwritten when `SuSEconfig` is executed.

Therefore, changes to the configuration of `syslog-ng` should be made in this file.

`/etc/syslog-ng/syslog-ng.conf`

`syslogd` and `syslog-ng` share two concepts that you have to understand to be able to configure either one:

- Facilities
- Priorities

The configuration of `syslog-ng` consists of several parts which are then combined to configure which information is logged where.

These are:

- Sources
- Filters

- Destinations
- Log Paths

Facilities

The facility refers to the subsystem that provides the corresponding message. Each program that uses syslog for logging is assigned such a facility, usually by its developer.

The following describes these facilities:

Table 6-2

Facility	Description
authpriv	Used by all services that have anything to do with system security or authorization. All PAM messages use this facility. The ssh daemon uses the auth facility.
cron	Accepts messages from the cron and at daemons.
daemon	Used by various daemons that do not have their own facility, such as the ppp daemon.
kern	All kernel messages.
lpr	Messages from the printer system.
mail	Messages from the mail system. This is important because many messages can arrive very quickly.
news	Messages from the news system. As with the mail system, many messages might need to be logged in a short time.
syslog	Internal messages of the syslog daemon.
user	A general facility for messages on a user level. For example, It is used by login to log failed login attempts.

Table 6-2 *(continued)*

Facility	Description
uucp	Messages from the uucp system.
local0 – local7	<p>These 8 facilities are available for your own configuration. All of the local categories can be used in your own programs.</p> <p>By configuring one of these facilities, messages from your own programs can be administered individually through entries in the file <code>/etc/syslog-ng/syslog-ng.conf</code>.</p>

Priorities

The priority gives details about the urgency of the message. The following priorities are available (listed in increasing degree of urgency):

Table 6-3

Priority	Description
debug	Should only be used for debugging purposes, since all messages of this category and higher are logged.
info	Used for messages that are purely informative.
notice	Used for messages that describe normal system states that should be noted.
warning	Used for messages displaying deviations from the normal state.
err	Used for messages displaying errors.
crit	Used for messages on critical conditions for the specified program.
alert	Used for messages that inform the system administrator that immediate action is required to keep the system functioning.

Table 6-3 *(continued)*

Priority	Description
emerg	Used for messages that warn you that the system is no longer usable.

Sources

A source is a collection of source drivers, which collect messages using a given method. These sources are used to gather log messages. The general syntax is as follows:

```
source <identifier> { source-driver(params); source-driver(params); ... };
```

The respective section in `/etc/syslog-ng/syslog-ng.conf` looks like this:

```
source src {
    # include internal syslog-ng messages
    # note: the internal() source is required!
    internal();

    # the following line will be replaced by the
    # socket list generated by SuSEconfig using
    # variables from /etc/sysconfig/syslog:
    unix-dgram("/dev/log");

    # uncomment to process log messages from network:
    #udp(ip("0.0.0.0") port(514));
};
```

In this example, one source for internal messages of syslog-ng and the socket `/dev/log` are defined.

Filters

Filters are boolean expressions that are applied to messages and are evaluated as either true or false. The general syntax is as follows:

```
filter <identifier> { expression; };
```

The identifier has to be unique within the configuration and is used later to configure the actual logging.

The following excerpt of `/etc/syslog-ng/syslog-ng.conf` shows some filters used in SUSE Linux Enterprise Server 10:

```
#
# Filter definitions
#
filter f_iptables    { facility(kern) and match("IN=") and match("OUT=");
};

filter f_console     { level(warn) and facility(kern) and not
                      filter(f_iptables) or level(err) and not facility(authpriv); };

filter f_newsnotice { level(notice) and facility(news); };
filter f_newscrit   { level(crit)   and facility(news); };
filter f_newserr    { level(err)    and facility(news); };
filter f_news       { facility(news); };
...
filter f_messages   { not facility(news, mail)
                      and not filter(f_iptables); };
...
```

As you can see, facility and priority (level) can be used within filters. However, it is also possible to filter according to the content of a line being logged, as in the `f_iptables` filter above.

Combining the expressions with “and”, “or”, or “and not” allows you to create very specific filters.

Destinations

Destinations defines where messages can be logged. The general syntax is as follows:

```
destination <identifier> {  
    destination-driver(params);  
    destination-driver(params); ... };
```

Possible destinations are files, fifos, sockets, ttys of certain users, programs, or other hosts.

A sample from `/etc/syslog-ng/syslog-ng.conf` looks like this:

```
destination console { file("/dev/tty10"    group(tty) perm(0620)); };  
destination messages { file("/var/log/messages"); };
```

Log Paths

Log paths are the point where it all comes together. They define which messages are logged where, depending on source, filter, and destination. The general syntax is as follows:

```
log { source(s1); source(s2); ...  
      filter(f1); filter(f2); ...  
      destination(d1); destination(d2); ...  
      flags(flag1[, flag2...]); };
```

The following entries in `/etc/syslog-ng/syslog-ng.conf` for instance are responsible for logging to `/dev/tty10` and `/var/log/messages`:

```
log { source(src); filter(f_console); destination(console); };  
log { source(src); filter(f_messages); destination(messages); };
```

In the first line, log messages that come in through sources defined in source `src` are logged to `tty10` if they match the filter `f_console`. In line two, messages that come in through sources defined in source `src` are logged to `/var/log/messages` if they match the filter `f_messages`.



For further details on the `syslog-ng.conf` file, enter **man 5 syslog-ng.conf**. The documentation in `/usr/share/doc/packages/syslog-ng/html/book1.html` gives a general overview of `syslog-ng` as well as details on the configuration.

Important Log Files

The log file to which most messages are written is the file **`/var/log/messages`**. Often hints can be found here about problems such as why a service does not function properly when it starts. If there is no hint in `/var/log/messages`, then a look at `/var/log/audit/audit.log`, the log file for AppArmor messages, might help. Firewall messages are logged in **`/var/log/firewall`**.

The best approach for reading the log files from the command line is to use the command `tail` (**`tail /var/log/messages`**). This displays the last 10 lines of the file, which are also the most current entries.

By using **`tail -n`** (such as **`tail -n 30`**) you can specify the number of lines to display.

If you want to have new messages displayed immediately, use the interactive mode with **`tail -f`**. For example, entering **`tail -20f /var/log/messages`** switches `tail` to interactive mode. The last 20 lines of the file `/var/log/messages` are displayed. If new messages are added these are displayed immediately.

You can stop `tail -f` by pressing **`Ctrl+c`**.

The following are important log files stored in the directory /var/log/:

Table 6-4

Log File	Description
/var/log/audit/	This directory stores the Novell AppArmor logfile audit.log.
/var/log/cups/	This directory stores the log files for the printing system CUPS.
/var/log/news/	This directory stores messages for the news system.
/var/log/YaST2/	This directory stores log files for YaST.
/var/log/boot.msg	<p>When the system boots, all boot script messages are displayed on the first virtual console. This often happens so fast that you cannot read all the messages.</p> <p>You can, however, read the boot messages in this file.</p>
/var/log/mail	<p>Messages from the mail system are written to this file. Because this system often generates a lot of messages, there are additional log files:</p> <ul style="list-style-type: none">■ /var/log/mail.err■ /var/log/mail.info■ /var/log/mail.warn
/var/log/wtmp	<p>This file contains information about which user was logged in from where and for how long (since the file was created).</p> <p>The file contents are in binary form and can only be displayed with the command last (/usr/bin/last).</p> <p>Because of the binary format, it is difficult to manipulate entries in this file.</p>

Table 6-4 *(continued)*

Log File	Description
<code>/var/log/lastlog</code>	<p>This file contains information about which user was last logged in, from where, and for how long.</p> <p>You can only view the contents with the command lastlog (<code>/usr/bin/lastlog</code>).</p>

Archive Log Files (logrotate)

It is important to ensure that log files do not get too large and require too much space inside the system. For this reason, the size and age of log files are monitored automatically by the program **logrotate** (`/usr/sbin/logrotate`).

The program is run daily by the cron daemon (**/etc/cron.daily/logrotate**). The program checks all log files listed in its configuration files and takes any action required by the configuration for the respective file.

You can configure the settings in the files to indicate whether files should be compressed or deleted in regular intervals or when a determined size is reached.

You can also configure how many compressed versions of a log file are kept over a specified period of time. Log files can also be forwarded by email.

The configuration file of logrotate is **/etc/logrotate.conf**, which contains general configuration settings. The following is an example of 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

# uncomment these to switch compression to bzip2
#compresscmd /usr/bin/bzip2
#uncompresscmd /usr/bin/bunzip2

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

The following table describes the options in the file:

Table 6-5

Option	Description
weekly	The log files are created or replaced once a week.
rotate 4	Unless the option rotate is specified, the old files are deleted. In this example, the last 4 versions of the log file are kept (rotate 4).
create	The old file is saved under a new name and a new, empty log file is created.
compress	If the option compress is activated, the copies are stored in a compressed form.

CNI USE ONLY-1 HARDCOPY PERMITTED

Many RPM packages contain preconfigured files for evaluation by logrotate, which are stored in **/etc/logrotate.d/**. The files contained in that directory are read by logrotate through the **include /etc/logrotate.d** entry in **/etc/logrotate.conf**.

Any settings in the logrotate.d files supersede the general settings in logrotate.conf.

All the files to monitor must be listed. This is done through the entries in **/etc/logrotate.conf** (such as **/var/log/wtmp** [*options*]) or in separate configuration files.

The following is an example of the file syslog in **/etc/logrotate.d/**:

```
#
# Please note, that changing of log file permissions in this
# file is not sufficient if syslog-ng is used as log daemon.
# It is required to specify the permissions in the syslog-ng
# configuration /etc/syslog-ng/syslog-ng.conf.in as well.
#
/var/log/warn /var/log/messages /var/log/allmessages
/var/log/localmessages /var/log/firewall {
    compress
    dateext
    maxage 365
    rotate 99
    missingok
    notifempty
    size +4096k
    create 640 root root
    sharedscripts
    postrotate
        /etc/init.d/syslog reload
    endscript
}
...
```

The files syslog and syslog-ng in **/etc/logrotate.d/** contain settings for configuring how the log files written by syslog (syslogd or syslog-ng) will be treated.

The following describes the options in the file:

Table 6-6

Option	Description
size +4096k	Files will not be rotated weekly, but as soon as they reach a size of 4096 KB.
rotate 99	Ninety-nine versions of each of the files will be kept.
compress	The old log files will be stored compressed.
maxage 365	As soon as a compressed file is older than 365 days, it is deleted.
notifempty	If a log file is empty, no rotation takes place.
create 640 root root	New log files are created after the rotation and owner, group, and permissions are specified.
postrotate . . . endscrip	Scripts can be called after the rotation. For example, some services have to be restarted after log files have been changed. In this example, the syslog daemon will reread its configuration files after the rotation (/etc/init.d/syslog reload). As this script is the same for syslogd and syslog-ng, as far as logrotate is concerned, it does not matter which one is used.

Most of the services whose log files should be monitored come with preconfigured files, so only minor adjustments are normally needed.



For a complete list of all possible options, enter **man logrotate**.

Exercise 6-2 *Manage System Logging*

In this exercise, you practice configuring syslog-ng and logrotate.

You will find this exercise in the workbook.

(End of Exercise)

Objective 3 Monitor Login Activity

One of the most critical tasks you have as an administrator is to make sure that any suspicious activity on your system that might indicate a compromise of security is noticed and acted upon.

Monitoring tasks include evaluating login activity for signs of security breach such as multiple failed logins.



Reviewing files such as `/var/log/messages` also gives you information about login activity.

To monitor login activity, you can use the following commands:

- **who.** This command shows who is currently logged in to the system and information such as the time of the last login.

You can use options such as **-H** (display column headings), **-r** (current runlevel), and **-a** (display information provided by most options).

For example, entering **who -H** returns information similar to the following:

```
da10:~ # who -H
NAME      LINE          TIME          COMMENT
root      pts/0         2006-05-24 10:33 (da1.digitalairlines.com)
geeko     :0            2006-05-24 13:54
geeko     pts/1         2006-05-24 13:54
```

- **w.** This command displays information about the users currently on the machine and their processes.

The first line includes information on the current time, how long the system has been running, how many users are currently logged on, and the system load averages for the past 1, 5, and 15 minutes.

Below the first line is an entry for each user that displays the login name, the tty name, the remote host, login time, idle time, JCPU, PCPU, and the command line of the user's current process.

The JCPU time is the time used by all processes attached to the tty. It does not include past background jobs, but does include currently running background jobs.

The PCPU time is the time used by the current process, named in the What field.

You can use options such as **-h** (don't display the header), **-s** (don't display the login time, JCPU, and PCPU), and **-V** (display version information).

For example, entering **w** returns information similar to the following:

```
da10:~ # w
15:06:45 up 4:35, 4 users, load average: 0.00, 0.00, 0.00
USER      TTY      LOGIN@  IDLE   JCPU   PCPU WHAT
root      pts/0    10:33   0.00s  0.73s  0.02s w
geeko     :0       13:54   ?xdm?  1:15   0.58s /bin/sh
/opt/kde3/bin/startkde
...
```

- **finger.** This command displays information about local and remote system users. By default, the following information is displayed about each user currently logged in to the local host:

- Login name
- User's full name
- Associated terminal name
- Idle time
- Login time (and from where)

You can use options such as **-l** (long format) and **-s** (short format).

For example, entering **finger -s** returns information similar to the following:

```
da10:~ # finger -s
Login      Name            Tty      Idle   Login Time   Where
geeko      Geeko           *:0      -      Wed 13:54
geeko      Geeko           pts/1    1:13   Wed 13:54
geeko      Geeko           *pts/3   1:02   Wed 13:55
root       root           pts/0    -      Wed 10:33 da1.digitalairl
```

- **last.** This command displays a listing of the last logged in users.

Last searches back through the file `/var/log/wtmp` (or the file designated by the option `-f`) and displays a list of all users logged in (and out) since the file was created.

You can specify names of users and tty's to only show information for those entries.

You can use options such as `-num` (where *num* is the number of lines to display), `-a` (display the hostname in the last column), and `-x` (display system shutdown entries and runlevel changes).

For example, entering **last -ax** returns information similar to the following:

```
da10:~ # last -ax
geeko    pts/3      Wed May 24 13:55   still logged in
geeko    pts/1      Wed May 24 13:54   still logged in
geeko    :0         Wed May 24 13:54   still logged in
geeko    :0         Wed May 24 13:45 - 13:53 (00:08)
root     pts/0      Wed May 24 10:33   still logged in da1.digitalairlin
runlevel (to lvl 5) Wed May 24 10:31 - 15:09 (04:37)    2.6.16.14-6-smp
reboot   system boot Wed May 24 10:31   (04:38)    2.6.16.14-6-smp
shutdown system down Tue May 23 17:30 - 15:09 (21:39)    2.6.16.14-6-smp
...
```

- **lastlog.** This command formats and prints the contents of the last login log file (`/var/log/lastlog`). The login name, port, and last login time are displayed.

Entering the command without options displays the entries sorted by numerical ID.

You can use options such as `-u login_name` (display information for designated user only) and `-h` (display a one-line help message).

If a user has never logged in, the message `**Never logged in**` is displayed instead of the port and time.

For example, entering **lastlog** returns information similar to the following:

```
dal0:~ # lastlog
Username          Port      Latest
at                **Never  logged in**
bin               **Never  logged in**
...
root              pts/0     Wed May 24 10:33:36 +0200 2006
sshd              **Never  logged in**
suse-ncc          **Never  logged in**
uucp              **Never  logged in**
wwwrun           **Never  logged in**
geeko             :0       Wed May 24 13:54:29 +0200 2006
...
```

- **faillog**. This command formats and displays the contents of the failure log (`/var/log/faillog`) and maintains failure counts and limits.

The faillog functionality has to be enabled by adding the module **pam_tally.so** to the respective file in `/etc/pam.d/`, for instance `/etc/pam.d/login`:

```
##PAM-1.0
auth      required      pam_securetty.so
auth      required      pam_tally.so no_magic_root per_user
auth      include       common-auth
auth      required      pam_nologin.so
account   required      pam_tally.so    no_magic_root
...
```

The rest of the file does not need to be changed.

If you want to have this functionality with graphical logins as well, add the above line to `/etc/pam.d/xdm` and/or `/etc/pam.d/gdm`, depending on which login manager you use.

You can use options such as **-u** *login_name* (display information for designated user only) and **-p** (display in UID order).

The command `faillog` only prints out users with no successful login since the last failure. To print out a user who has had a successful login since his last failure, you must explicitly request the user with the **-u** option.

Entering **faillog** returns information similar to the following:

```
da10:~ # faillog
Login      Failures Maximum Latest           On
geeko      1          3    05/24/06 15:39:35 +0200  /dev/tty2
```

The command `faillog` is also used to set limits for failed logins: **faillog -m 3** sets the limit to three failed logins for all users. To prevent root from being locked out, make sure there is no limit for root: **faillog -u root -m 0** (the sequence of options is relevant: `faillog -m 0 -u root` removes the limit for *all* users, not just for root).

To grant access again to a user who had more failures than the limit, enter **faillog -r user**.

Summary

Objective	Summary
1. Monitor a SUSE Linux Enterprise Server 10 System	<p>After installation you may have questions similar to the following;</p> <ul style="list-style-type: none">■ Did the system boot normally?■ What is the kernel version?■ What services are running?■ What is the load on the system? <p>In this objective, you were introduced to tools that help you gather the information needed to answer these questions, like dmesg, hwinfo, siga, sitar, uptime, uname, and others. Files in /proc and its subdirectories are also a source of valuable information.</p>
2. Use System Logging Services	<p>In a Linux system, there are many logs that track various aspects of system operation. Many services log their activities to their own log files, and the level of detail can be set on a per-service basis. In addition, system logs in /var/log/ track system-level events.</p> <p>logrotate is the utility to archive log files.</p>
3. Monitor Login Activity	<p>In addition to log files, several programs exist to specifically monitor login activity, like who, w, last, and faillog</p>

CNI USE ONLY-1 HARDCOPY PERMITTED

SECTION 7 Manage System Initialization

In this section you learn how the SUSE Linux system boots and how to manage that process by setting runlevels, kernel parameters, boot loader options, and other system configurations.

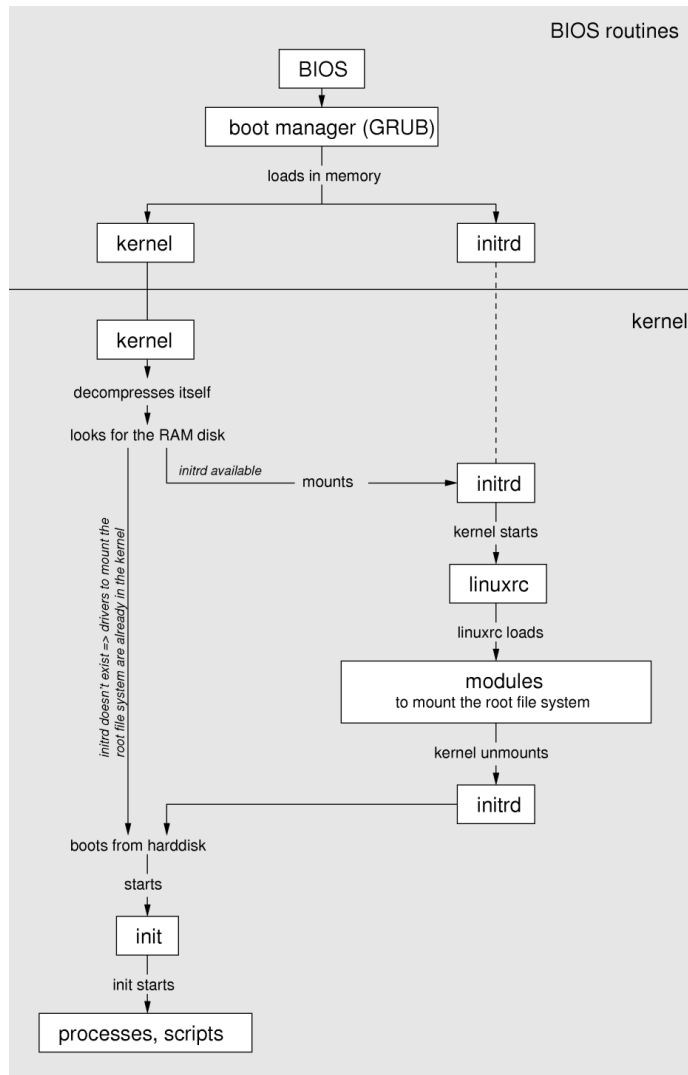
Objectives

1. Describe the Linux Load Procedure
2. GRUB (Grand Unified Bootloader)
3. Manage Runlevels

Objective 1 Describe the Linux Load Procedure

The following represents the basic steps of booting a computer with a Linux system installed:

Figure 7-1



The following describe the process:

- BIOS and Boot Manager
- Kernel
- initramfs (Initial RAM File System)
- init

BIOS and Boot Manager

Tasks performed by the BIOS (Basic Input Output System) include performing a power-on self test, conducting the initial detection and setup of hardware, and accessing bootable devices (such as a CD or hard drive).

If the bootable device is a hard drive, BIOS also reads the MBR (master boot record). Using the code in the MBR, the BIOS starts the boot manager.

The *boot manager* (such as GRUB) loads the kernel and the *initrd* to memory and starts the kernel.

Kernel

The kernel (`/boot/vmlinuz`, which is a link to `/boot/vmlinuz-kernelversion`) uncompresses itself and then organizes and takes control of the continued booting of the system.

The kernel checks and sets the console (the BIOS registers of graphics cards and the screen output format), reads BIOS settings, and initializes basic hardware interfaces.

Next, the drivers, which are part of the kernel, probe existing hardware and initialize it accordingly.

The kernel controls the entire system, managing hardware access and allocating CPU time and memory to programs.

initramfs (Initial RAM File System)

initramfs is a 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. **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.

Former SUSE Linux versions used an initial RAM disk, **initrd**, instead. Despite the fact that the format changed, the file name is still **/boot/initrd**. **/boot/initrd** is a link to **/boot/initrd-*kernelversion***, the file that holds the gzipped cpio archive.

The kernel starts the program **init** contained in the **initramfs**. It is a shell script that, amongst other things, loads the kernel modules needed to mount the actual root file system, mounts the root file system and then finally starts **/sbin/init** from the root file system.

To look at the script **init** in **initramfs**, unpack the cpio archive:

```
da10:~ # mkdir /tmp/initramfs
da10:~ # cd /tmp/initramfs/
da10:/tmp/initramfs # gunzip -c /boot/initrd-2.6.16.14-6-smp | cpio -i
12765 blocks
da10:/tmp/initramfs # ls
bin bootsplash dev etc init lib proc root sbin sys tmp
da10:/tmp/initramfs # less init
```

The initramfs is created with the proper modules included, for instance those needed to access the file system, during installation. The modules to include are listed in the variable **INITRD_MODULES=** in **/etc/sysconfig/kernel**. If additional or different modules are needed, for instance due to a hardware change, you would edit the list of modules, and then rebuild the initramfs. The command is the same as the one to build an initrd, **mkinitrd**:

```
da10:~ # mkinitrd
Root device:      /dev/sda2 (mounted on / as reiserfs)
Module list:      piix aic7xxx sym53c8xx processor thermal
fan reiserfs edd (xennet xenblk)

Kernel image:     /boot/vmlinuz-2.6.16.14-6-smp
Initrd image:     /boot/initrd-2.6.16.14-6-smp
Shared libs:      lib/ld-2.4.so lib/libacl.so.1.1.0
lib/libattr.so.1.1.0 lib/libc-2.4.so lib/libdl-2.4.so
lib/libhistory.so.5.1 lib/libncurses.so.5.5
lib/libpthread-2.4.so lib/libreadline.so.5.1
lib/librt-2.4.so lib/libuuid.so.1.2

Driver modules:   ide-core ide-disk scsi_mod sd_mod piix
scsi_transport_spi aic7xxx sym53c8xx processor thermal fan
edd
Filesystem modules: reiserfs
Including:         initramfs fsck.reiserfs
Bootsplash:      SuSE-SLES (1024x768)
12765 blocks
```



The manual page for **mkinitrd** lists the parameters that can be passed to the **init** program in the **initramfs** via the kernel command line.

init

After checking the partitions and mounting the root file system, the program **init** located in **initramfs** starts **/sbin/init**, which boots the system with all its programs and configurations.

The init process is always assigned a process ID number of 1, and relies on the **/etc/inittab** file for configuration information on how to run the initialization process.

Once the init process starts, it begins by accessing the `/etc/init.d/boot` script. The `/etc/init.d/boot` script controls the start of services such as initializing disk quotas and mounting local file systems.

After the boot script has been completed, init starts the `/etc/init.d/rc` script which uses configured runlevels to start services and daemons.

Each runlevel has its own set of services that are initiated. For example, runlevel 5 includes the X Window components that run the Linux desktop.



For additional details on init, see “Manage Runlevels” on 7-22.

Objective 2 **GRUB (Grand Unified Bootloader)**

To manage GRUB, the Grand Unified Bootloader, you need to know the following:

- What a Boot Manager Is
- Boot Managers in SUSE Linux
- Start the GRUB Shell
- Modify the GRUB Configuration File
- Configure GRUB with YaST
- Boot a System Directly into a Shell

What a Boot Manager Is

To boot a system, you need a program that can load the respective operating system into memory. This program, called the *boot loader*, loads the operating system kernel, which then loads the system.

After running the Power-On Self Test (POST), the PC BIOS searches various media configured in the BIOS for a boot loader. If it finds one, it turns control of the boot process over to the boot loader.

The boot loader then locates the operating system files on the hard drive and starts the operating system.

A *boot manager* is not only a boot loader, but it can handle several operating systems. If there is more than one operating system present, the boot manager presents a menu allowing you to select a specific operating system to be loaded.

Linux boot managers can be used to load Linux or other operating systems, such as Microsoft Windows.

GRUB is designed with the following 2-stage architecture:

- **Stage 1.** The first stage of a boot loader is usually installed in the master boot record (MBR) of the hard disk (first stage boot loader).

As the space in the MBR is limited to 446 bytes, this program code merely contains the information for loading the next stage.

Stage 1 can be installed in the MBR, in the boot sectors of partitions, or on a floppy disk.

- **Stage 2.** This stage usually contains the actual boot loader. The files of the second stage boot loader are located in the directory `/boot/`.

Boot Managers in SUSE Linux

SUSE Linux Enterprise Server provides 2 boot managers for the Linux environment: GRUB (GRand Unified Bootloader) and LILO (LIinux LOader).

To understand something about these boot managers, you need to know the following:

- GRUB Boot Manager
- LILO Boot Manager
- Map Files, GRUB, and LILO

GRUB Boot Manager

GRUB is the standard boot manager in SUSE Linux Enterprise Server. The following are some special features of GRUB:

- **File system support.** Stage 2 includes file system drivers for ReiserFS, ext2, ext3, Minix, JFS, XFS, FAT, and FFS (BSD). For this reason, the boot manager can access files through filenames even before the operating system is loaded.

This feature is useful when the boot manager configuration is faulty and you have to search for and load the kernel.

- **Interactive control.** GRUB has its own shell that enables interactive control of the boot manager.

LILO Boot Manager

Because LILO is not the default boot manager of SUSE Linux Enterprise Server, it is only covered briefly in this objective.

The LILO configuration file is `/etc/lilo.conf`. Its structure is similar to that of the GRUB configuration file.



When you modify the configuration file `/etc/lilo.conf`, you need to enter the command **lilo** for the changes to be applied.

You also need to use the command `lilo` when moving the kernel or the `initrd` on your hard disk.

Map Files, GRUB, and LILO

The main obstacle for booting an operating system is that the kernel is usually a file within a file system on a partition on a disk. These concepts are unknown to the BIOS. To circumvent this, maps and map files were introduced.

These maps simply note the physical block numbers on the disk that comprise the logical files. When such a map is processed, the BIOS loads all the physical blocks in sequence as noted in the map, building the logical file in memory.

In contrast to LILO, which relies entirely on maps, GRUB tries to become independent from the fixed maps at an early stage. GRUB achieves this by means of the file system code, which enables access to files by using the path specification instead of the block numbers.



More information on GRUB and LILO can be found in the respective manual and info pages and in `/usr/share/doc/packages/grub` and `/usr/share/doc/packages/lilo`.

Start the GRUB Shell

As GRUB has its own shell, you can boot the system manually if the Linux system does not start due to an error in the boot manager.

There are two ways to start the GRUB shell:

- Start the GRUB Shell in the Running System
- Start the GRUB Shell at the Boot Prompt

Start the GRUB Shell in the Running System

To start the GRUB shell during operation, enter the command `grub` as root. The following appears:

```
GNU GRUB  version 0.94  (640K lower / 3072K upper memory)

[ Minimal BASH-like line editing is supported.  For the first word, TAB
  lists possible command completions.  Anywhere else TAB lists the
  possible completions of a device/filename. ]

grub>
```

As in a bash shell, you can complete GRUB shell commands with the Tab key. To find out which partition contains the kernel, enter the command **find**, as in the following:

```
grub> find /boot/vmlinuz
(hd0,2)

grub>
```

In this example, the kernel (/boot/vmlinuz) is located in the third partition of the first hard disk (hd0,2).

Close the GRUB shell by entering **quit**.

Start the GRUB Shell at the Boot Prompt

Start the GRUB shell at the boot prompt by doing the following:

1. From the graphical boot selection menu, press **Esc**.
A text-based menu appears.
2. Start the GRUB shell by typing **c** (US keyboard layout).

Modify the GRUB Configuration File

Configure GRUB by editing the file **/boot/grub/menu.lst**. The following is the general structure of the file:

- First, there are general options:
 - **color white/blue black/light-gray**. Colors of the boot manager menu.
 - **default 0**. The first entry (numbering from 0) is the default boot entry that starts automatically if no other entry is selected with the keyboard.
 - **timeout 8**. The default boot entry is started automatically after 8 seconds.

- **gfxmenu (hd0,0)/boot/message.** This defines where the graphical menu is stored.
- The general options are followed by options for the various operating systems that can be booted with the GRUB.
 - **title *title*.** Each entry for an operating system begins with title.
 - **root (hd0,0).** The following entries are relative to this hard disk partition given in the syntax of GRUB, in this example the first partition on the first hard disk. With this entry it is not necessary to specify the partition on each of the following entries like kernel.

Note the following regarding the designations for hard disks and partitions:

GRUB does not distinguish between IDE and SCSI hard disks. The hard disk that is recognized by the BIOS as the first hard disk is designated as hd0, the second hard disk as hd1, and so on.

The first partition on the first hard disk is called hd0,0, the second partition hd0,1, and so on.

- **kernel /boot/vmlinuz.** This entry describes the kernel location, relative to the partition specified above. It is followed by kernel parameters, like root=/dev/hda1, vga=normal, etc.
- **initrd /boot/initrd.** This entry sets the location of the initial ramdisk (initramfs in SLES 10), relative to root (hd0,0) specified above. The initrd contains hardware drivers that are needed before the kernel can access the hard disk (such as a driver for the IDE or SCSI controller).

The following is an example of the configuration file
/boot/grub/menu.lst:

```
# Modified by YaST2. Last modification on Mon May 15 08:38:29 UTC 2006

color white/blue black/light-gray
default 0
timeout 8
gfxmenu (hd0,1)/boot/message

###Don't change this comment - YaST2 identifier: Original name: linux###
title SUSE Linux Enterprise Server 10
    root (hd0,1)
    kernel /boot/vmlinuz root=/dev/sda2 vga=0x317    resume=/dev/sda1
splash=silent showopts
    initrd /boot/initrd

###Don't change this comment - YaST2 identifier: Original name: floppy###
title Floppy
    chainloader (fd0)+1

###Don't change this comment - YaST2 identifier: Original name:
failsafe###
title Failsafe -- SUSE Linux Enterprise Server 10
    root (hd0,1)
    kernel /boot/vmlinuz root=/dev/sda2 vga=normal showopts ide=nodma
apm=off acpi=off noresume nosmp noapic maxcpus=0 edd=off 3
    initrd /boot/initrd
```

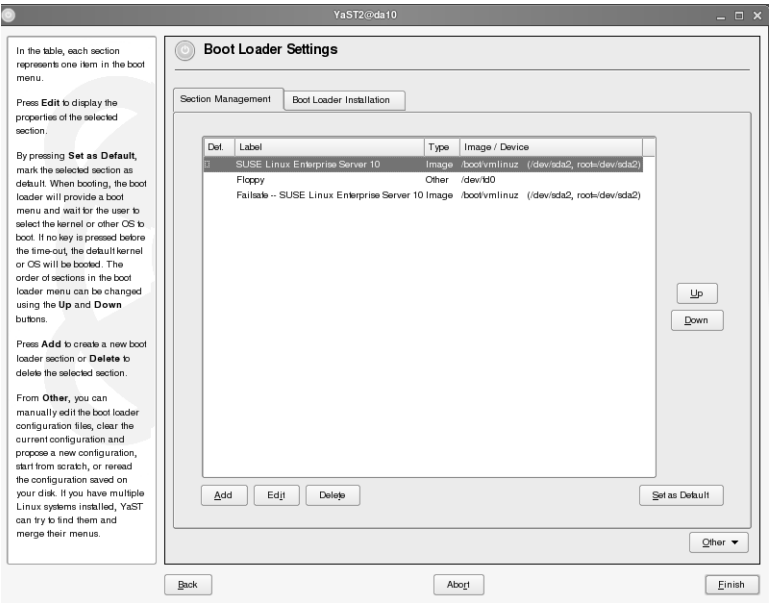
Configure GRUB with YaST

While you can use YaST (bootloader Configuration module) to simplify the configuration of the boot loader, you should not experiment with this module unless you understand the concepts behind it.

To start the YaST Boot Loader module, start YaST, enter the root password, then select **System > Boot Loader**, or start the Boot Loader module directly from a terminal window by entering as root **yast2 bootloader**.

The following appears:

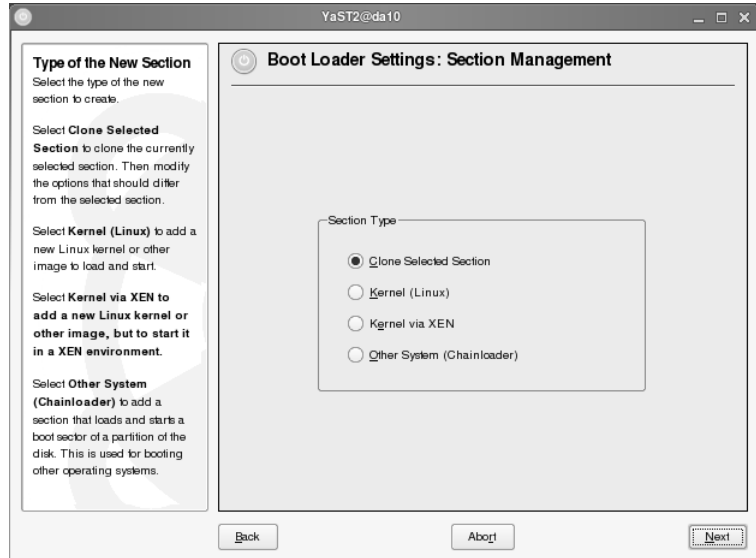
Figure 7-2



When the Section Management tab is selected, you see the current GRUB settings for your system. There is a **Def** (Default) column that indicates which entry is selected as the default when booting the system.

When you select Add, you are offered four choices:

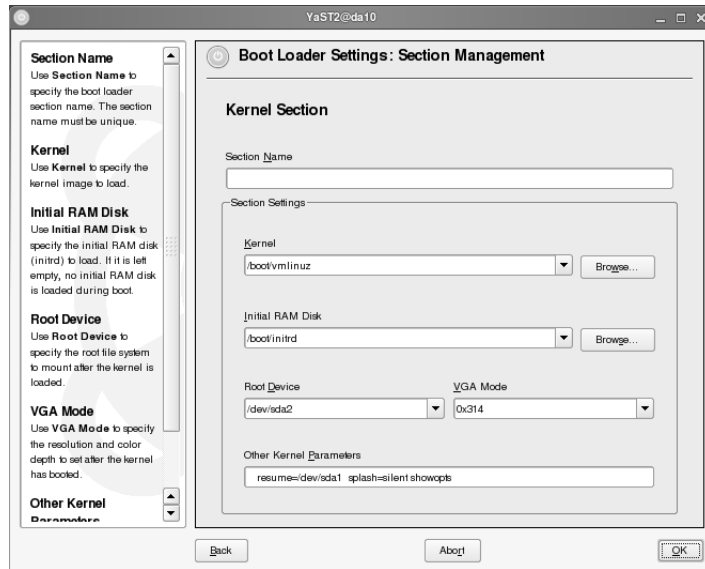
Figure 7-3



Each of the four Section Types is explained in the help text on the left.

When you select **Clone Selected Section** and click **Next**, the dialog is filled with the values from the selected section. With the two following options, the dialog is the same, but the lines are empty:

Figure 7-4



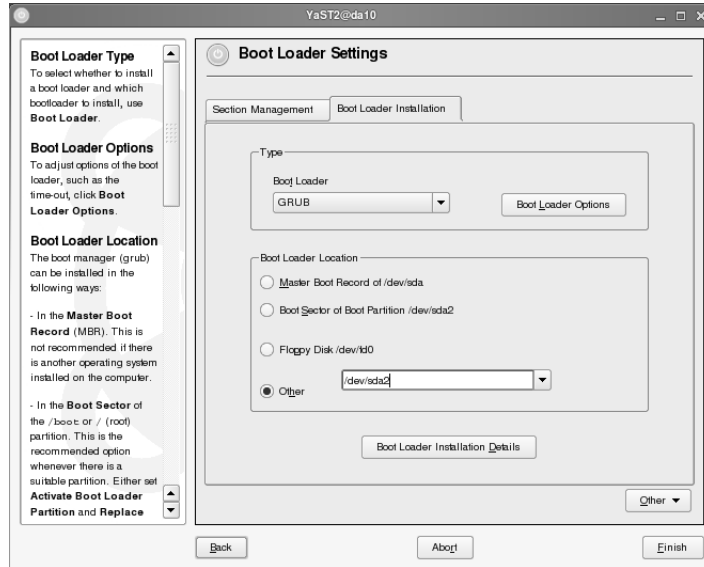
The dialog for the last choice, Other System (Chainloader), offers a line for a section name and a device from where to load another boot loader.

When you select **Edit** in the Boot Loader Settings dialog (Figure 7-2), the same dialogs opens up, where you can change the existing settings.

To delete an entry, select it and then click on **Delete**.

When you select the Boot Loader Installation tab, you see the following dialog:

Figure 7-5



- **Boot Loader Type.** You can use this option to switch between GRUB and LILO. A dialog lets you specify the way this change should be performed.
- **Boot Loader Location.** You can use this option to define whether to install the boot loader in the MBR, in the boot sector of the boot partition (if available), or on a floppy disk.
Use Others to specify a different location.
- **Boot Loader Installation Details.** This offers specialized configuration options, like activating a certain partition or changing the order of disks to correspond with the sequence in the BIOS.
- **Other.** When you select this, a drop-down menu with the following additional choices opens up:

- ❑ **Edit Configuration Files.** Display and edit the configuration files (/boot/grub/device.map, /boot/grub/menu.lst, or /boot/grub.conf).
 - ❑ **Propose New Configuration.** This option generates a new configuration suggestion. Older Linux versions or other operating systems found on other partitions are included in the boot menu, enabling you to boot Linux or its old boot loader. The latter takes you to a second boot menu.
 - ❑ **Start from Scratch.** This option lets you create the entire configuration from scratch. No suggestions are generated.
 - ❑ **Reread Configuration from Disk.** If you already performed some changes and are not satisfied with the result, you can reload your current configuration with this option.
 - ❑ **Propose and Merge with Existing GRUB Menus.** If another operating system and an older Linux version are installed in other partitions, the menu is generated from an entry for the new SUSE Linux, an entry for the other system, and all entries of the old boot loader menu.

This procedure might take some time and is only available with GRUB.
 - ❑ **Restore MBR from Hard Disk.** The MBR saved on the hard disk is restored.
1. When you finish configuring the boot loader, save the configuration changes by selecting **Finish**.

Boot a System Directly into a Shell

The boot screen of the GRUB boot loader lets you pass parameters that modify the Linux kernel before the kernel is actually loaded.

At the bottom of the GRUB boot screen is a Boot Options field. To add a boot option, select an operating system and type the additional boot option in the Boot Options field.

One way to access a system that is not booting anymore is to set a different program for the init process. Normally, the Linux kernel tries to find a program with the name `init` and starts this program as the first process. All other processes are then started by `init`.

With the boot parameter `init=new_init_program`, you can change the first program loaded by the kernel. For example, by entering the boot parameter **`init=/bin/bash`**, the system is started directly into a bash shell. You are directly logged in as root without being asked for a password.

You can use this bash file to access the file system and to fix a misconfiguration.



The file systems are mounted as read-only after booting into a shell. To change configuration files, you need to remount the file system with the following command:

```
mount -o remount,rw,sync -t filesystem_type device_name mount_point
```

Entering **`exec /sbin/init`** at the bash prompt replaces the shell by the `init` program and continues the boot process until the default runlevel is reached.

If you want to prevent access to the machine as described above, you can change the boot configuration to require a password before the kernel command line can be edited.

In the file `/boot/grub/menu.lst`, the line

`password secret`

within the general options makes sure that the choices defined further below in the file (title `SUSE SLES 10`, etc.) can only be selected in unmodified form. The use of additional kernel parameters requires the password “*secret*”.

As the graphical boot menu could be used to circumvent the password feature, it is automatically disabled.

GRUB can also handle MD5-encrypted passwords that are generated as follows:

```
da10:~ # grub-md5-crypt
Password:
Retype password:
$1$FtTeK1$qaV.tOrzbg3EYAgVfNup40
```

This string can be copied to the file `/boot/grub/menu.lst`, with the following syntax:

password --md5 \$1\$FtTeK1\$qaV.tOrzbg3EYAgVfNup40

The parameter **lock** within a title section can be used to force the password query before these title entries can be selected.

```
title Floppy
    lock
    chainloader (fd0)+1
```

Selecting Floppy in the boot menu is now only possible after entering the password.

The parameter `password` can also be used in individual title entries to define a special password for those title entries.

Please note that the password feature only moderately enhances security, as it does not prevent booting the computer from another medium, like the SLES 10 rescue system, and accessing the files on the hard disk.



If you want to decide for each service (postfix, sshd, etc.) whether to start it or not during booting, use the parameter “confirm” at the bootprompt.

Exercise 7-1 *Manage the Boot Loader*

In this exercise, you practice booting into a shell and modifying /boot/grub/menu.lst.

You will find this exercise in the workbook.

(End of Exercise)

Objective 3 **Manage Runlevels**

Managing runlevels is an essential part of Linux system administration. In this objective, you learn what runlevels are, the role of the program `init`, and how to configure and change runlevels:

- The `init` Program and Linux Runlevels
- `init` Scripts and Runlevel Directories
- Change the Runlevel

The `init` Program and Linux Runlevels

- The `init` Program
- The Runlevels
- `init` Configuration File (`/etc/inittab`)

The `init` Program

The system is initialized by `/sbin/init`, which is started by the kernel as the first process of the system.

This process, or one of its child processes, starts all additional processes. In addition, because **`init`** is the last process running, it ensures that all other processes are correctly ended. This means that `init` controls the entire booting up and shutting down of the system.

Because of this position of priority, signal 9 (`SIGKILL`), with which all processes can normally be ended, has no effect on `init`.

The main configuration file of `init` is **`/etc/inittab`**. Various scripts are started by `init`, depending on entries in this file. All these scripts are located in the directory `/etc/init.d/`.

Part of the configuration in `/etc/inittab` is the runlevel the system uses after booting.

The Runlevels

In Linux, various runlevels define the state of the system. The following are the available runlevels:

Table 7-1

Command	Description
0	Halt
S	Used to boot into single-user mode (US keyboard layout)
1	Single-user mode
2	Multiuser mode without network server services
3	Multiuser mode with network
4	Not used
5	Multiuser mode with network and display manager
6	Reboot

The command `runlevel` displays the runlevel you are currently in (second number) and the previous runlevel (first number), as in the following:

```
da10:~ # runlevel
N 5
da10:~ #
```

init Configuration File (/etc/inittab)

To understand the contents of the file `/etc/inittab`, you need to know the following:

- `inittab` Syntax
- `inittab` Standard Entries

inittab Syntax

The following is the syntax of each line in the file `/etc/inittab`:

```
id:rl:action:process
```

The following describes the parameters:

- ***id***. A unique name for the entry in `/etc/inittab`. It can be up to four characters long.
- ***rl***. Refers to one or more runlevels in which this entry should be evaluated.
- ***action***. Describes what init is to do.
- ***process***. Is the process connected to this entry.

inittab Standard Entries

The first entry in the file `/etc/inittab` contains the following parameters:

```
id:5:initdefault:
```

The parameter `initdefault` signals to the `init` process which level it should bring the system to. The standard default runlevel is normally 3 or 5.

The next entry in `/etc/inittab` looks like this:

```
si:bootwait:/etc/init.d/boot
```

The parameter `bootwait` indicates to carry out this command while booting and wait until it has finished.

The next few entries describe the actions for runlevels 0 to 6:

```
10:0:wait:/etc/init.d/rc 0
11:1:wait:/etc/init.d/rc 1
12:2:wait:/etc/init.d/rc 2
13:3:wait:/etc/init.d/rc 3
#14:4:wait:/etc/init.d/rc 4
15:5:wait:/etc/init.d/rc 5
16:6:wait:/etc/init.d/rc 6

ls:S:wait:/etc/init.d/rc S
~~:S:respawn:/sbin/sulogin
```

The parameter `wait` means that when the system changes to the indicated level, the appropriate command is carried out and `init` waits until it has been completed. The parameter also means that further entries for the level are only performed after this process is completed.

The single user mode `S` is a special case, as it works even if the file `/etc/inittab` is missing. In such a case, enter `S` at the boot prompt when the computer starts. The command `sulogin` is started, which allows only the system administrator to log in. The parameter `respawn` indicates to `init` to wait for the end of the process and to then restart it.

`/etc/inittab` also defines the `Ctrl+Alt+Del` key combination for restarting:

```
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now
```

The action `ctrlaltdel` is carried out by the `init` process only if these keys are pressed. If you do not want to allow this action, comment out (`#`) or remove the line.

The final large block of entries describes in which runlevels `getty` processes (login processes) are started:

```
1:2345:respawn:/sbin/mingetty --noclear tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

The `getty` processes provide the login prompt and in return expect a user name as input. They are started in runlevels 2, 3, and 5.



Runlevel 4 in the above example is ignored because the line that defines the actions for the runlevel is commented out earlier in the file (**#14:4:wait:/etc/init.d/rc 4**).

If a session ends, the processes are started again by `init`. If a line is disabled here, no further login is possible at the corresponding virtual console.



You should take great care when making changes to the file `/etc/inittab`. If the file is corrupted, the system will no longer boot correctly.

If an error does occur, first try entering `S` at the kernel command line in the GRUB boot menu. If this does not work, it is still possible to boot the system. Enter **`init=/bin/bash`** at the kernel command line in the GRUB boot menu.

In this way, the `init` process is replaced by a shell (so `inittab` is not read) and you can then repair the system manually.

When you changed `/etc/inittab`, use **`init q`** to have `init` reload its configuration.

init Scripts and Runlevel Directories

`/etc/inittab` defines the runlevel the system uses after booting is complete. The services that need to be started in a certain runlevel are not defined in `/etc/inittab` itself. These are configured by symbolic links in directories `/etc/init.d/rcx.d/` which point to scripts in `/etc/init.d/`. To be able to manage runlevels, you need to understand the following:

- `init` Scripts
- Runlevel Symbolic Links
- How `init` Determines which Services to Start and Stop
- Activate and Deactivate Services for a Runlevel
- Activate and Deactivate Services for a Runlevel with YaST

init Scripts

The directory `/etc/init.d/` contains shell scripts that are used to perform certain tasks at boot up and start and stop services in the running system. The following shows some of the files in `/etc/init.d/`:

```
da10:~ # ls -al /etc/init.d/
total 635
drwxr-xr-x 11 root root 3336 May 24 13:40 .
drwxr-xr-x 77 root root 6712 May 25 13:19 ..
-rw-r--r-- 1 root root 1393 May 24 13:40 .depend.boot
-rw-r--r-- 1 root root 3465 May 24 13:40 .depend.start
-rw-r--r-- 1 root root 3002 May 24 13:40 .depend.stop
-rw-r--r-- 1 root root 482 Aug 25 2004 Makefile
-rw-r--r-- 1 root root 7827 May 10 18:17 README
-rwxr-xr-x 1 root root 1257 May 8 20:09 SuSEfirewall2_init
-rwxr-xr-x 1 root root 1650 May 8 20:09 SuSEfirewall2_setup
-rwxr-xr-x 1 root root 2696 May 8 20:29 aaeventd
-rwxr--r-- 1 root root 5729 May 8 20:15 acpid
-rwxr-xr-x 1 root root 5265 May 8 21:01 alsasound
-rwxr-xr-x 1 root root 3689 May 9 14:49 atd
-rwxr-xr-x 1 root root 6691 May 9 15:03 auditd
-rwxr--r-- 1 root root 9234 May 9 15:01 autofs
-rwxr-xr-x 1 root root 2967 Mar 14 13:40 autoyast
-rwxr-xr-x 1 root root 7072 Apr 20 15:02 boot
-rwxr-xr-x 1 root root 2792 May 8 20:29 boot.apparmor
...
```

The files `.depend.{boot,start,stop}` are created by `insserv` and contain dependencies that are used to determine the proper sequence for starting services.

The shell scripts can be called up in the following ways:

- Directly by `init` when you boot the system, when the system is shut down, or when you stop the system with **Ctrl+Alt+Del**. Examples for these scripts are `/etc/init.d/boot` or `/etc/init.d/rc`.

- Indirectly by init when you change the runlevel. In this case, it is the script `/etc/init.d/rc` that calls the necessary scripts in the correct order and with the correct parameter during the runlevel change.
- Directly by *`/etc/init.d/script parameter`*.

You can also enter *`rcscript parameter`* if corresponding links are set in `/sbin/` or `/usr/sbin/`.

The following parameters may be used:

Table 7-2

Parameter	Description
start	Starts a service that is not running.
restart	Stops a running service and restarts it.
stop	Stops a running service.
reload	Rereads the configuration of the service without stopping and restarting the service itself.
force-reload	Reloads the configuration if the service supports this. Otherwise, it does the same thing as restart.
status	Displays the current status of the service.

When a script is called without parameters, a message informs you about the possible parameters.

Some of the more important scripts stored in `/etc/init.d/` are:

- **boot.** This script is started directly by init when the system starts. It is run once and once only. It evaluates the directory `/etc/init.d/boot.d/` and starts all the scripts linked by filenames with an “S” at the beginning of their names (see “Runlevel Symbolic Links” on 7-30).

These scripts perform, for instance, the following tasks:

- Check the file systems

- ❑ Set up of LVM
- ❑ Delete unnecessary files in `/var/lock/`
- ❑ Set the system time
- ❑ Configure PnP hardware with the `isapnp` tools
- **boot.local.** This script includes additional commands to execute at boot before changing into a runlevel. You can add your own system extensions to this script.
- **halt.** This script is run if runlevel 0 or 6 is entered. It is called up either with the command `halt` (the system is completely shut down) or with the command `reboot` (the system is shut down and then rebooted).
- **rc.** This script is responsible for the correct change from one runlevel to another. It runs the stop scripts for the current runlevel, and then it runs the start scripts for the new one.
- **service.** Each service (like `cron`, `apache2`, `cups`) comes with a script allowing you to start and stop the service, to reload its configuration, or to view its status. To create your own scripts, you can use the file `/etc/init.d/skeleton` as a template.

Runlevel Symbolic Links

To enter a certain runlevel, `init` calls the script `/etc/init.d/rc` with the runlevel as parameter. This script examines the respective runlevel directory **`/etc/init.d/rcx.d/`** and starts and stops services depending on the links in this directory.

For each runlevel, there is a corresponding subdirectory in `/etc/init.d/`. For runlevel 1 it is `/etc/init.d/rc1.d/`, for runlevel 2 it is `/etc/init.d/rc2.d/`, and so on.

When you view the files in a directory such as `/etc/init.d/rc3.d/`, you see two kinds of files—those that start with a “K” and those that start with an “S”:

```
da10:~ # ls /etc/init.d/rc3.d/
K10cron          K17network       S07auditd
K10smbfs          K20haldaemon     S07portmap
K11nscd           K21acpid         S07splash_early
K11postfix        K21dbus          S08nfs
K12alsasound      K21fbset         S08nfsboot
K12boot.apparmor  K21irq_balancer  S10alsasound
K12cups           K21random        S10boot.apparmor
K12microcode      K21resmgr        S10cups
K12powersaved     S01acpid         S10kbd
K12splash         S01dbus          S10microcode
K12sshd           S01fbset         S10powersaved
K14nfs            S01irq_balancer  S10splash
K14nfsboot        S01random        S10sshd
K15auditd         S01resmgr        S11nscd
K15portmap        S02haldaemon     S11postfix
K15splash_early   S05network       S12cron
K16novell-zmd     S06novell-zmd    S12smbfs
K16slpd           S06slpd
K16syslog         S06syslog
```

The first letter is always followed by 2 digits and the name of a service. Whether a service is started in a specific runlevel depends on whether there are **Sxxservice** and **Kxxservice** files in the `/etc/init.d/rcx.d/` directory.

Entering **ls -l** in an `/etc/init.d/rcx.d/` directory indicates that these files are actually symbolic links pointing to service scripts in `/etc/init.d/` (as in the following):

```
da10:~ # ls -l /etc/init.d/rc3.d/
total 0
lrwxrwxrwx 1 root root 7 May 15 10:32 K10cron -> ../cron
lrwxrwxrwx 1 root root 8 May 15 10:48 K10smbfs -> ../smbfs
lrwxrwxrwx 1 root root 7 May 15 10:32 K11nscd -> ../nscd
lrwxrwxrwx 1 root root 10 May 15 10:32 K11postfix -> ../postfix
lrwxrwxrwx 1 root root 12 May 15 10:26 K12alsasound -> ../alsasound
...
lrwxrwxrwx 1 root root 7 May 15 10:31 S10sshd -> ../sshd
lrwxrwxrwx 1 root root 7 May 15 10:32 S11nscd -> ../nscd
lrwxrwxrwx 1 root root 10 May 15 10:32 S11postfix -> ../postfix
lrwxrwxrwx 1 root root 7 May 15 10:32 S12cron -> ../cron
lrwxrwxrwx 1 root root 8 May 15 10:48 S12smbfs -> ../smbfs
```

By using symbolic links in subdirectories only the version in `/etc/init.d/` needs to be modified in case of necessary changes to the script.

Usually, two links within a runlevel directory point to the same script. For example, if you enter

ls -l *network

in the `/etc/init.d/rc3.d/` directory, you see that two network links both point to the script `/etc/init.d/network`:

```
da10:~ # ls -l /etc/init.d/rc3.d/*network
lrwxrwxrwx 1 root root 10 May 15 10:23 /etc/init.d/rc3.d/K17network ->
../network
lrwxrwxrwx 1 root root 10 May 15 10:23 /etc/init.d/rc3.d/S05network ->
../network
```



Sometimes **Kxx** links are referred to as *kill scripts*, while **Sxx** links are referred to as *start scripts*. In fact, there are no separate scripts for starting and stopping services, but the script is either called with the parameter `stop` or with the parameter `start`.

How init Determines which Services to Start and Stop

You already know that a service is started with the parameter `start`, and stopped with the parameter `stop`. The same parameters are also used when changing from one runlevel to another.

When the runlevel is changed, `init` calls the script `rc` with the new runlevel as parameter, like **`/etc/init.d/rc 3`**. The script `/etc/init.d/rc` examines the directories `/etc/init.d/rccurrentrl.d/` and `/etc/init.d/rcnewrl.d/` and determines what to do.

Let's say we change from our current runlevel 5 to the new runlevel 3. There are three possibilities:

- There is a **Kxx** link for a certain service in `/etc/init.d/rc5.d/` and there is an **Sxx** link in `/etc/init.d/rc3.d/` for the same service.

In this case, the service is neither started nor stopped; the corresponding script in `/etc/init.d/` is not called at all.

- There is a **Kxx** link for a certain service in `/etc/init.d/rc5.d/` and there is no corresponding **Sxx** link in `/etc/init.d/rc3.d/`.

In this case, the script in `/etc/init.d/service` is called with the parameter `stop` and the service is stopped.

- There is an **Sxx** link in `/etc/init.d/rc3.d/` and there is no corresponding **Kxx** link for the service in `/etc/init.d/rc5.d/`.

In this case, the script in `/etc/init.d/service` is called with the parameter `start` and the service is started.

The number after the K or S determines the sequence in which the scripts are called.

Therefore script `K10cron` is called before script `K20haldaemon`, which means that `cron` is shut down before `haldaemon`.

Script `S05network` is called before `S11postfix`, which means that the service `network` starts before `postfix`. This is important if `postfix` depends on a running service `network`.

For example the following happens when you change from runlevel 3 to runlevel 5:

1. You tell init to change to a different runlevel by entering (as root) **init 5**.
2. init checks its configuration file (/etc/inittab) and determines it should start /etc/init.d/rc with the new runlevel (**5**) as a parameter.
3. rc calls the stop scripts (**Kxx**) of the current runlevel for those services for which there is no start script (**Sxx**) in the new runlevel.
4. The start scripts in the new runlevel for those services for which there was no kill script in the old runlevel are launched.

When changing to the same runlevel as the current runlevel, init only checks /etc/inittab for changes and starts the appropriate steps (such as starting a getty on another interface).

Activate and Deactivate Services for a Runlevel

Services are activated or deactivated in a runlevel by adding or removing the respective K**service and S**service links in the runlevel directories /etc/init.d/rcx.d/.

Although you could create symbolic links in the runlevel subdirectories yourself to modify services, an easier way is to edit the header of a script and then call **insserv**.

The INIT INFO block at the beginning of the script for a service describes in which runlevel the service should start or stop and what services should run as a prerequisite:

```
### BEGIN INIT INFO
# Provides:          syslog
# Required-Start:    network
# Should-Start:      earlysyslog
# Required-Stop:     network
# Default-Start:     2 3 5
# Default-Stop:
# Description:       Start the system logging daemons
### END INIT INFO
```

The INIT INFO block is used by the program `insserv` to determine in which runlevel subdirectories links need to be placed and what numbers need to be put after K and S.



For details on the program `insserv`, enter **man 8 insserv**.

The entry `Default-Start` determines in which runlevel directories links are to be placed. The entry `Required-Start` determines which services have to be started before the one being considered.

After editing the INIT INFO block, enter **`insserv -d service`** (default) to create the needed links and renumber the existing ones as needed.

To remove all links for a service (disabling the service), stop the service (if it is running) by entering **`/etc/init.d/service stop`**, and then enter **`insserv -r service`** (remove).

Within the INIT INFO block, the use of certain variables is possible. These are explained and defined in `/etc/insserv.conf`.

A tool with similar functionality is **chkconfig**. It can be used to disable or enable services and also to list which services are enabled in which runlevel. The following gives a brief overview on how to use **chkconfig**:

```
da10:~ # chkconfig cron
cron on
da10:~ # chkconfig cron -l
cron          0:off  1:off  2:on   3:on   4:off  5:on   6:off
da10:~ # chkconfig cron off
da10:~ # chkconfig cron -l
cron          0:off  1:off  2:off  3:off  4:off  5:off  6:off
da10:~ # chkconfig cron on
da10:~ # chkconfig -l
Makefile      0:off  1:off  2:off  3:off  4:off  5:off  6:off
SuSEfirewall2_init 0:off  1:off  2:off  3:off  4:off  5:off  6:off
SuSEfirewall2_setup 0:off  1:off  2:off  3:off  4:off  5:off  6:off
aaeventd      0:off  1:off  2:off  3:off  4:off  5:off  6:off
acpid         0:off  1:off  2:on   3:on   4:off  5:on   6:off
alsasound     0:off  1:off  2:on   3:on   4:off  5:on   6:off
atd           0:off  1:off  2:off  3:off  4:off  5:off  6:off
auditd        0:off  1:off  2:off  3:on   4:off  5:on   6:off
...
```

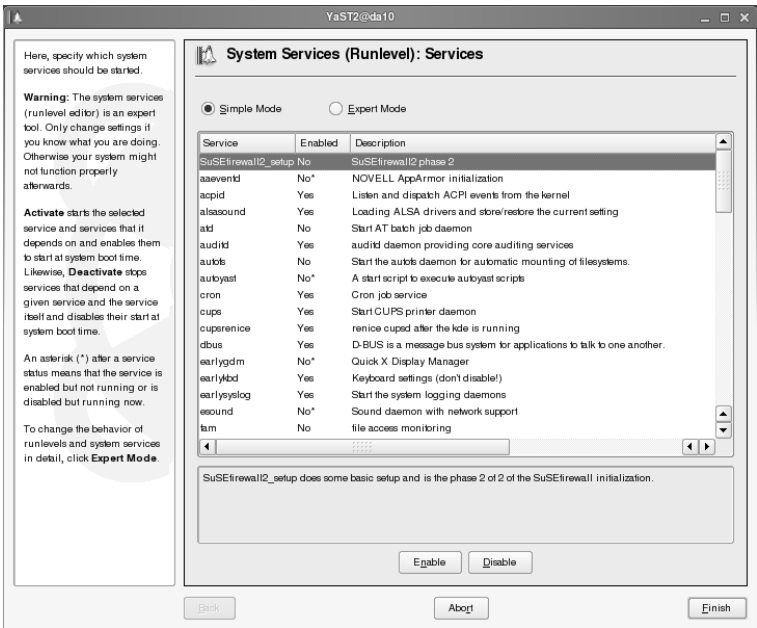
You can also use the YaST runlevel editor to set these links. We recommend that you either use **insserv/chkconfig** or **YaST**. Switching between methods can lead to errors.

Activate and Deactivate Services for a Runlevel with YaST

To configure runlevels with YaST, start the YaST Runlevel Editor module by starting **YaST** and then selecting **System > System Services (Runlevel)**, or open a terminal window and as root enter **yast2 runlevel**.

The following appears:

Figure 7-6



From this dialog, you can select from the following modes:

- **Simple Mode.** This mode displays a list of all available services and the current status of each service.

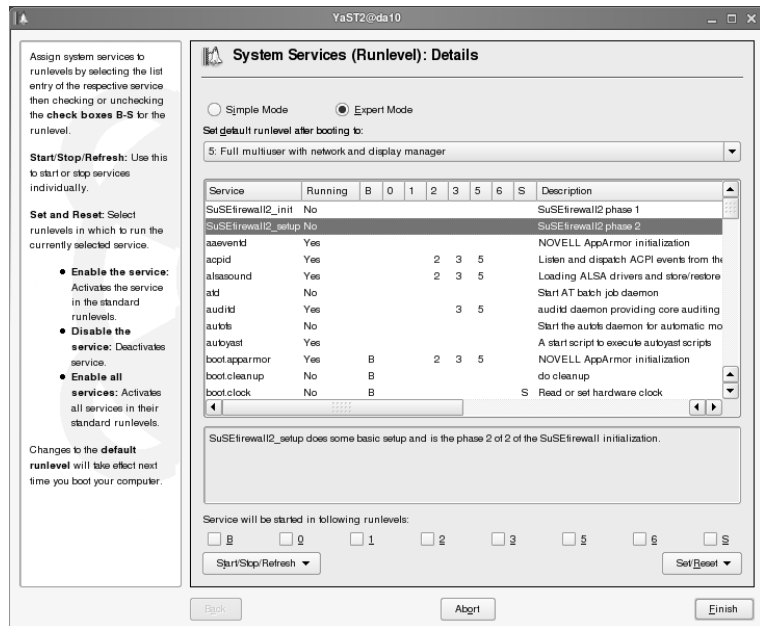
You can select a service, and then select **Enable** or **Disable**.

Selecting **Enable** starts the service (and services it depends on) and enables them to start at system boot time. Selecting **Disable** stops dependent services and the service itself and disables their start at system boot time.

- **Expert Mode.** This mode gives you control over the runlevels in which a service is started or stopped and lets you change the default runlevel.

Expert Mode looks like the following:

Figure 7-7



In this mode, the dialog displays the current default runlevel at the top. You can select a new default runlevel from the drop-down menu.

Normally, the default runlevel of a SUSE Linux system is runlevel 5 (full multiuser with network and graphical environment). A suitable alternative might be runlevel 3 (full multiuser with network). Runlevel 4 is initially undefined to allow creation of a custom runlevel.

Changes to the default runlevel take effect the next time you boot your computer.

To configure a service, select a service from the list, then from the options below the list, select the **runlevels** you want associated with the service.

The list includes the services and daemons available, indicates whether they are currently enabled on your system, and lists the runlevels currently assigned.

If you want a service activated after editing the runlevels, from the drop-down list select **Start now**, **Stop now**, or **Refresh status**.

You can use Refresh status to check the current status (if this has not been done automatically).

From the Set/Reset drop-down list, select one of the following:

- **Enable the service:** activates the service in the standard runlevels.
- **Disable the service:** deactivates the service.
- **Enable all services:** Activates all services in their standard runlevels.

When you finish configuring the runlevels, save the configuration by selecting **Finish**.

Remember that faulty runlevel settings can make a system unusable. Before applying your changes, make absolutely sure you know the impact of the changes.

Change the Runlevel

When starting the system, you can choose a runlevel different from the default runlevel defined in /etc/inittab. The runlevel can also be changed in the running system.

To change the runlevel, you need to understand:

- Change the Runlevel at Boot

■ Manage Runlevels from the Command Line

Change the Runlevel at Boot

The standard runlevel is 3 or 5, as defined in the file `/etc/inittab` by the entry `initdefault`. However, it is also possible to boot to another runlevel by specifying the runlevel on the kernel command line of GRUB.

Any parameters that are not evaluated by the kernel itself are passed to `init` as parameters by the kernel. The desired runlevel is simply appended to the boot options already specified in GRUB (in the file `/boot/grub/menu.lst`), as in the following example:

```
root=/dev/hda1 vga=0x317 resume=/dev/hda2 splash=silent showopts 1
```

As root partition `/dev/hda1` is transmitted to the kernel, various parameters including the framebuffer are set, and the system boots to runlevel 1 (single user mode for administration).

Manage Runlevels from the Command Line

You can change to another runlevel once the system is running by using the command **init**. For example, you can change to runlevel 1 from a command line by entering **init 1**.

In the same way, you can change back to the standard runlevel where all programs needed for operation are run and where individual users can log in to the system.

For example, you can return to a full GUI desktop and network interface (runlevel 5) by entering **init 5**.



If the partition /usr of a system is mounted through NFS, you should not use runlevel 2 because NFS file systems are not available in this runlevel.

Like most modern operating systems, Linux reacts sensitively to being switched off without warning. If this happens, the file systems need to be checked and corrected before the system can be used again.

For this reason, the system should always be shut down properly. With the appropriate hardware, Linux can also switch off the machine as the last stage of shutting down.

You stop the system by entering **init 0**; you restart the system by entering **init 6**. The commands **halt** and **poweroff** are equivalent to **init 0**; the command **reboot** is equivalent to **init 6**.

The command **shutdown** shuts down the system after the specified time (+m: minutes from now; hh:mm: time in hours:minutes, when Linux should shut down; now: system is stopped immediately). The option -h causes a system halt, if you use the option -r instead the system is rebooted. Without options, it changes to runlevel 1 (single user mode).

The command **shutdown** controls the shutdown of the system in a special way, compared with the other stop commands. The command informs all users that the system will be shut down and does not allow other users to log in before it shuts down.

The command **shutdown** can also be supplied with a warning message, such as the following:

```
shutdown +5 The new hard drive has arrived
```

If a shutdown planned for a later time should not be carried out after all, you can revoke the shutdown by entering **shutdown -c**.

Exercise 7-2 Manage Runlevels

In this exercise, you practice configuring runlevels.

You will find this exercise in the workbook.

(End of Exercise)

Summary

Objective	Summary
1. Describe the Linux Load Procedure	<p>In this objective, you learned the following about the basic steps of booting a computer with a Linux system:</p> <ul style="list-style-type: none">■ BIOS and Boot Manager■ Kernel■ initramfs (Initial RAM File System)■ init
2. GRUB (Grand Unified Bootloader)	<p>The default boot manager in SLES 10 is GRUB. It is responsible for loading the operating system.</p> <p>Its configuration file is <code>/boot/grub/menu.lst</code>.</p> <p>The GRUB shell allows, amongst other things, to search for and view the content of files before the operating system is running.</p>

Objective	Summary
3. Manage Runlevels	<p>The initialization of the system is done by /sbin/init, which is started by the kernel as the first process of the system.</p> <p>The central configuration file of init is /etc/inittab.</p> <p>Various scripts are started by init. These scripts are located in the directory /etc/init.d/.</p> <p>In Linux, various runlevels define the state of the system.</p> <p>The system administrator can change to another runlevel with the command init.</p> <p>The command runlevel displays the previous and the current runlevel.</p>

SECTION 8 **Manage Software for SUSE Linux Enterprise Server**

In this section, you learn how to manage software packages on your SUSE Linux Enterprise server with RPM Package Manager (RPM) and YaST. You are also introduced to dynamic software libraries.

Objectives

1. Manage RPM Software Packages
2. Verify and Update Software Library Access

Objective 1 **Manage RPM Software Packages**

While there are several software package formats available for Linux, the format used most commonly in SUSE Linux installations is the RPM Package Manager (RPM) format.

Installing software in the RPM format can be done with YaST or by using the command `rpm`. YaST ensures the automatic resolution of dependencies, while `rpm` only controls them (resolution must be performed manually).

To manage installation of RPM software packages, you need to know the following:

- RPM Components and Features
- RPM Basics
- Manage Software Packages with `rpm`

RPM Components and Features

RPM Package Manager (or RPM) is a package management system primarily intended for Linux. RPM installs, updates, uninstalls, verifies software, and allows various queries about the installed software.

The following are the basic components of RPM:

- **RPM Package Manager.** The utility that handles installing and uninstalling RPM packages.
- **RPM database.** The RPM database works in the background of the package manager and contains a list of all information on all installed RPM packages.

The database keeps track of all files that are changed and created when a user installs a program. This helps the package manager to easily remove the same files that were originally installed.

- **RPM package.** RPM lets you take software source code and package it into source and binary packages for end users. These are called RPM packages or RPM archives.
- **Package label.** Every RPM package includes a package label that contains information such as the software name, version, and the package release number.

This information helps the package manager track the installed versions of software to make it easier to manage software installations on a Linux computer.

Some of the advantages of using RPM package manager and RPM packages include the following:

- Provides a consistent method for users to install programs in Linux.
- Makes it easier to uninstall programs (because of the RPM database).
- Original source archives (such as tar.gz, .tar.bz2) are included and easy to verify.
- You can use RPM tools to enable software installations using noninteractive scripts.
- You can use RPM tools to verify that the software installed correctly.
- RPM can track dependent software, preventing deinstallation of packages needed by other packages. It also informs the administrator if required software is missing when she tries to install a software package.
- RPM allows for all packaged software to use public-key technology to digitally sign the software.

RPM Basics

To manage software packages with RPM, you need to understand the following:

- RPM Package File Naming Convention
- RPM Configuration File
- RPM Database

RPM Package File Naming Convention

RPM package files use the following naming format:

software_name-software_version-release_number.architecture.rpm, for instance `apache2-2.2.0-21.i586.rpm`

The following describes each component of the naming format:

- ***software_name***. This is normally the name of the software being installed.
- ***software_version***. This is the version number of the software in the RPM package and is normally a number.
- ***release_number***. This is the number of times the package has been rebuilt using the same version of the software.
- ***architecture***. This indicates the architecture the package was built under (such as `i586`, `i686`, `ppc`, ...) or the type of package content.

For example, if the package has an `i586` architecture, you can install it on 32-bit Intel-compatible machines that are Pentium class or higher.

If the package has a `noarch` extension, it does not include any binary code.

- **rpm.** RPM archives normally have the extension `.rpm`. The distribution also includes source packages, called source RPMs, which have the filename extension `.src.rpm` (`.spm` or `.srpm` are also possible).

Note: Source packages are not included in the RPM database and thus are not recorded.

RPM Configuration File

The global RPM configuration file of the command `rpm` is `/usr/lib/rpm/rpmrc`. However, when the `rpm` command is updated, all changes to this file are lost.

To prevent this from happening, write the changes to the file `/etc/rpmrc` (for the system configuration) or to file `~/rpmrc` (for the user configuration).

RPM Database

The files of the RPM database are stored 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 by entering **`rpm --rebuilddb`**. Before doing this, make a backup of the old database.

The cron script `suse.de-backup-rpmdb` stored in `/etc/cron.daily/` checks daily to see if there are any changes. If so, a copy of the database is made (compressed with `gzip`) and stored in `/var/adm/backup/rpmdb/`.

The number of copies is controlled by the variable `MAX_RPMDDB_BACKUPS` (default is 5) in `/etc/sysconfig/backup`.

The size of a single backup is approximately 5 MB for 1 GB in `/usr`.

CNI USE ONLY-1 HARDCOPY PERMITTED

Manage Software Packages with rpm

You can use the command `rpm` to manage software packages. This includes querying the RPM database for detailed information about the installed software.

The command provides the following modes for managing software packages:

- Installing, uninstalling, or updating software packages
- Querying the RPM database or individual RPM archives
- Checking the integrity of packages
- Rebuilding the RPM database

You can use the command **`rpmbuild`** to build installable RPM packages from pristine sources. `rpmbuild` is not covered in this course.

RPM packages contain program, configuration, and documentation files to install, and certain meta information used during installation by RPM to configure the software package. This same information is stored in the RPM database after installation for documentation purposes.

To manage software packages with RPM, you need to know how to do the following:

- Verify Package Authenticity
- Install, Update, and Uninstall Packages
- Query the RPM Database and RPM Archives
- Update Software with Patch RPMs
- YaST as a Frontend to RPM

Verify Package Authenticity

All SUSE Linux RPM packages are signed with the following GnuPG key:

```
da10:~ # gpg --list-keys -v --fingerprint "build@suse.de"
pub   1024D/9C800ACA 2000-10-19 [expires: 2008-06-21]
       Key fingerprint = 79C1 79B2 E1C8 20C1 890F  9994 A84E DAE8 9C80 0ACA
uid           SuSE Package Signing Key <build@suse.de>
sub   2048g/8495160C 2000-10-19 [expires: 2008-06-21]
```

You can enter the command **rpm --checksig *package_name*** (such as **rpm --checksig apache2-2.2.0-10.i586.rpm**) to verify the signature of an RPM package. This lets you determine whether the package originated from SUSE or from another trustworthy facility.

Verifying the package signature is especially recommended for update packages from the Internet.

The SUSE public package signature key is stored in the directories `/root/.gnupg/` and `/usr/lib/rpm/gnupg/`. Storing the key in `/usr/lib/rpm/gnupg/` lets normal users verify the signature of RPM packages.

Install, Update, and Uninstall Packages

To manage RPM software packages, you need to know how to do the following:

- Install an RPM Package
- Update an RPM Package
- Uninstall an RPM Package

Install an RPM Package

For most RPM packages, you use the following command to install the software:

rpm -i *package_name.rpm*

When you install an RPM package, the executable programs, documentation files, configuration files, and start scripts are copied to the appropriate directories in the file system.

During installation, the RPM database ensures that no conflicts arise (such as a file belonging to more than 1 package). The package is installed only if its dependencies are fulfilled and there are no conflicts with other packages.

If dependencies are not fulfilled, RPM lists those packages that need to be installed to meet dependency requirements. Packages that conflict with the packages to be installed are also listed.

You could use other options to ignore these errors (like `--nodeps` to ignore dependencies, or `--force` to overwrite existing files), but this is only for experts. If you force the installation despite dependency requirements not being met, the installed software most likely will not work properly.

With the option `-v` (verbose) more information is displayed, and the option `-h` (hash) produces a progress bar consisting of # signs during package installation.



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 (such as the most recent GNOME packages).

Such packages can be identified by the name extension `-devel`, such as the packages `alsa-devel` or `gimp-devel`.

Update an RPM Package

You can use the options `-U` (or `--upgrade`) and `-F` (or `--freshen`) to update a package by using the following syntax:

`rpm -F package_name.rpm`

This command removes the files of the old version and immediately installs the new files. If there is no previous version installed, the package is not installed.

If there is an old version installed, the option `-U` does the same as `-F`, however if there is no previous version installed, `-U` installs the new version.

RPM updates configuration files carefully using the following guidelines:

- 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). It then 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. Be sure to delete all `.rpmorig` and `.rpmsave` files afterwards to avoid problems with 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.

- A set of .rpmnew files are created if the configuration file already exists and if the noreplace label was specified in the file controlling the package creation (the so-called .spec-file).

This is used to not overwrite certain configuration files (such as /etc/httpd/httpd.conf) to ensure continued operation.

.rpmnew does not disclose any information as to whether the system administrator has made any changes to the configuration file.

The script /etc/init.d/rpmconfigcheck searches for such files and writes a list of these files to /var/adm/rpmconfigcheck.



The option -U is *not* equivalent to uninstalling with the -e option and installing with the -i option. Use -U whenever possible for updating packages.

Uninstall an RPM Package

To uninstall (remove) an RPM package, enter the following:

rpm -e package_name

When you uninstall a package, all files except modified configuration files are removed from the system with the help of the RPM database. This ensures a clean uninstall.

RPM will delete the package only if this does not break dependencies. If other packages depend on the package you want to delete, these are listed in the error message.

You could force deletion of the package with the parameter --nodeps, however this is not advisable as the dependent software will most likely not work anymore.

Query the RPM Database and RPM Archives

With the **-q** option, you can query the RPM database of installed packages and, by adding the option **-p**, inspect RPM archives that are not yet installed.

The following are the most commonly-used RPM query options:

Table 8-1

Option	Results
-a	List all installed packages
-i	List package information
-l	Display a file list
-f <i>file</i>	Find out to which package <i>file</i> belongs (the full path must be specified with <i>file</i>)
-d	List only documentation files (implies -l)
-c	List only configuration files (implies -l)
--dump	Display a 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	List the capabilities the package requires
--scripts	List installation scripts (preinstall, postinstall, uninstall)
--changelog	Displays a detailed list of information (updates, configuration, modifications, etc.) about a specific package.

For example, entering the command **rpm -qi wget** displays the following information:

```
da10:~ # rpm -qi wget
Name       : wget                      Relocations: (not relocatable)
Version    : 1.10.2                  Vendor: SUSE LINUX Products GmbH,
Nuernberg, Germany
Release    : 15                      Build Date: Mon May  8 21:16:26
2006
Install Date: Mon May 15 10:28:23 2006  Build Host: nicolai.suse.de
Group      : Productivity/Networking/Web/Utilities  Source RPM:
wget-1.10.2-15.src.rpm
Size       : 1532429                  License: GPL
Signature  : DSA/SHA1, Mon May  8 21:20:08 2006, Key ID a84edae89c800aca
Packager   : http://bugs.opensuse.org
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.

Authors:
-----
    Hrvoje Niksic <hniksic@srce.hr>
Distribution: SUSE Linux Enterprise 10 (i586)
```

The option **-f** only works if you specify the complete filename with a full path. You can enter several filenames, as in the following:

```
da10:~ # rpm -qf /bin/rpm /usr/bin/wget
rpm-4.4.2-43
wget-1.10.2-15
```

This returns information for both **/bin/rpm** and **/usr/bin/wget**.

With the help of the RPM database, you can perform verification checks with the option **-V**, or **--verify**. If any files in a package have been changed since installation they are displayed.

RPM uses the following character symbols to provide hints about the changes:

Table 8-2

Character	Description
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 displayed. The following is an example for changes to `/etc/wgetrc` (`wget`):

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

Update Software with Patch RPMs

To guarantee the operational security of a system, you should update packages frequently by installing patched packages.

You could update the complete package, or you could use a patch RPM suitable to the installed RPM package. The patch RPM has the advantage of being smaller, reducing the download time.

When planning an update, you need to consider the following (using the package **procmail** as an example):

- Is the patch RPM suitable for my system?

To check this, first query the installed version of the package:

```
da10:~ # rpm -q procmail
procmail-3.22-42
```

The output indicates the currently installed version of procmail. Then check if the patch RPM is suitable for this version of procmail:

```
da10:~ # rpm -qp --basedon \
procmail-3.22-42.4.i586.patch.rpm
procmail = 3.22-42
procmail = 3.22-42.2
```

The output indicates that the patch is suitable for 2 different versions of procmail. 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 option `-P` lets you select special patch features.

You can display the list of files with the following command:

```
da10:~ # rpm -qPpl procmail-3.22-42.4.i586.patch.rpm
/usr/bin/formail
/usr/bin/lockfile
/usr/bin/procmail
```

If the patch is already installed, use the following command:

```
da10:~ # rpm -qPl procmail
/usr/bin/formail
/usr/bin/lockfile
/usr/bin/procmail
```

- 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?

You can display a list of all patches installed in the system with the command **rpm -qPa**. If only the patch for procmail is installed in a new system, the following list appears:

```
da10:~ # rpm -qPa
procmail-3.22-42.4
```

If at a later date you want to know which package version was originally installed, you can query the RPM database.

For procmail, this information can be displayed with the following command:

```
da10:~ # rpm -q --basedon procmail
procmail = 3.22-42
```



For additional details about the patch feature of RPM, enter **man rpm** or **man rpmbuild**.

YaST as a Frontend to RPM

One of the major functions of YaST is software installation. If you know the name of a software package, the option **-i** (install) is very useful.

The following is an example:

```
da10:~ # yast -i ethereal
```

This example installs the `ethereal` package plus any software package that is needed by `ethereal` from the installation media. The advantage of using `yast -i` is that any dependencies are automatically resolved.

You can also install any rpm package like that:

```
da10:~ # yast -i apache2-2.2.0-10.i586.rpm
```

However, dependencies are not resolved in this case.

Exercise 8-1 Manage Software with RPM

In this exercise, you practice gathering information on installed software and installing software packages.

You will find this exercise in the workbook.

(End of Exercise)

Objective 2 **Verify and Update Software Library Access**

In addition to checking for software package dependencies, you might also need to verify that the system is configured properly to access dynamic libraries an application uses.

Normally this is handled by the software installation, but occasionally you might need to verify software library access after installation.

For example, if an application that has been installed fails to start, try starting it from a terminal window. If the application reports that a library could not be found, then you might need to verify access to the dynamic libraries.

To verify the libraries needed for an application, you need to know the following:

- Software Library Basics
- View Shared Library Dependencies (ldd)
- Modify the Software Library Configuration File (/etc/ld.so.conf)
- Update the Library Cache (/etc/ld.so.cache)

Software Library Basics

To understand the role of software libraries in SUSE Linux, you need to know the following:

- Dynamic Software Libraries
- Static Software Libraries
- Library Naming Syntax

Dynamic Software Libraries

In a Linux environment, most programs share some code through the use of shared libraries. This provides advantages from a development and a system management standpoint.

For developers, it means their programs include only the code that is unique to the program itself, sharing functions that other programs have in common with it.

This reduces the size of the program executable, thus reducing the amount of disk space required for the application (an advantage for system administrators).

Unlike some other operating systems, a Linux system locates its dynamic libraries through a configuration file that points to the locations, eliminating confusion about which version of which dynamic library is used by each piece of software.



Developers still have the ability to link everything into their executable. This can be important if the program will be used on a system that might not include all of the necessary libraries, such as an emergency rescue disk or minimal Linux installation.

Static Software Libraries

In contrast to dynamic program linking, you can link the needed libraries statically when a program is compiled.

Although static linking increases the program size, it provides independence from libraries at runtime, and is especially useful for system maintenance purposes.

An example of a program with statically linked libraries is **sash**. **sash** (stand-alone shell) is useful for recovering from certain types of system failures. It was created in order to cope with the problem of missing shared libraries or important executables. Built in commands include `-mount`, `-mknod`, `-kill`, `-ln`, `-gzip`, `-gunzip`, and others.

Library Naming Syntax

Library filenames normally use the following syntax:

`libname.so.version`

The letters “so” indicate a shared dynamic library; the letter “a” (as in `/usr/lib/libc.a`) is used for static libraries. The version indicates a major version number of the library (such as 1, 2, or 6).

For example, the library used for the ncurses screen library (version 4.2) might be named:

`libncurses.so.4.2`

View Shared Library Dependencies (ldd)

You can view the shared libraries required by a specific program or shared library by using the command `ldd`.

The following is the syntax of the command:

`ldd option filename`

For example, if you enter **ldd -v /opt/gnome/bin/nautilus**, information similar to the following appears:

```
da10:~ # ldd -v /opt/gnome/bin/nautilus
linux-gate.so.1 => (0xffffe000)
libnautilus-private.so.2 =>
    /opt/gnome/lib/libnautilus-private.so.2 (0xb7ece000)
libbeagle.so.0 => /usr/lib/libbeagle.so.0 (0xb7ebd000)
libnautilus-extension.so.1 =>
    /opt/gnome/lib/libnautilus-extension.so.1 (0xb7eb6000)
libeel-2.so.2 => /opt/gnome/lib/libeel-2.so.2 (0xb7e2a000)
...
Version information:
/opt/gnome/bin/nautilus:
    libm.so.6 (GLIBC_2.0) => /lib/libm.so.6
    libpthread.so.0 (GLIBC_2.0) => /lib/libpthread.so.0
    libc.so.6 (GLIBC_2.2) => /lib/libc.so.6
    libc.so.6 (GLIBC_2.3.4) => /lib/libc.so.6
    libc.so.6 (GLIBC_2.1) => /lib/libc.so.6
    libc.so.6 (GLIBC_2.0) => /lib/libc.so.6
/opt/gnome/lib/libnautilus-private.so.2:
    libm.so.6 (GLIBC_2.0) => /lib/libm.so.6
    libc.so.6 (GLIBC_2.2) => /lib/libc.so.6
    libc.so.6 (GLIBC_2.1.3) => /lib/libc.so.6
    libc.so.6 (GLIBC_2.3.4) => /lib/libc.so.6
...
```

By using the command **ldd**, you can also find out if all required libraries are installed on a system for a specific program. The output of **ldd** would indicate “**not found**” for the missing library.

For additional information on the command **ldd**, from a terminal window, enter **man ldd**.

Modify the Software Library Configuration File (/etc/ld.so.conf)

The file `/etc/ld.so.conf` contains a list of paths the Linux system uses to search for libraries, as in the following:

```
da10:~ # cat /etc/ld.so.conf
/usr/X11R6/lib/Xaw3d
/usr/X11R6/lib
/usr/i486-linux-libc5/lib=libc5
/usr/i386-suse-linux/lib
/usr/local/lib
/opt/kde3/lib
/opt/gnome/lib
include /etc/ld.so.conf.d/*.conf
```

In order to modify the file `/etc/ld.so.conf`, you need to be authenticated as the root user. The file format for this file is simply a list of system directories containing dynamic libraries.

Typical library directories include the following: `/lib/`, `/usr/lib/`, `/usr/local/lib/`, and `/usr/X11R6/lib/`.

As the directories `/lib/` and `/usr/lib/` are taken into account in all cases, they are not listed in this file. You can enter the command **`/sbin/ldconfig -p`** to list all libraries available in the cache that will be found by the system.

If a library is located in a directory not listed above, you can set the variable **`LD_LIBRARY_PATH=`***path* to make sure that it is loaded:

`export LD_LIBRARY_PATH=`*path*



For a listing of variables that can be used, enter **`man 8 ld.so`**.

Update the Library Cache (/etc/ld.so.cache)

The program **/lib/ld-linux.so.2** (this is a link to **/lib/ld-2.4.so**), referred to as the *runtime linker*, makes sure that the needed libraries are found and loaded when a program is started.

If you modify the **/etc/ld.so.conf** to reflect any new dynamic library paths, you need to enter the command **ldconfig** to update the library cache. If new libraries are installed during operation, you also need to run **ldconfig** manually.

This is the same command used to update the library cache when rebooting the system.

The command sets the required links to the current shared libraries that are either located in directories listed in the file **/etc/ld.so.conf** or in the directories **/usr/lib/** and **/lib/**.

The library cache file is **/etc/ld.so.cache** and is read by the runtime linker. The cache file contains a list of all the system libraries stored in a binary format to speed the location of the libraries on the system.

Running **ldconfig** with the option **-v** displays detailed information about the libraries **ldconfig** has found.

Exercise 8-2 Manage Shared Libraries

In this exercise, you use some common utilities to manage the shared libraries on your SLES 10 server.

You will find this exercise in the workbook.

(End of Exercise)

Summary

Objective	Summary
1. Manage RPM Software Packages	<p>RPM packages are packed in a special binary format. Apart from the executable programs, they also contain information about the configuration of the software package, as well as information about dependencies on other packages (including shared libraries).</p> <p>You can use the command <code>rpm</code> to install software packages (<code>rpm -i</code>, or <code>rpm -U</code>, or <code>rpm -F</code>), uninstall software packages (<code>rpm -e</code>), and query information from the RPM database (<code>rpm -q</code>).</p>
2. Verify and Update Software Library Access	<p>To verify the libraries needed for an application, you learned about the following:</p> <ul style="list-style-type: none">■ Software Library Basics■ View Shared Library Dependencies (<code>ldd</code>)■ Modify the Software Library Configuration File (<code>/etc/ld.so.conf</code>)■ Update the Library Cache (<code>/etc/ld.so.cache</code>)

CNI USE ONLY-1 HARDCOPY PERMITTED

SECTION 9 Manage Backup and Recovery

In this section, you learn how to develop a backup strategy and how to use the backup tools shipped with SUSE Linux Enterprise Server 10.

Objectives

1. Develop a Backup Strategy
2. Backup Files with YaST
3. Create Backups with tar
4. Work with Magnetic Tapes
5. Copy Data with dd
6. Mirror Directories with rsync
7. Automate Data Backups with cron

Introduction

Even the best security measures cannot guarantee that data will never be lost. There is always the possibility that

- A hard disk failure occurs, destroying data on the affected disk.
- Users will delete files by accident.
- A virus will delete important files on a desktop computer.
- A notebook will be lost or destroyed.
- An attacker will delete data on a server.
- Natural influences like thunderstorms will destroy storage systems.

It is very important to ensure that you have a reliable backup of important data.

In this section you learn how to develop a backup strategy and how to use the standard UNIX backup tools tar, rsync, and dd.

You will learn about possible issues during the boot process and how to configure the GRUB boot loader.

Objective 1 **Develop a Backup Strategy**

Backing up data is one of the most important tasks of a system administrator. But before you can actually back up data, you need to develop a backup strategy by doing the following:

- Choose a Backup Method
- Choose the Right Backup Media

Choose a Backup Method

The best possible method of data backup is the *full backup*.

In a full backup, all system data is copied to a backup media once a day. To restore the data, the most current backup media is copied back to the system's hard disk.

The disadvantage of this method is the *backup window*. The backup window is the time frame available to perform backups.

Backups should be performed when the system is not used, to avoid data changes on the disk during the backup. These data changes would lead to inconsistent data on the backup media.

Therefore, a backup is normally performed at night when systems are not needed.

In some cases, especially in larger companies, the backup window might be too small to perform a full backup every day.

This can happen for the following reasons:

- The amount of data to be backed up is so large, it takes too long to copy all data to a backup media during the backup window.
- The affected systems have to be available around the clock, so the backup window is very small.

In most cases, a combination of both reasons prevents you from using a full backup.

To circumvent this problem, you can use a backup method other than full backup. The following are 2 basic backup alternatives:

- Perform an Incremental Backup
- Perform a Differential Backup

Perform an Incremental Backup

In an incremental backup, you normally perform a full backup once a week (such as on the weekend). Then you perform a backup every day that copies only files that have changed since the backup the day before.

For example, if you might perform a full backup on Sunday, while on Monday you just backup the files which have changed since Sunday. On Tuesday you back up the files which have changed since Monday, and so on.

Before performing an incremental backup, you need to understand the following advantage and disadvantage of this method:

- **Advantage.** Because you only back up files that have changed since the last backup, the backup window can be much smaller than the one you need for a daily full backup.
- **Disadvantage.** The recovery time is longer. For example, you have perform a full backup on Sunday and incremental backups on Monday, Tuesday and Wednesday. On Thursday the server crashes and all data is lost.

To restore the server you now need all incremental backups and the full backup since last Sunday. All these backups need to be copied to the server in the correct order.

Perform a Differential Backup

In a differential backup, you perform a full backup once a week, then you perform backups every day to record the files that have changed since the last full backup.

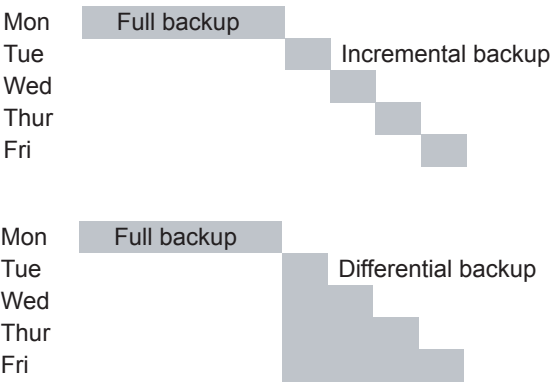
For example, suppose you perform a full backup on Sunday. On Monday you back up the files that have changed since Sunday, on Tuesday you also back up the files that have changed since Sunday, and so on.

Before performing a differential backup, you need to understand the following advantage and disadvantage of the method:

- **Advantage.** To restore data from a differential backup, you need just 2 backup media: the last full backup and the last differential backup. This makes the average time needed to restore a system shorter.
- **Disadvantage.** The amount of data to be backed up grows every day. At the end of the backup cycle, the amount of data might be too large for the available backup window.

The following illustrates the difference between incremental and differential backups:

Figure 9-1



Choose the Right Backup Media

You must choose the suitable backup media for the amount of data to be backed up and the backup method.

Tape drives are used most often because they still have the best price-to-capacity ratio. Normally these are SCSI drives, so that all kinds of tape drives can be accessed in the same way (such as DAT, EXABYTE, and DLT). In addition, tapes can be reused.

Other media for data backup include writable CDs or DVDs, removable hard drives, and magnetic-optical (MO) drives.

Another option are Storage Area Networks (SANs). With a SAN, a storage network is set up to exclusively back up data from different computers on a central backup server. But even a SAN often uses magnetic tapes to store the data.

Backup media should always be stored separately from the backed up systems. This prevents the backups from being lost in case of a fire in the server room. Sensitive backup media should be stored safely offsite.

Objective 2 **Backup Files with YaST**

To back up and restore a file system with YaST on SUSE Linux Enterprise Server, you need to know how to:

- Back Up System Data with YaST
- Restore System Data with YaST

Back Up System Data with YaST

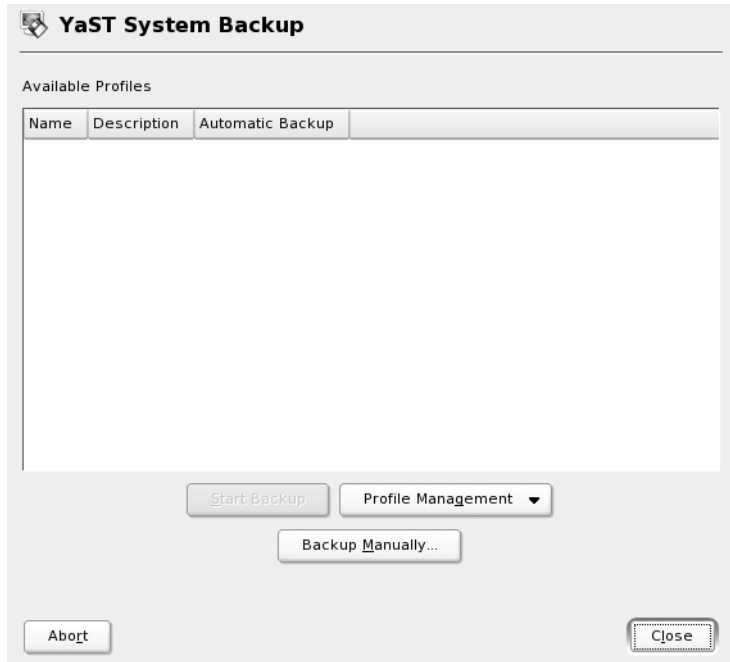
The YaST System Backup module lets you create a backup of your system. The backup does not comprise the entire system, but only saves information about changed packages and copies of critical storage areas and configuration files.

To create a backup with YaST, do the following:

1. From the KDE desktop, start the YaST System Backup module by doing one of the following:
 - Select the **YaST** icon, enter the root *password*, and select **OK**; then select **System > System Backup**.
 - or*
 - Open a terminal window and enter **su -** and the root *password*; then enter **yast2 backup**.

The following appears:

Figure 9-2



This dialog shows the list of currently stored backup profiles. A *backup profile* is used to name a group of different settings, such as name of an archive and how to search for files.

You can have a number of profiles, each with a unique name.

From the Profile Management drop-down list, you can add a new profile (**Add**) based on default values, duplicate an existing profile (**Duplicate**), edit the settings stored in a profile (**Change**), delete a profile (**Delete**), or configure automatic backup settings.

You can also use **Backup Manually** to configure a backup without creating a backup profile.

2. Create a profile by selecting **Profile Management > Add**.
3. Enter a ***name*** for the profile that will be used in the profile list; then select **OK**.

The following appears:

Figure 9-3

4. In the **File Name** field, enter a ***filename*** for the backup file.
You need to enter a full path (absolute path) with the filename (such as **/etc/backup_1**).
5. Save the backup file to a local directory by selecting **Local file**, or save the backup file to a remote server by selecting **Network (NFS)** and entering the remote server and directory.

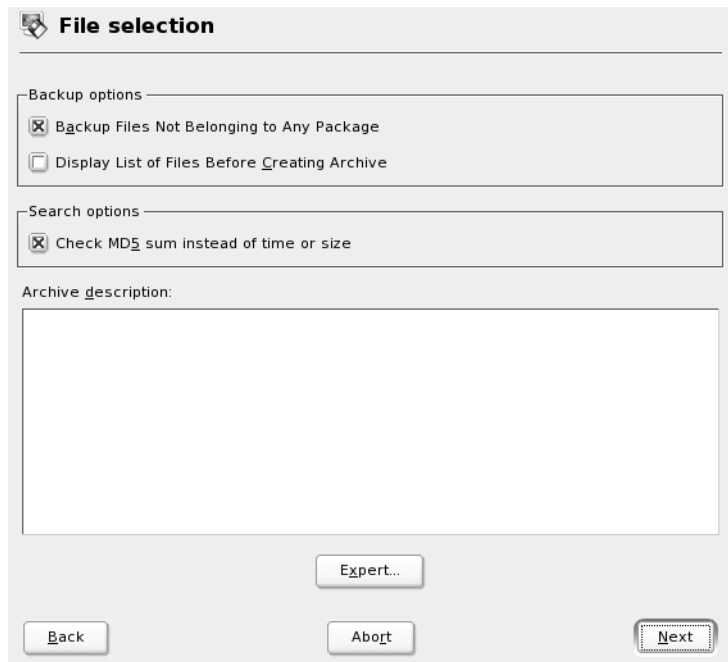
6. Create a backup file that contains the backup data by selecting **Create Backup Archive**, or select **Only Create List of Files Found**.

The Create Backup Archive option lets you select an archive type (such as **tar with tar-gzip**) from a drop-down list, and configure additional options (such as multivolume archive) by selecting **Options**.

7. When you finish configuring the archive settings, continue by selecting **Next**.

The following appears:

Figure 9-4



From this dialog you can select which parts of the system to search and back up.

The archive will contain files from packages that were changed since package installation or upgrade.

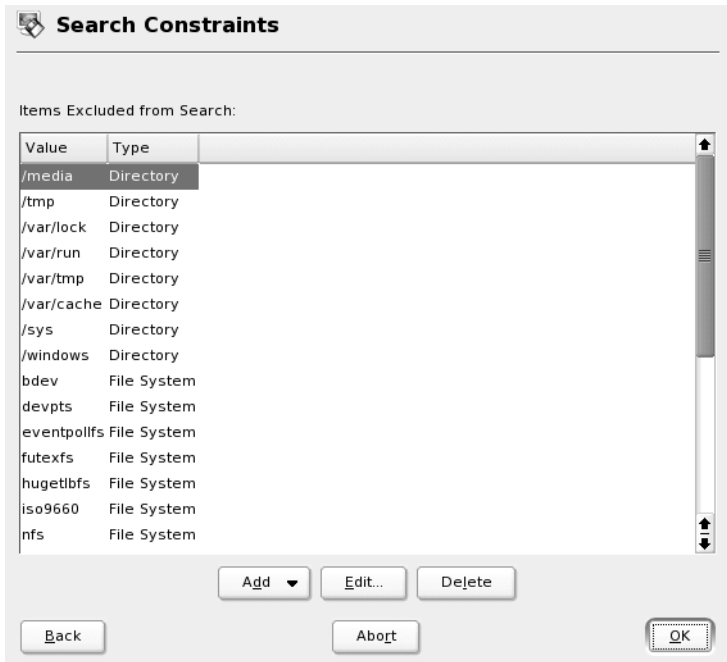
8. Select one or both of the following options:
 - ❑ **Backup Files Not Belonging to Any Package.** Includes these files in the backup.
 - ❑ **Display List of Files Before Creating Archive.** Lets you show and edit a list of files found before creating the backup archive.
9. (Optional) In the **Archive Description** field, enter a *description* of the backup archive.
10. Use MD5 sum checking by selecting **Check MD5 sum instead of time or size.**

You can use MD5 sum to determine if the file was changed. It is more reliable than checking size or modification time, but takes more time.
11. (Optional) Configure advanced options (such as adding the partition table to the backup) by selecting **Expert.**

For most backups, you do not need to change the default Expert options.
12. When you finish configuring, continue by selecting **Next.**

The following appears:

Figure 9-5



This dialog lists all the items you want excluded from the backup, including the following exclusion types:

- ❑ **Directories.** All files located in the specified directories will not be backed up.
- ❑ **File Systems.** You can exclude all files located on a certain type of file system (such as ReiserFS or Ext2). The root directory will always be searched, even if its file system is selected.

File systems that cannot be used on a local disk (such as network file systems) are excluded by default.

- ❑ **Regular expressions.** Any filename that matches any of the regular expressions will not be backed up. Use perl regular expressions. For example, to exclude *.bak files, add the regular expression **\.bak\$**.
- 13. Add an item to the exclusion list by selecting **Add > exclusion type** and entering a *directory*, *file system*, or *expression*; then select **OK**.
- 14. Edit or remove an item from the list by selecting the *item*; then select **Edit** or **Delete**.
- 15. When you finish, continue by selecting **OK**.

You are returned to the YaST System Backup dialog where the new profile appears in the list.
- 16. Start the backup by doing one of the following:
 - ❑ Select the profile; then select **Create Backup**.
 - ❑ Set an automatic backup by selecting **Profile Management > Automatic Backup**.

You can set options such as backup frequency, backup start time, and maximum number of old backups.
- 17. When you finish configuring system backups, select **Close**.

Restore System Data with YaST

You can use the YaST Restore system module to restore a system backup by doing the following:

1. From the KDE desktop, start the YaST Restore system module by doing one of the following:
 - ❑ Select the **YaST** icon, enter the root *password*, and select **OK**; then select **System > System Restoration**.
 - or*
 - ❑ Open a terminal window and enter **su -** and the root *password*; then enter **yast2 restore**.

The following appears:

Figure 9-6

Archive selection

Backup archive

☒ Local file

Archive file name:

☐ Network (NFS)

IP address or name of NFS server:

Archive file name:

☐ Removable device

Device:

Archive file name:

2. Do one of the following:

- ❑ If the backup file is stored locally, select **Local file**; then enter the *archive filename* (include the full path) or locate and enter the file by selecting **Select file**.

or

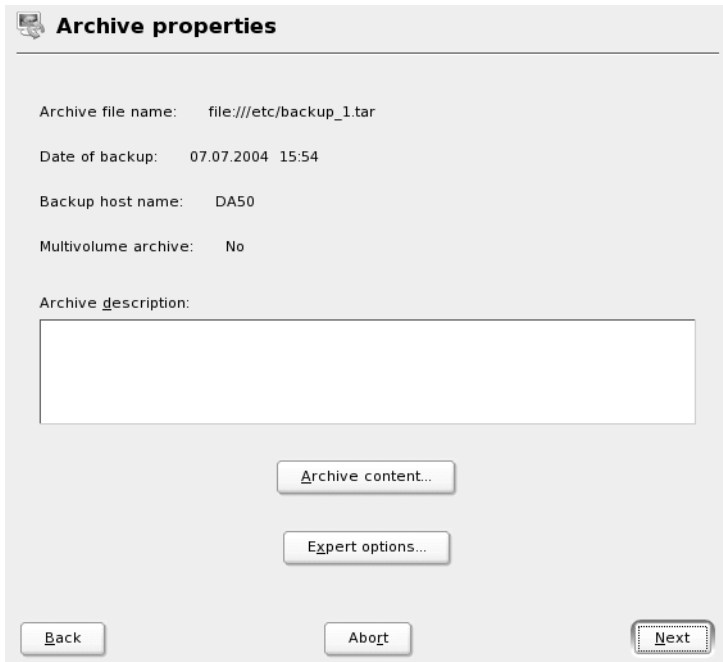
- ❑ If the backup file is stored on a network server, select **Network (NFS)**; then enter the *remote server* and the full path of the *archive backup file*.

or

- ❑ If the backup file is on a removable device (such as a diskette or tape drive), select **Removable device**; then select the *device* from the drop-down list and enter the full path of the *archive backup file* (or use **Select file**).
3. When you finish, continue by selecting **Next**.

YaST reads the contents of the archive file and the following appears:

Figure 9-7



This dialog lists the properties of the archive file.

- 4. View the archive contents by selecting **Archive content**.
- 5. Configure options such as activating the boot loader configuration after restoration and entering the target directory by selecting **Expert Options**.

6. When you finish, continue by selecting **Next**.



If this is a multivolume archive, selecting **Next** displays an Archive properties dialog for each volume.

The following appears:

Figure 9-8



This dialog lets you select which files you want restored from the archive (all are selected by default).

The first column in the list displays the restoration status of the package. It can be **X** (package will be restored), **empty** (package will not be restored), or **P** (package will be restored partially).

The number of selected files that will be restored from the archive is in the second column.

Press **Select Files** to restore a package partially.

7. Do one of the following:
 - ❑ Select all packages in the list by selecting **Select all**.
or
 - ❑ Deselect all packages in the list by selecting **Deselect all**.
or
 - ❑ Restore particular files in a highlighted package by selecting **Select files**; then select or deselect the listed files.
8. (Optional) If the RPM database exists in the archive, restore it by selecting **Restore the RPM database**.
9. When you finish selecting packages, start restoring files by selecting **Accept**.

When the restoration is complete, a summary dialog appears listing the status of the restored files.
10. (Optional) Save the summary to a file by selecting **Save to file**.
11. Close the dialog by selecting **Finish**.

Exercise 9-1 Backup Files with YaST

In this exercise, you learn how to perform a system backup with YaST.

You will find this exercise in the workbook.

(End of Exercise)

Objective 3 **Create Backups with tar**

The tar (tape archiver) tool is the most commonly used application for data backup on Linux systems. It archives files in a special format, either directly on a backup medium (such as magnetic tape or floppy disk), or to an archive file.

The following are tasks you perform when backing up files with tar:

- Create tar Archives
- Unpack tar Archives
- Exclude Files from Backup
- Perform Incremental and Differential Backups
- Use tar Command Line Options

Create tar Archives

The tar format is a container format for files and directory structures. By convention, the extension of the archive files end in .tar.

tar archives can be saved to a file to store them on a file system, or they can be written directly to a backup tape.

Normally the data in the archive files is not compressed, but you can enable compression with additional compression commands. If archive files are compressed (usually with the command gzip), then the extension of the filename is either .tar.gz or .tgz.

The tar command first expects an option, then the name of the archive to be written (or the device file of a tape recorder), and the name of the directory to be backed up. All directories and files under this directory are also saved.

Directories are typically backed up with a command such as the following:

tar -cvf /backup/etc.tar /etc

In this example, the tar command backs up the complete contents of the directory /etc to the file /backup/etc.tar.

The option **-c** (create) creates the archive. The option **-v** (verbose) displays a more detailed output of the backup process. The name of the archive to be created entered after the option **-f** (file).

This can either be a normal file or a device file (such as a tape drive), as in the following:

tar -cvf /dev/st0 /home

In this example, the /home directory is backed up to the tape recorder /dev/st0.

When an archive is created, absolute paths are made relative by default. This means that the leading / is removed, as in the following output:

```
tar: Removing leading / from member names
```

You can view the contents of an archive by entering the following:

tar -tvf /backup/etc.tar***Unpack tar Archives***

To unpack files from an archive, enter the following command:

tar -xvf /dev/st0

This writes all files in the archive to the current directory. Due to the relative path specifications in the tar archive, the directory structure of the archive is created here.

If you want to extract to another directory, this can be done with the option **-C**, followed by the directory name.

If you want to extract just one file, you can specify the name of the file with the **-C** option, as in the following:

```
tar -xvf /test1/backup.tar -C /home/user1/.bashrc
```

Exclude Files from Backup

If you want to exclude specific files from the backup, a list of these files must be written in an exclude file, line by line, as in the following:

```
/home/user1/.bashrc  
/home/user2/Text*
```

In this example, the file `/home/user1/.bashrc` from user1 and all files that begin with `Text` in the home directory of user2 will be excluded from the backup.

This list is then passed to tar with the option **-X**, as in the following:

```
tar -cv -X exclude.files -f /dev/st0 /home
```

Perform Incremental and Differential Backups

In an incremental or differential backup, only files that have been changed or newly created since a specific date must be backed up.

The following are 2 methods you can use to accomplish the same thing with tar:

- Use a Snapshot File for Incremental Backups
- Use find to Search for Files to Back Up

Use a Snapshot File for Incremental Backups

Tar lets you use a snapshot file that contains information about the last backup process. This file needs to be specified with the **-g** option.

First, you need to make a full backup with a tar command, as in the following:

```
tar -cz -g /backup/snapshot_file -f /backup/backup_full.tar.gz /home
```

In this example, the directory /home is backed up to the file /backup/backup_full.tar.gz. The snapshot file /backup/snapshot_file does not exist and is created.

The next time, you can perform an incremental backup with the following command:

```
tar -cz -g /backup/snapshot_file -f /backup/backup_mon.tar.gz /home
```

In this example, tar uses the snapshot file to determine which files or directories have changed since the last backup. Only changed files are included in the new backup /backup/backup_mon.tar.gz.

Use find to Search for Files to Back Up

You can also use the **find** command to find files that need to be backed up as a differential backup.

First, you use the following command to make a full backup:

```
tar -czf /backup/backup_full.tar.gz /home
```

In this example, the /home directory is backed up into the file /backup/backup_full.tar.gz. Then you can use the following command to back up all files that are newer than the full backup:

```
find /home -type f -newer /backup/backup_full.tar.gz \ -print0 |  
tar --null -cvf /backup/backup_mon.tar.gz -T -
```

In this example, all files (-type f) in the directory /home that are newer than the file /backup/backup_mon.tar.gz are archived.

The options **-print0** and **--null** ensure that files with spaces in their names are also archived. The option **-T** determines that files piped to stdin are included in the archive.


One problem with the previous command line might be caused by its long execution time (when you have to backup a lot of data). If a file is created or changed after the backup command is started but before the backup is completed, this file is older than the reference backup archive but at the same time not included in this archive.

This would lead to the situation, that such a file is not backed up in the next incremental backup run, as only files are included which are newer than the reference archive. Instead of the previous backup archive, you can also create a file with the command **touch** and use this file as reference in the find/tar command line.

Use tar Command Line Options

The following are some useful tar options:

Table 9-1

Options	Description
-c	Creates an archive.
-C	Changes to the specified directory.
-d	Compares files in the archive with those in the file system.
-f	Uses the specified archive file or device.
-j	Directly compresses or decompresses the tar archive using bzip2, a modern efficient compression program.
-r	Appends files to an archive.
-u	Only includes files in an archive that are newer than the version in the archive (update).
-v	Displays the files, which are being processed (verbose mode).
-x	Extracts files from an archive.
-X	Excludes files listed in a file.
-z	Directly compresses or decompresses the tar archive using gzip.
 For more information about tar, consult the man page for tar.	

Exercise 9-2 Create Backup Files with tar

In this exercise, you learn how to use tar.

You will find the exercise in the workbook.

(End of Exercise)

Objective 4 Work with Magnetic Tapes

To work with magnetic tapes in SUSE Linux Enterprise Server 10, use the command **mt**. With this command, you can position tapes, switch compression on or off (with some SCSI-2 tape drives), and query the tape status.

Magnetic tape drives used under Linux are always SCSI devices and can be accessed with the following device names:

- **/dev/st0**. Refers to the first tape drive.
- **/dev/nst0**. Addresses the same tape drive in the no rewind mode. This means that after writing or reading, the tape remains at that position and is not rewound back to the beginning.

For reasons of compatibility with other UNIX versions, 2 symbolic links exist: **/dev/rmt0** and **/dev/nrmt0**.

You can query the status of the tape by entering the following command:

mt -f /dev/st0 status

In this example, the **-f** option is used to indicate the device name of the tape drive. The command status displays the status of the tape drive.

The output of the command looks like the following:

```
drive type = Generic SCSI-2 tape drive
status = 620756992
sense key error = 0
residue count = 0
file number = 0
block number = 0
Tape block size 0 bytes. Density code 0x25 (unknown). Soft error count
since last status=0
General status bits on (41010000):
  BOT ONLINE IM_REP_EN
```


The most important information in this example is the file number (file number, starting at 0) and the block numbers (block number, starting at 0).

These parameters determine the position of the tape. In this example, the tape is positioned at the beginning of the first file.



The file count starts with 0.

To position the tape at the beginning of the next file, use the following command:

mt -f /dev/nst0 fsf 1

In this example, the command `fsf` forwards the tape by the given number of files, and the tape will start before the first block of the second file.

This can be verified with the status command, as in the following:

```
mt -f /dev/nst0 status
drive type = Generic SCSI-2 tape drive
status = 620756992
sense key error = 0
residue count = 0
file number = 1
block number = 0
Tape block size 0 bytes.
Density code 0x25 (unknown).
Soft error count since last status=0
General status bits on (81010000):
    EOF ONLINE IM_REP_EN
```

Now the file number is set to 1, and the final line of the output contains EOF (end of file) instead of BOT (beginning of tape).

With the option **bsf**, the tape can be repositioned back by a corresponding number of files.

In general, when positioning the tape, you should use a non rewinding device file like `/dev/nst0`.

If you want the tape to be spooled back to the beginning after the reading or writing process, enter the following command:

`mt -f /dev/nst0 rewind`

If you want to eject the tape from the drive, then enter the following command:

`mt -f /dev/nst0 offline`

Normally, tapes should always be written without compression, otherwise you cannot recover the subsequent data in case of a write or read error.

To check whether data compression is switched on or off, enter the following command:

`mt -f /dev/st0 datcompression`

The command shows whether data compression is switched on or off.

If the parameter `on` or `off` is specified at the end of the command, then data compression will be switched on or off. By default, compression is switched on.

Objective 5 Copy Data with dd

You can use the command **dd** to convert and copy files byte-wise. Normally dd reads from the standard input and writes the result to the standard output. But with the corresponding parameters, files can also be addressed directly.

You can copy all kinds of data with this command, including entire hard disk partitions. Exact copies of an installed system (or just parts of it) can be created very simply.

In the simplest case, a file can be copied with the following command:

dd if=/etc/protocols of=protocols.org

The output of dd during the copying process looks like following:

```
12+1 records in
12+1 records out
```

Use the option **if=** (input file) to specify the file to be copied, and the option **of=** (output file) to specify the name of the copy.

Copying files in this way is done using records. The standard size for a record is 512 bytes. The output shown above indicates that 12 complete records of the standard size and an incomplete record (that is, less than 512 bytes) were copied.

If the record size is now modified by the option **bs=***block size*, then the output will also be modified:

```
dd if=/etc/protocols of=protocols.old bs=1
6561+0 records in
6561+0 records out
```

A file listing shows that their sizes are identical:

```
ls -l protocols*
-rw-r--r-- 1 root root 6561 Apr 30 11:28 protocols
-rw-r--r-- 1 root root 6561 Apr 30 11:30 protocols.old
```

If you want to copy a complete partition, then the corresponding device file of the partition should be given as the input, as in the following:

dd if=/dev/sda1 of=boot.partition

In this example, the whole partition /dev/sda1 is written to the file boot.partition.

You can also use dd to create a backup copy of the MBR (master boot record), as in the following:

dd if=/dev/sda of=/tmp/mbr_copy bs=512 count=1

In this example, a copy of the MBR is created from the hard disk /dev/sda and is written to the file /tmp/mbr_copy.

Exercise 9-3 Create Drive Images with dd

In this exercise, you use dd to create a drive image.

You will find the exercise in the workbook.

(End of Exercise)

Objective 6 **Mirror Directories with rsync**

The command **rsync** (remote synchronization) is actually intended to create copies of complete directories across a network to a different computer.

When copying data, rsync compares the source and the target directory and transfers only data that has changed or been created.

rsync is the ideal tool to mirror the content of directories or to back up data across a network.

You can use rsync in 2 different ways:

- Perform Local Copying with rsync
- Perform Remote Copying with rsync

Perform Local Copying with rsync

You can mirror all home directories by entering the following:

rsync -a /home /shadow

In this example, the mirroring is made to the directory /shadow.

The directory /home is first created in the directory /shadow, and then the actual home directories of the users are created under /home.

If you want to mirror the content of a directory and not the directory itself, you can use a command such as the following:

rsync -a /home/. /shadow

By adding a /. to the end of the source directory, only the data under /home is copied.

If you run the same command again, only files that have changed or that are new will be transferred.

The option `-a` used in the examples puts `rsync` into archive mode. Archive mode is a combination of various other options (namely `rlptgoD`) and ensures that the characteristics of the copied files are identical to the originals.

The following describes these options:

- Symbolic links (option **l**)
- Access permissions (option **p**)
- Owners (option **o**)
- Group membership (option **g**)
- Time stamp (option **t**)

The option `-r` ensures that directories are copied recursively.

The following are some useful `rsync` options:

Table 9-2

Option	Description
-a	Puts <code>rsync</code> into the archive mode.
-x	Saves files on one file system only, which means that <code>rsync</code> does not follow symbolic links to other file systems.
-v	Enables the verbose mode. Use verbose mode to outputs information about the transferred files and the progress of the copying process.
-z	Compresses the data during the transfer. This is especially useful for remote synchronization.
--delete	Deletes files that no longer exist in the original directory from the mirrored directory.
--exclude-from	Does not back up files listed in an exclude file.

The last option can be used as follows:

```
rsync -a --exclude-from=/home/exclude /home/. /shadow/home
```

In this example, all files listed in the file `/home/exclude` are not backed up. Empty lines or lines beginning with `;` or `#` are ignored.

Perform Remote Copying with rsync

With `rsync` and `SSH`, you can log in to other systems and perform data synchronization remotely over the network.

The following command copies the home directory of the user `tux` to a backup server:

```
rsync -ave ssh root@DA1:/home/tux /backup/home/
```

In this example, the option `-e` specifies the remote shell (`ssh`) that should be used for the transmission. The source directory is specified by the expression `root@DA1:/home/tux`. This means that `rsync` should log in to `DA1` as `root` and transfer the directory `/home/tux`.

Of course, this also works in the other direction. In the following example, the backup of the home directory is copied back to the `DA1` system:

```
rsync -ave ssh /backup/home/tux root@DA1:/home/
```



`rsync` must be installed on both the source and the target computer.

There is also another way to perform remote synchronization with `rsync` by running an `rsync` server. This way you can enable remote synchronization without allowing an `SSH` login.



For more information, consult the rsync documentation at
<http://samba.anu.edu.au/rsync/>.

Exercise 9-4 Create a Backup of a Home Directory with rsync

In this exercise, you learn how to use rsync.

You will find this exercise in the workbook.

(End of Exercise)

Objective 7 Automate Data Backups with cron

Backing up data is a task that you should perform on a regular basis. You can automate backups in Linux with the cron service.

System jobs are controlled with the file `/etc/crontab` and the files in the directory `/etc/cron.d`. They are defined with the scripts in the directories `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly`, and `/etc/cron.monthly`.

Specifying which users can create cron jobs is done through the files `/var/spool/cron/allow` and `/var/spool/cron/deny`, which are evaluated in this order. If both files do not exist, then only root can define jobs.

The jobs of individual users are stored in files in the directory `/var/spool/cron/tabs` with names matching the user names. These files are processed with the command **`crontab`**.

The following is an example of a cron job:

```
0 22 * * 5 /root/bin/backup
```

In this example, the script `/root/bin/backup` is started every Friday at 10 P.M. The format for the line is described in `man crontab`.

Exercise 9-5 Configure a cron Job for Data Backups

In this exercise, you use cron for data backup.

You will find this exercise in the workbook.

(End of Exercise)

Summary

Objective	Summary
1. Develop a Backup Strategy	<p>To develop a backup strategy, you need to complete the following steps:</p> <ul style="list-style-type: none">■ Choose a backup method■ Choose a backup media <p>There are 3 basic backup strategies:</p> <ul style="list-style-type: none">■ Full backup. All data is backed up every day.■ Incremental backup. Only the data that has been changed since the last Incremental or full backup is saved every day.■ Differential backup. Only the data that has been changed since the last full backup is saved every day. <p>Which method you use depends on the backup window.</p> <p>The backup window is the time period in which a system is not used and is available for a backup.</p>

Objective	Summary
2. Backup Files with YaST	<p>YaST provides a backup and a restore module, which can be used to create system backups.</p> <p>The modules are located in the system section of the YaST control center and are called:</p> <ul style="list-style-type: none">■ System Backupand■ System Restoration

Objective	Summary
3. Create Backups with tar	<p>tar is a commonly used tool for performing data backups under Linux.</p> <p>tar can write data directly to a backup media or to an archive file.</p> <p>Archive files normally end in .tar, if they are compressed in .tar.gz or .tgz.</p> <p>The following is the basic syntax to create a tar archive:</p> <p>tar -cvf home.tar /home</p> <p>To unpack a tar archive, use the following command:</p> <p>tar -xvf /home.tar</p> <p>If you want to use tar with gzip for compression, you need to add the option z to the tar command.</p> <p>Archives can also be written directly to tape drives.</p> <p>In this case, the device name of the tape drive must be used instead of a filename.</p> <p>tar can also be used for incremental or differential backups.</p>

Objective	Summary
4. Work with Magnetic Tapes	<p>mt is the Linux standard tool to work with magnetic tapes.</p> <p>Use the following command to query the status of the drive:</p> <p>mt -f /dev/st0 status</p> <p>The following command moves the tape to the beginning of the next file:</p> <p>mt -f /dev/nst0 fsf 1</p> <p>To rewind the tape by a certain amount of files, use the bsf command.</p> <p>To rewind the tape to the beginning, use the following:</p> <p>mt -f /dev/nst0 rewind</p> <p>The following command ejects the tape from the drive:</p> <p>mt -f /dev/nst0 offline</p>
5. Copy Data with dd	<p>With the command dd files can be converted and copied byte-wise.</p> <p>To copy a file, use the following command:</p> <p>dd if=/etc/protocols of=protocols.org</p> <p>To copy an entire partition into a file, use the following command:</p> <p>dd if=/dev/sda1 of=boot.partition</p>

Objective	Summary
6. Mirror Directories with rsync	<p>The command rsync is used to synchronize the content of directories, locally or remotely, over the network.</p> <p>rsync uses special algorithms to ensure that only those files are transferred that are new or have been changed since the last synchronization.</p> <p>The basic command to synchronize the content of two local directories is the following:</p> <p>rsync -a /home /shadow</p> <p>To perform a remote synchronization, use a command like the following:</p> <p>rsync -ave ssh root@DA1:/home/tux /backup/home/</p>
7. Automate Data Backups with cron	<p>Because backups are recurring tasks, they can be automated with the cron daemon.</p> <p>System jobs are controlled using the file <code>/etc/crontab</code> and the files in the directory <code>/etc/cron.d</code>.</p> <p>The jobs are defined by the scripts in the directories <code>/etc/cron.hourly</code>, <code>/etc/cron.daily</code>, <code>/etc/cron.weekly</code> and <code>/etc/cron.monthly</code>.</p> <p>The following is an example of a job entry:</p> <p>0 22 * * 5 /bin/backup</p>

CNI USE ONLY-1 HARDCOPY PERMITTED

SECTION 10 Manage Printing

The first objective in this section covers configuring printing on the local machine, using either a locally connected printer or a printer available in the local network.

The next objective deals with management of the print queues using CUPS (Common UNIX Printing System) command line tools.

The following objectives cover information on how CUPS works and how to make local printers available for others in the network. Access control, CUPS configuration and other advanced topics are also covered.

CUPS is based on the Internet Printing Protocol (IPP). This protocol is supported by most printer manufacturers and operating systems. IPP is a standardized printer protocol that enables authentication and access control.

While SUSE Linux Enterprise Server 10 also supports the traditional LPRng printing system, this section is limited to CUPS, because it is the default printing system for the SUSE Linux Enterprise Server 10.

Objectives

1. Configure Local Printing
2. Manage Print Jobs and Queues
3. Understand How CUPS Works
4. Configure and Manage a Print Server
5. Use the Web Interface to Manage a CUPS Server

Objective 1 Configure Local Printing

YaST provides printer installation and configuration functionality. To configure a printer, you need to know the following:

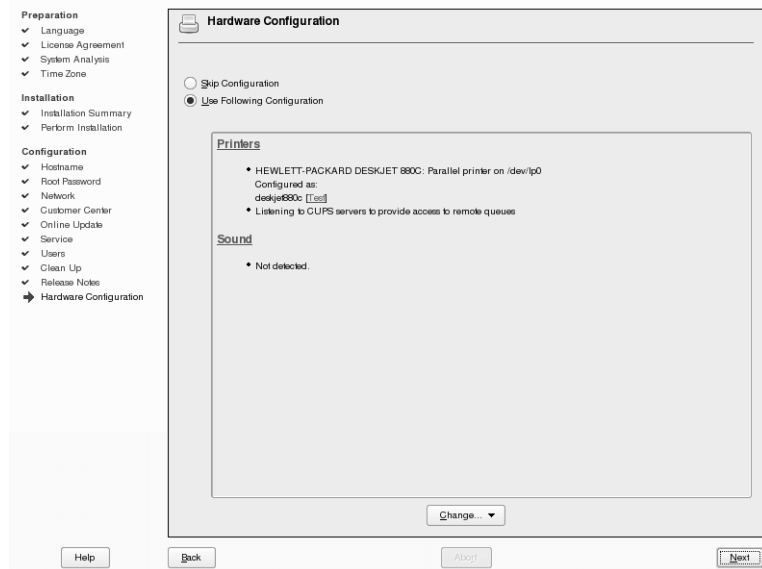
- When to Configure a Printer
- Required Printing Software
- Add a Printer

When to Configure a Printer

You can configure your printer at the following times:

- **During installation.** If you are at the Hardware Configuration dialog during installation (see in the following figure) and your automatic detection is not correct, select the **Printers** link or use the **Change...** drop-down list:

Figure 10-1



Note that during installation, only locally connected printers are detected automatically and listed under Printers.

- **After installation.** You can change your printer configuration settings from the YaST Control Center by selecting **Hardware > Printer**.

With the command **yast2 printer**, it is also possible to start the YaST printer configuration module directly from a terminal window.

Required Printing Software

The following packages are needed to set up a print server:

Table 10-1

Package	Content
cups	Provides the printer daemon cupsd
cups-client	Provides the command-line printing tools
cups-libs	Should always be installed, because a number of programs (such as Samba) are linked against the CUPS libraries
cups-drivers cups-drivers-stp	Provide the PPD files for print queues
cups-SUSE-ppds-dat	Provides a pregenerated file /etc/cups/ppds.dat

These files are installed automatically if YaST is used for printer configuration.

YaST also creates the symbolic links in runlevel directories to ensure that the CUPS daemon is started automatically when booting.

Other packages required by the printing system, such as ghostscript-library, are automatically selected during a standard installation.

Add a Printer

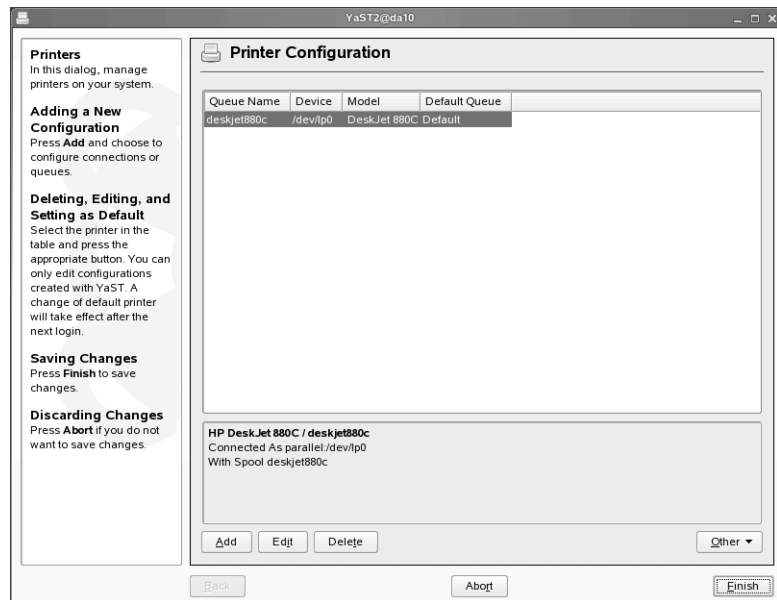
There are two ways to add printers:

- Add a Printer with YaST
- Add a Printer from the Command Line

Add a Printer with YaST

The Printer Configuration dialog to configure your printer is the same during and after installation:

Figure 10-2

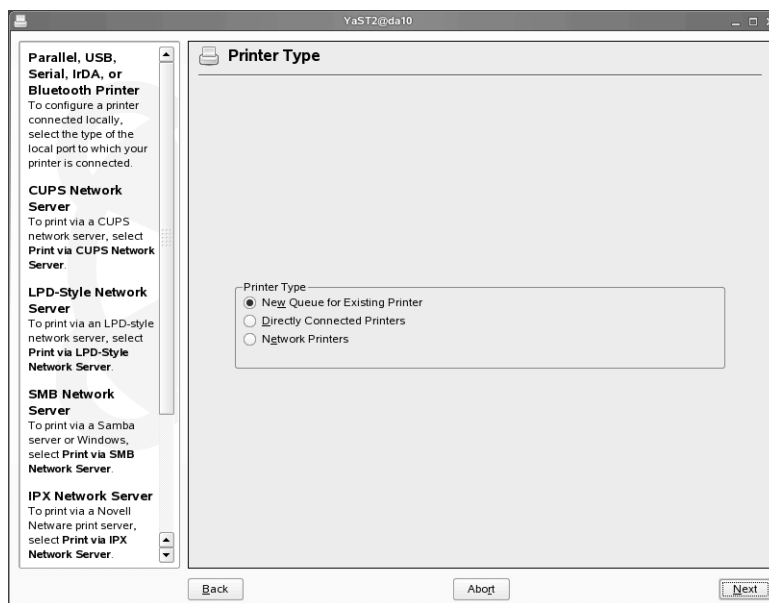


The upper part of the Printer Configuration dialog lists the printers that have already been configured and any automatically detected printers.

If there are any printers listed in the upper part, the lower part of the dialog shows details for the selected printer.

To add a printer that does not show up in the upper part of the dialog, for example, a network printer, select **Add**. The following appears:

Figure 10-3



Depending on your selection here, the next dialog offers more specific choices.

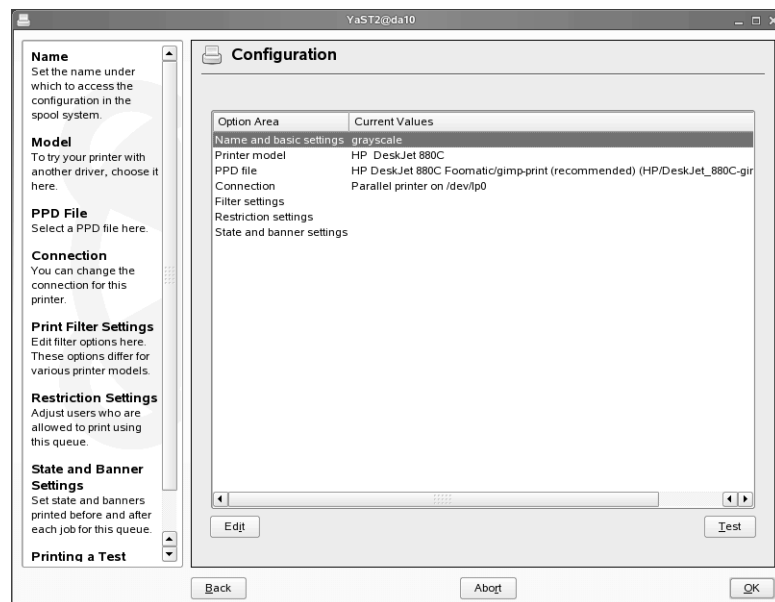
- New Queue for Existing Printer
- Directly Connected Printers
- Network Printers

New Queue for Existing Printer

This option appears only if there is already a printer configured. Adding a queue to an existing printer is useful if you want to use, for instance, a different resolution or print quality by printing to a different queue. Selecting **New Queue for Existing Printer > Next** opens a dialog where you select the printer to which you want to add a queue. Select **Next** once more.

In the following Configuration dialog select an entry and then **Edit**.

Figure 10-4

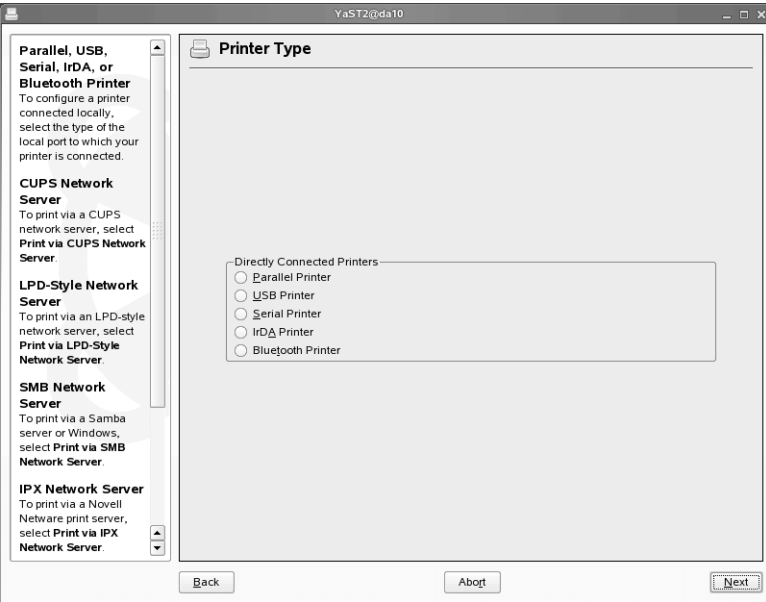


Make any desired changes for that queue. You can test your printer configuration by selecting **Test**. **OK** creates the queue and returns you to the Printer Configuration dialog.

Directly Connected Printers

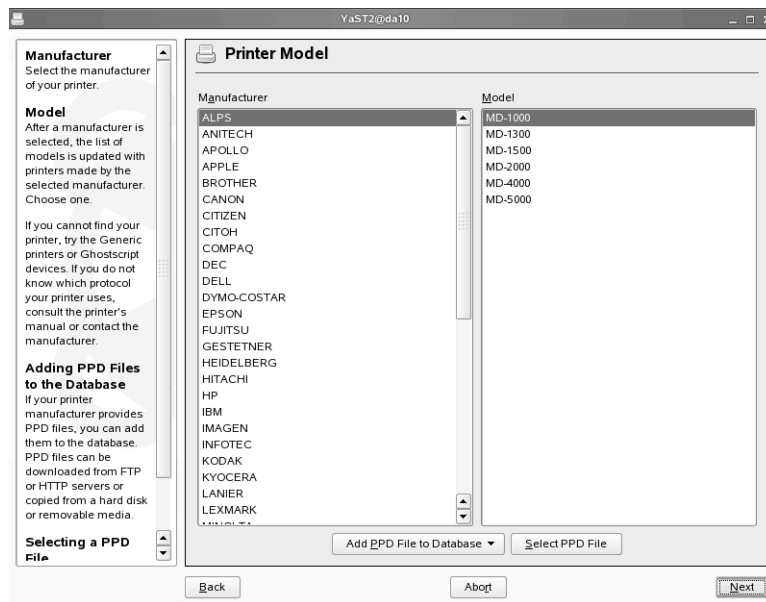
Most directly connected printers are detected automatically. If not, you can select the connection type:

Figure 10-5



You are then guided through a number of dialogs where you can add details for the connection type, and select the printer model as in the following:

Figure 10-6



In the next dialog you can modify the configuration for this model as needed (same dialog as shown in Figure 10-4).

Network Printers

CUPS supports the IPP, LPD, SMB, and socket protocols. The following describes these protocols:

- **IPP (Internet Printing Protocol).** IPP is a relatively new protocol (since 1999) that is based on the HTTP protocol. Compared to other protocols, it is possible to transmit much more job-related data.

CUPS uses IPP for the internal data transmission. This is the preferred protocol for a forwarding queue between CUPS servers.

The port number for IPP is 631.

Device URI (Universal Resource Identifier) example:

ipp://cupsserver/printers/printqueue. The Device URI can be used to specify a printer. See “Add a Printer from the Command Line” on page 10-19.

- **LPD (Line Printer Daemon)**. The LPD protocol is described in RFC 1179 (requests for comments can be found at <http://www.ietf.org/rfc.html>).

Some job-related data such as the printer queue is sent before the actual print data. This means that a printer queue must be specified when configuring the LPD protocol for the data transmission.

The implementations of most printer manufacturers are flexible enough to accept any name as the printer queue. If necessary, the printer manual might indicate which name to use (such as LPT, LPT1, or LP1).

Of course, an LPD queue can also be configured on a different Linux or UNIX host in a network that uses the CUPS system. The port number for an LPD service is 515.

Device URI example: **lpd://host-printer/LPT1**

- **SMB (Standard Message Block)**. CUPS also supports printing on printers connected to Windows shares. The protocol used for this purpose is SMB.

SMB uses port numbers 137, 138, and 139.

Device URI examples:

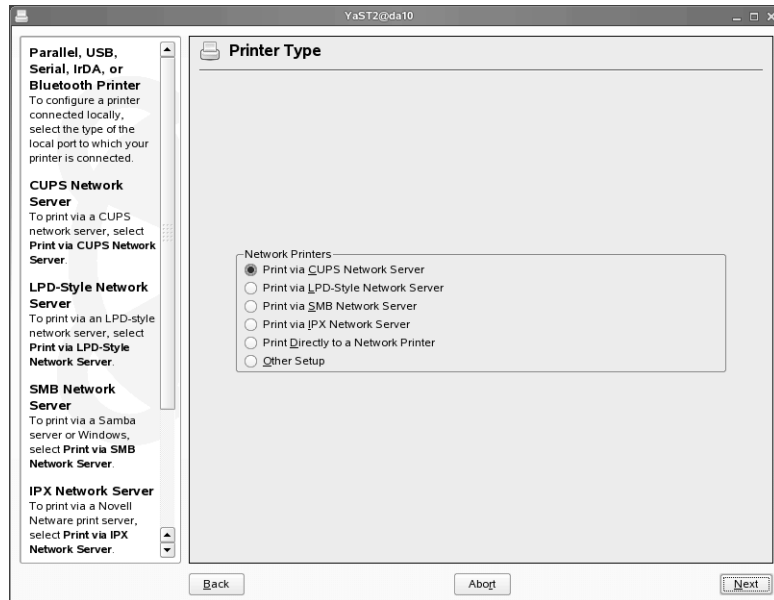
smb://user:password@workgroup/server/printer
smb://user:password@host/printer
smb://server/printer

- **socket.** This protocol is used to connect to a printer equipped with a network port, as HP's JetDirect technology. Some of the socket port numbers that are commonly used are 9100 or 35.

Device URI example: **socket://host-printer:9100/**

The choice **Network Printer** in the **Printer Type** dialog (Figure 10-3) opens a dialog where you can select these protocols:

Figure 10-7



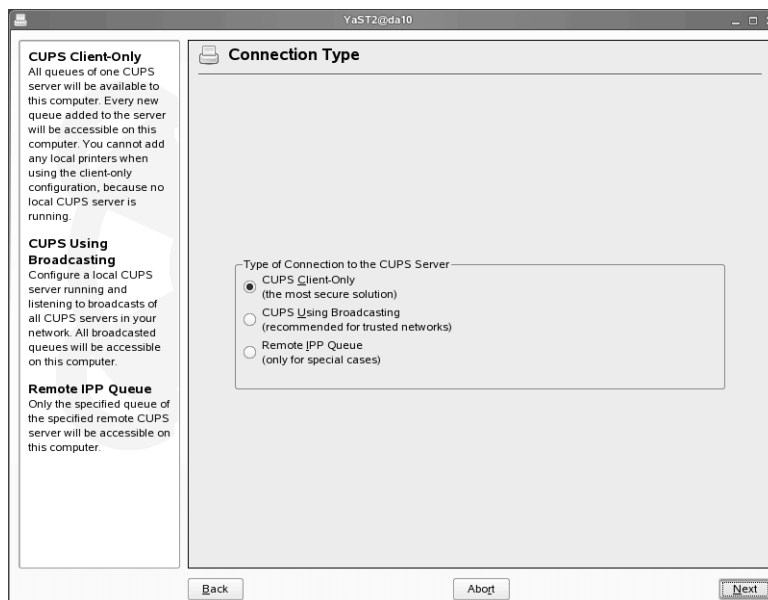
The next dialogs vary depending on your choice. We will cover two of them here:

- Print via CUPS Network Server
- Print Directly to a Network Printer

Print via CUPS Network Server

If you choose **Print via CUPS Network Server**, the following dialog comes up:

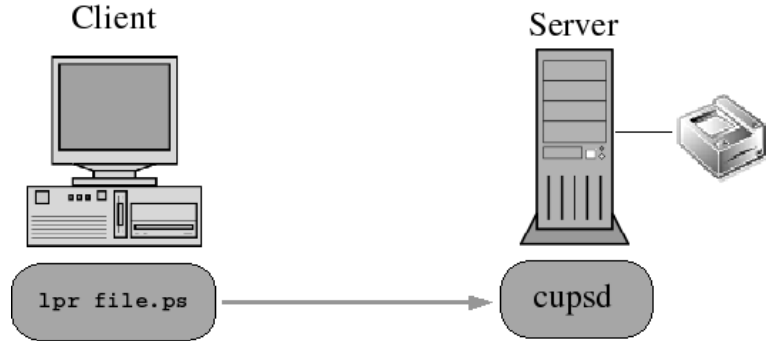
Figure 10-8



The options are described in the help text to the left.

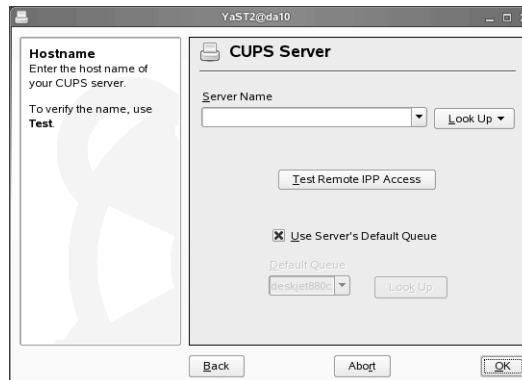
CUPS Client Only. Allows you to access the print queues on one specific server only. No printer daemon is running locally:

Figure 10-9



After a warning message, the next dialog allows you to enter the IP address or fully qualified domain name (FQDN) of the print server and which queue to use.

Figure 10-10



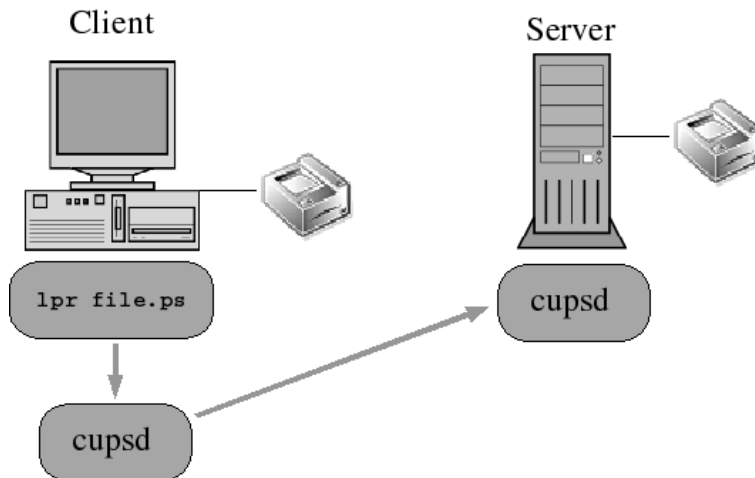
The CUPS server to use is added to the file `/etc/cups/client.conf`.

```
# /etc/cups/client.conf
#
ServerName 10.0.0.2
```

This type of setup is a good choice only if you have just one print server for the entire network.

CUPS Using Broadcasting. Probably the best choice within a local network. With CUPS running locally, you can print on your locally connected printers as well as on those that are broadcasted by other servers within your network. New printers broadcasted in the network appear automatically and are available to the users.

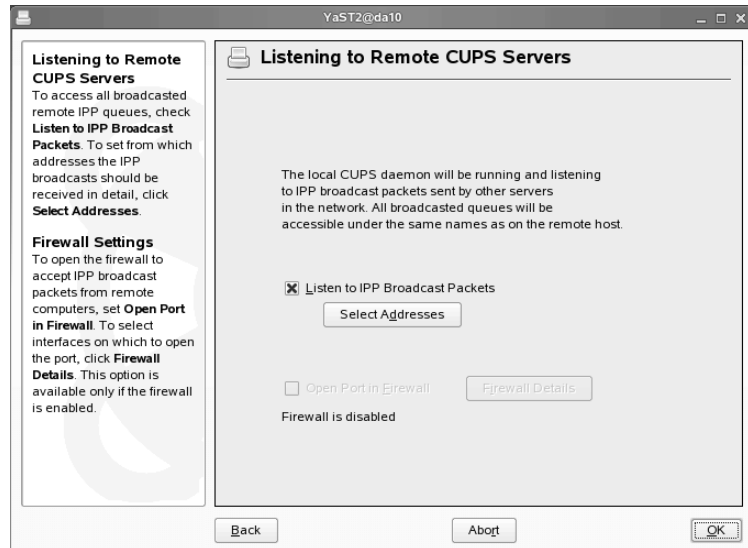
Figure 10-11



Details on how to make local printers available for others or to restrict access via the network to local printers are covered in the objective “Configure and Manage a Print Server” on page 10-42.

With CUPS Using Broadcasting, you can modify the firewall settings (if the firewall is active) and to specify from which addresses to accept broadcasts.

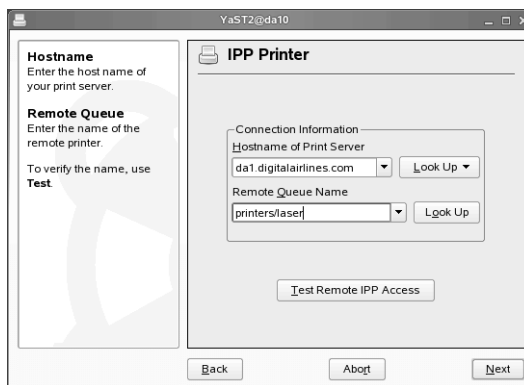
Figure 10-12



OK returns you to the Printer Configuration dialog.

Remote IPP Queue. Accesses a specific queue on a specific server:

Figure 10-13

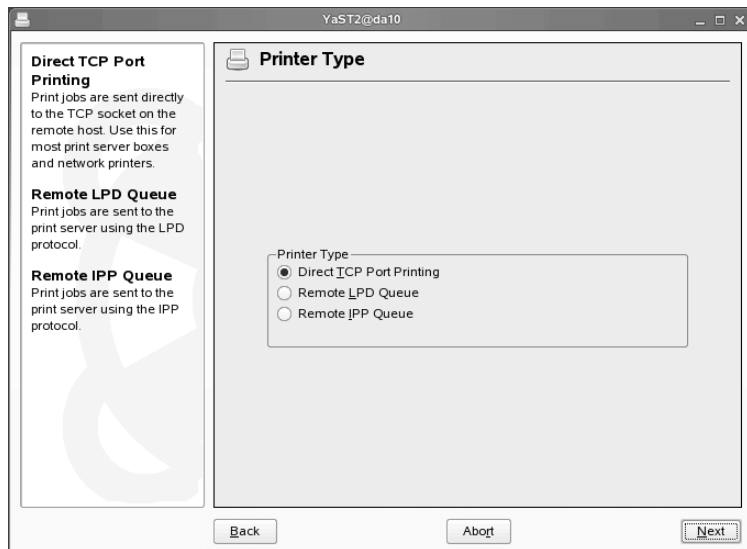


Unlike in the client-only configuration, a local CUPS server is running, and locally connected printers remain accessible.

Print Directly to a Network Printer

Print Directly to a Network Printer is the option to choose for printers equipped with a network card. For such a printer select **Direct TCP Port Printing** in the this dialog:

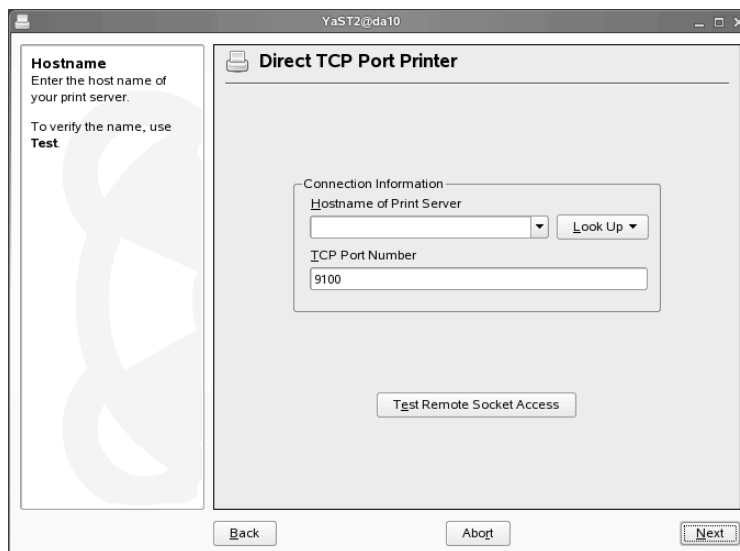
Figure 10-14



The option **Remote LPD Queue** leads to the same dialog as Print via LPD Style Network Server in the previous dialog (see Figure 10-7). **Remote IPP Queue** could be used for possible future CUPS queues that are not configurable via YaST.

After selecting **Direct TCP Port Printing** and clicking on **Next**, enter the hostname or IP address of the printer.

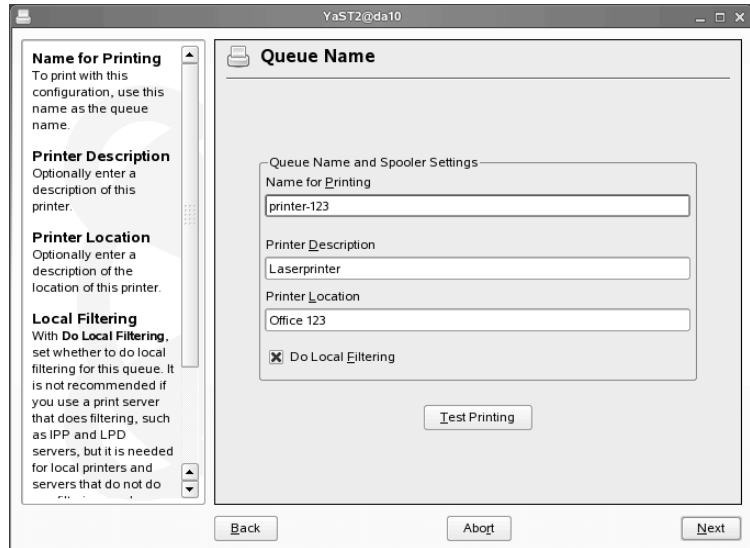
Figure 10-15



The screenshot shows a window titled "YaST2@da10" with a "Direct TCP Port Printer" configuration panel. On the left, a sidebar titled "Hostname" contains instructions: "Enter the host name of your print server." and "To verify the name, use **Test**". The main panel has a "Connection Information" section with a "Hostname of Print Server" dropdown menu, a "Look Up" button, and a "TCP Port Number" text field containing "9100". Below this is a "Test Remote Socket Access" button. At the bottom of the window are "Back", "Abort", and "Next" buttons.

Next brings you to a dialog where you can give a name to the print queue, and add information on the printer and its location.

Figure 10-16



Clicking **Next** again brings you to the dialogs already covered in Figure 10-6 and Figure 10-4, then you are returned to the Printer Configuration dialog (Figure 10-2).

If you want to edit a printer configuration, select it in the Printer Configuration dialog and then click **Edit**. If you want to delete a print queue, select it and click **Delete**.

Add a Printer from the Command Line

Besides using YaST, you can also configure CUPS with command line tools. After collecting the information you need (such as the PPD (Postscript Printer Description) file and the name of the device), enter the following:

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

The option **-p** specifies the print queue name of the printer, the option **-v** sets the device URI attribute of the printer queue (see “Network Printers” on page 10-8), and the option **-P** is used to specify the PPD file.

Do not use **-E** as the first option. For all CUPS commands, **-E** as the first argument implies the use of an encrypted connection and **-E** at the end enables the printer to accept print jobs.

For example, to enable a parallel printer, enter a command similar to the following:

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

To enable a network printer, enter a command similar to the following:

```
lpadmin -p ps -v socket://10.0.0.200:9100/ -P \  
/usr/share/cups/model/Postscript-level1.ppd.gz -E
```

Exercise 10-1 Change Your Printer Configuration

In this exercise you add a local printer and print to a remote queue.

You will find this exercise in the workbook.

(End of Exercise)

Objective 2 **Manage Print Jobs and Queues**

CUPS comes with several command line tools to start, stop, and modify print queues. The command line tools for the CUPS printing system and their man pages are included in the package cups-client.

Documentation for these tools is installed with the package cups in /usr/share/doc/packages/cups/:

- ***CUPS Software Users Manual***: sum.html and sum.pdf
- ***CUPS Software Administration Manual***: sam.html and sam.pdf

The CUPS tools allow you to use commands according to two different styles or conventions, which are called.

- Berkeley style (Berkeley style commands are identical to those used with the LPRng printing system)
- System V style

Compared with Berkely style, System V provides a somewhat more extensive range of features for printer administration.

To manage printer queues, you need to know how to do the following:

- Generate a Print Job
- Display Information on Print Jobs
- Cancel Print Jobs
- Manage Queues
- Configure Queues
- Start and Stop CUPS

Print queues can also be managed via a web interface, which is covered later in this section.

Generate a Print Job

Use the following commands to generate a print job:

- Berkeley: **lpr -P *queue file***
- System V: **lp -d *queue file***

Example:

```
geeko@dal10:~ # lpr -P color chart.ps
```

or:

```
geeko@dal10:~ # lp -d color chart.ps
```

With these commands, the file `chart.ps` is submitted to the color queue.

If no queue is specified, the job is printed to the default queue.

The `-o` parameter needs to be used whenever any additional print options are specified:

```
geeko@dal10:~ # lpr -P lp -o duplex=none order.ps
```

or:

```
geeko@dal10:~ # lp -d lp -o duplex=none order.ps
```

This submits the file `order.ps` to the `lp` queue and also disables duplex printing for the corresponding device (`duplex=none`). To view possible options, enter **lpoptions -l** (see “Configure Queues” on page 10-26).

The command must be given in a slightly different form to print through a remote queue:

- Berkeley: **lpr -P *queue*@*server file***
- System V: **lp -d *queue* -h *server file***

Example:

```
geeko@da10:~ # lpr -P lp@da101.digitalairlines.com \
/etc/motd
```

or:

```
geeko@da10:~ # lp -d lp -h da101.digitalairlines.com \
/etc/motd
```

This submits the file `/etc/motd` to the `lp` queue located on the print server `da101.digitalairlines.com`.



For more information on these command line tools, enter **man lpr** and **man lp**.

Display Information on Print Jobs

Use the following commands to display print job information:

- Berkeley: **lpq -P *queue***
- System V: **lpstat -o *queue* -p *queue***

The `lpq` command displays active print jobs of the default queue in the following way:

```
geeko@da10:~ # lpq
draft is ready and printing
Rank   Owner   Job      File(s)   Total Size
active  root    14       fstab     1024 bytes
```

lpq -l lists the same information in a slightly different format.

To display the print jobs of another queue, enter the option **-P queue**:

```
geeko@dal10:~ # lpq -P printer
printer is ready
no entries
```

To display the active print jobs of all available queues, enter **lpq -a**:

```
geeko@dal10:~ # lpq -a
no entries
```

To actualize the output in a fixed interval, enter **lpq -P queue +seconds**

The following shows the output of **lpstat -o queue -p queue**; **lpstat -a** shows information on the accepting state:

```
geeko@dal10:~ # lpstat -o draft -p draft
draft-14      root      1024    Thu Mar 30 15:08:54 2006
printer draft now printing draft-14.  enabled since Jan 01
00:00

        Connected to host, sending print job...
geeko@dal10:~ # lpstat -a
draft accepting requests since Jan 01 00:00
printer accepting requests since Jan 01 00:00
```



For more information on these commands, enter **man lpq** and **man lpstat**.

Cancel Print Jobs

Use the following commands to cancel a print job:

- Berkeley: **lprm -P queue jobnumber**
- System V: **cancel [-h server] queue-jobnumber**



For more information on these commands, enter **man lprm** and **man cancel**.

Manage Queues

In addition to controlling single jobs in a queue, you can also control the queue as such.

- Disable printing on a queue while jobs can still be sent to it by entering **/usr/bin/disable printer**

Queues that are disabled still accept jobs for printing but won't actually print any files until they are enabled again.

Disabling a print queue is useful if a printer malfunctions and you need time to fix the problem.

- Start printing again on a queue that is disabled by entering **/usr/bin/enable printer**

If there are any queued print jobs, they are printed after the printer is enabled.

You need to enter the path with the command, as enable is also a bash built-in command.

- Stop accepting print jobs on a queue by entering **/usr/sbin/reject printer**

With the command /usr/sbin/reject, the printer finishes the print jobs in the queue but rejects any new print jobs.

This command is useful for times when you need to perform maintenance on a printer and the printer will not be available for a significant period of time.

Note: **lpstat -a** shows information on the accepting state of the queues.

- Accept print jobs again on a queue that rejected them by entering **/usr/sbin/accept printer**

By using this command, you can reset the print queue to begin accepting new print jobs. (Note: If the queue is also disabled, actual printing starts only after enabling the queue again.)

Configure Queues

Printer-specific options that affect the physical aspects of the output are stored in the PPD file for each queue in the directory

/etc/cups/ppd/

PPD (PostScript Printer Description) is the computer language that describes the properties (such as resolution) and options (such as duplex unit) of PostScript printers. These descriptions are necessary to use the various printer options in CUPS.

During the installation of SUSE Linux Enterprise Server, a lot of PPD files are preinstalled. In this way, even printers that do not have built-in PostScript support can be used.

If a PostScript printer is configured, the best approach is to get a suitable PPD file and store it in the directory `/usr/share/cups/model/`. You can then select the PPD file during the installation. If the model does not show up, select **Add PPD File to Database** in the Printer Model dialog (Figure 10-6) and follow the simple steps to add the PPD file to the database.

Users can see the current settings of a local queue by entering

lpoptions -p queue -l

Note: The sequence of options is important, if you specify `-l` first, the settings of the default queue are listed, no matter what you specify after `-p`.

The output of this command has the following structure:

option/string: value value value ...

The following is an example:

```
da10:~ # lpoptions -l
HalftoningAlgorithm/Halftoning Algorithm: Accurate *Standard WTS
REt/REt Setting: Dark Light *Medium Off
TonerDensity/Toner Density: 1 2 *3 4 5
Duplex/Double-Sided Printing: *DuplexNoTumble DuplexTumble None
Manualfeed/Manual Feed of Paper: Off On
InputSlot/Media Source: *Default Tray1 Tray2 Tray3 Tray4 Envelope Manual
Auto
Copies/Number of Copies: *1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ...
PageSize/Page Size: *A4 Letter 11x17 A3 A5 B5 Env10 EnvC5 EnvDL EnvISOB5
EnvMonarch Executive Legal
PageRegion/PageRegion: A4 Letter 11x17 A3 A5 B5 Env10 EnvC5 EnvDL EnvISOB5
EnvMonarch Executive Legal
Resolution/Resolution: 75x75dpi *150x150dpi 300x300dpi 600x600dpi
Economode/Toner Saving: *Off On
LowToner/Behaviour when Toner Low: *Continue Stop
```

The “*” symbol in front of a value indicates the currently active setting. The significance of some of these options is as follows:

- **REt/REt Setting.** (Resolution Enhancement) There are three modes to improve the quality of the dark, light, and medium print jobs.
Generally the difference in print quality is small.
- **TonerDensity/Toner Density.** This option specifies the quantity of toner (1=little, 5=much).
- **Duplex/Double-Sided Printing.** This option disables or enables double-sided printing, assuming that your printer supports duplex printing.
- **InputSlot/Media Source.** If your printer has different paper trays, you can select the tray for your print job with this option.
- **Copies/Number of Copies.** Number of copies printed.
- **PageSize/Page Size.** The physical size of the paper in the selected paper tray.
- **PageRegion/PageRegion.** Normally equal to the page size.

This option is read by the PostScript interpreter.

- **Resolution/Resolution.** Resolution used for the print queue.
- **Economode/Toner Saving.** To save toner, you can enable economode here, but the quality of your prints degrades.
- **LowToner/Behaviour when Toner Low.** Define if the printer continues or stops printing when the toner gets low.

To change any of the options for a local queue, enter a command with the following syntax:

lpoptions -p *queue* -o *option=value*

The following command changes the page size of the lp queue to Letter:

```
da10:~ # lpoptions -p lp -o PageSize=Letter
```

However, the range of users affected by the new settings varies, depending on which user has actually changed the settings:

- If a normal user (such as geeko) enters a command as above, the changes only apply to that user and are stored in the file `~/.lpoptions` (in the user's home directory).
- If root enters the command, changes apply to all users on the corresponding host.

They are then used as default and stored in the file `/etc/cups/lpoptions`.

The PPD file of the queue, however, is not modified by this.

There is a way for root to change the defaults in the PPD file of any local queue. Such changes would apply network wide to all users submitting print jobs to the corresponding queue.

To achieve this, enter (as root)

lpadmin -p *queue* -o *option=value*

CNI USE ONLY-1 HARDCOPY PERMITTED

For example, to set the default page size for the lp queue, enter

```
da10:~ # lpadmin -p lp -o PageSize=Letter
```

CUPS provides collections of printers called printer classes. Jobs sent to a class are forwarded to the first available printer in the class. You can also use the **lpadmin** command to

- Define classes of printers or queues.
- Edit such classes (by adding a queue to a class or deleting a queue from a class).
- Delete classes.

For example, to add a queue to a class, enter

lpadmin -p *queue* -c *class*

If the class does not exist yet, it will be automatically created.

To remove a queue from a class, enter

lpadmin -p *queue* -r *class*

If the class is empty (with no other queues left in it) as a result of such a command, it will be automatically deleted.

To see which queues belong to which class on a given host, look at the file `/etc/cups/classes.conf`.



For more information on all the available options of lpadmin, enter **man lpadmin**.



You can also get information on the commands covered above in a browser using the URL

file:///usr/share/doc/packages/cups/sum.html#USING_SYSTEM

and the URL

file:///usr/share/doc/packages/cups/ sum.html#STANDARD_OPTIONS.

For details about how to save printer options, read

/usr/share/doc/packages/cups/sum.html#SAVING_OPTIONS.

Start and Stop CUPS

As the root user, you can start or stop cupsd manually with the following commands:

- ***/etc/init.d/cups start or rccups start***
- ***/etc/init.d/cups stop or rccups stop***

If you make changes manually to the file `/etc/cups/cupsd.conf`, you need to restart the daemon by entering ***/etc/init.d/cups restart*** or ***rccups restart***.

Exercise 10-2 *Manage Printers from the Command Line.*

In this exercise, you practice managing printer queues from the command line.

You will find this exercise in the workbook.

(End of Exercise)

Objective 3 Understand How CUPS Works

To understand how CUPS works, you need to understand the following:

- Steps of the Printing Process
- Print Queues
- Log Files

Steps of the Printing Process

The printing process involves the following steps:

1. A print job is submitted by a user or by a program.
2. The file destined for the printer is stored in a print queue, which creates two files per print job in the directory

/var/spool/cups/

One of the file contains the actual data to print. The other one contains information about the print job; for example, the identity of the user who created the print job and the printer to use.

3. The cupsd printer daemon acts as the print spooler. It is responsible for watching all print queues and for starting the filters required to convert data into the printer-specific format.
4. The conversion of print data is done in the following way:
 - a. The data type is determined using the entries in
/etc/cups/mime.types
 - b. Subsequently, data is converted into PostScript using the program specified in
/etc/cups/mime.convs
 - c. After that, the program **pstops** (`/usr/lib/cups/filter/pstops`) is used to determine the number of pages, which is written to

/var/log/cups/page_log

- d. CUPS uses other filtering capabilities of pstops as needed, depending on the options set for the print job.

For instance, the **psselect** option of pstops makes it possible to limit the printout to a certain selection of pages, or the **ps-n-up** option of **pstops** allows to print several pages on one sheet.



To learn how to activate these filtering functions, see [/usr/share/doc/packages/cups/sum.html](http://usr/share/doc/packages/cups/sum.html).

- e. If the selected printer is not a PostScript printer, cupsd will start the appropriate filter to convert data into the printer-specific format.

One of these filter programs is

/usr/lib/cups/filter/cupsomatic

which in turn relies on ghostscript for conversion.

Filters are responsible for processing all printer-specific options, including resolution, paper size, and others.

- f. For the actual transfer of the data stream to the printer device, CUPS uses another type of filter, or back end, depending on how the printer is connected to the host.

These back ends are found in the directory

/usr/lib/cups/backend/.

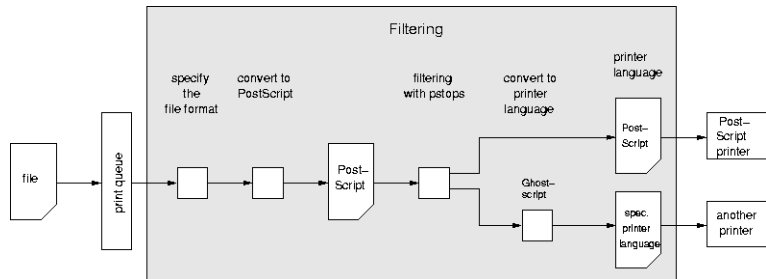
```
da10:~ # ls /usr/lib/cups/backend/
.  canon http lpd      parallel scsi      smb       usb
.. epson ipp   novell pipe      serial  socket
```

5. Once the print job has been transferred to the printer, the print spooler deletes the job from the queue and starts processing the next job. When the job is deleted, the print data file in `/var/spool/cups/` is removed.

The file that has information about the print job is not deleted. The filename for the first print job is labeled c00001. The number in each of the following print jobs is increased by one.

The following is a schematic representation of the filtering process:

Figure 10-17



Print Queues

With CUPS, printer devices are addressed using print queues. Rather than sending print jobs directly to the printer, they are sent to a print queue associated with the device. On a print server, each print queue is registered with its name in the file

/etc/cups/printers.conf

Among other things, this file defines through which queues the printer is addressed, how it is connected, and to which interface it is connected.

Several print queues can be defined for one printer, as in the following example:

```
da10:~ # cat /etc/cups/printers.conf
# Printer configuration file for CUPS v1.1.23
# Written by cupsd on Thu Mar 30 16:39:17 2006
<DefaultPrinter draft>
Info Laserjet 4050TN
Location Office Training Services
DeviceURI socket://muc-hp4050TN-3.muc.novell.com:9100
State Idle
Accepting Yes
JobSheets none none
QuotaPeriod 0
PageLimit 0
KLimit 0
</Printer>
<Printer color>
Info HEWLETT-PACKARD DESKJET 880C
Location Parallel printer on /dev/lp0
DeviceURI parallel:/dev/lp0
State Idle
Accepting Yes
JobSheets none none
QuotaPeriod 0
PageLimit 0
KLimit 0
</Printer>
<Printer grayscale>
Info HEWLETT-PACKARD DESKJET 880C
Location Parallel printer on /dev/lp0
DeviceURI parallel:/dev/lp0
State Idle
Accepting Yes
JobSheets none none
QuotaPeriod 0
PageLimit 0
KLimit 0
</Printer>
...
```

For instance, in the case of color printers, it is useful to have at least two queues, one for black-and-white printing of text documents and one for color printing.

The following explains some entries in `/etc/cups/printers.conf`:

- **<DefaultPrinter *queuename*>**. The entry for the default printer.
- **<Printer color>** and **<DefaultPrinter grayscale>**. The queues as defined for the printer “HEWLETT-PACKARD DESKJET 880C”.
- **State Idle**. Currently, there is no print job in this queue.
- **Accepting Yes**. The queue is accepting print jobs.
- **JobSheets none none**. No starting banner and no ending banner will be printed.

Each existing queue has its own configuration file, which is stored on the print server in the directory

`/etc/cups/ppd/`

These files contain entries to configure the paper size, the resolution, and other settings.

By contrast, on the client side the names of queues are registered in the file **`/etc/printcap`**:

```
da10:~ # cat /etc/printcap
# This file was automatically generated by cupsd(8) from
# the /etc/cups/printers.conf file. All changes to this
# file will be lost.
draft|Laserjet 4050TN:rm=da10:rp=draft:
color|HEWLETT-PACKARD DESKJET 880C:rm=da10:rp=color:
grayscale|HEWLETT-PACKARD DESKJET 880C:rm=da10:
                                     rp=grayscale:
printer|Laserjet 4050TN:rm=da10:rp=printer:
```

In fact `/etc/printcap` is a link to `/etc/cups/printcap`. This file is generated and updated automatically by `cupsd` and is relevant for a number of applications (such as OpenOffice.org) that use the entries in it to list the available printers in their printer selection dialogs.

You should not change the file `/etc/printcap` manually.

Log Files

The log files of CUPS are stored in the directory

`/var/log/cups/`

There are three files:

- The `access_log` File
- The `error_log` File
- The `page_log` File

For troubleshooting CUPS issues:

- Set the Log Level to Record Errors

The access_log File

The `access_log` file lists each HTTP resource that is accessed by a web browser or CUPS/IPP client.

Lines in the log file looks like this:

```
localhost - - [31/Mar/2006:09:48:47 +0200] "POST / HTTP/1.1" 200 132
localhost - - [31/Mar/2006:09:48:47 +0200] "POST / HTTP/1.1" 200 132
localhost - - [31/Mar/2006:09:48:47 +0200] "POST / HTTP/1.1" 200 72
localhost - - [31/Mar/2006:09:48:47 +0200] "POST /printers/grayscale
HTTP/1.1" 200 799
```

The parts of a line are (from left to right):

- The **host** field (in the example, `localhost`).
- The **group** field always contains "-" in CUPS.

- The **user** field is the authenticated user name of the requesting user.

If a user name and password are not supplied for the request, this field contains "-".

- The **date-time** field (in this example: [31/Mar/2006:09:48:47+0200]) shows the date and time of the request in local time.

The format is: **[DD/MON/YYYY:HH:MM:SS +ZZZZ]** where **ZZZZ** is the time zone offset in hours and minutes from coordinated universal time (UTC).

- The **method** field is the HTTP method used (such as “GET”, “PUT”, and “POST”)

- The **resource** field is the filename of the requested resource. Possible resources are

- /
- /admin/
- /printers/
- /jobs/

- The **version** field is the HTTP version used by the client.

For CUPS clients, this is always “HTTP/1.1”.

- The **status** field contains the HTTP result status of the request.

Usually it is “200”, but other HTTP status codes are possible. For example, “401” indicates unauthorized access.

- The **bytes** field contains the number of bytes in the request.

For POST requests, the bytes field contains the number of bytes that were received from the client.

The error_log File

The error_log file lists messages from the scheduler (such as errors and warnings):

```
I [31/Mar/2006:09:48:47 +0200] Adding start banner page "none" to job 16.
I [31/Mar/2006:09:48:47 +0200] Adding end banner page "none" to job 16.
I [31/Mar/2006:09:48:47 +0200] Job 16 queued on 'grayscale' by 'root'.
I [31/Mar/2006:09:48:47 +0200] Started filter
/usr/lib/cups/filter/texttops (PID 4088) for job 16.
I [31/Mar/2006:09:48:47 +0200] Started filter /usr/lib/cups/filter/pstops
(PID 4089) for job 16.
I [31/Mar/2006:09:48:47 +0200] Started filter
/usr/lib/cups/filter/foomatic-rip (PID 4090) for job 16.
I [31/Mar/2006:09:48:47 +0200] Started backend
/usr/lib/cups/backend/parallel (PID 4091) for job 16.
```

The following explains the entries in the lines (from left to right):

- The **level** field contains the type of message:
 - **E.** An error occurred.
 - **W.** The server was unable to perform an action.
 - **I.** Informational message.
 - **D.** Debugging message.
- The **date-time** field contains the date and time of the entry, for instance when a page started printing.
 The format of this field is identical to the data-time field in the access_log file.
- The **message** field contains a free-form textual message.

The page_log File

The page_log file lists each page that is sent to a printer.

```
grayscale root 16 [31/Mar/2006:09:48:57 +0200] 1 1 - localhost
```

Each line contains the following information (from left to right):

- The **printer** field contains the name of the printer that printed the page (in this example, grayscale).

If you send a job to a printer class, this field contains the name of the printer that was assigned the job.

- The **user** field contains the name of the user that submitted this file for printing.
- The **job-id** field contains the job number of the page being printed (in this example, 16).
- The **date-time** field contains the date and time the page started printing.

The format of this field is identical to the date-time field in the access_log file.

- The **page-number** field contain the number of pages (in this example, 1).
- The **num-pages** field contains the number of copies (in this example, 1).

For printers that cannot produce copies on their own, the num-pages field will always be 1.

- The **job-billing** field contains a copy of the job-billing attribute provided with the IPP create-job or print-job requests or "-" if none was provided.
- The **hostname** field contains the name of the host that originated the print job (in this example localhost).

Set the Log Level to Record Errors

Messages from cupsd are written to the file /var/log/cups/error_log. With the default log level **info**, only requests and status changes are logged to the file.

If you want errors recorded, you need to change the LogLevel option in the cupsd configuration file /etc/cups/cupsd.conf:

```
# LogLevel: controls the number of messages logged to the ErrorLog
# file and can be one of the following:
#
#   debug2    Log everything.
#   debug     Log almost everything.
#   info      Log all requests and state changes.
#   warn      Log errors and warnings.
#   error     Log only errors.
#   none      Log nothing.
#
LogLevel debug2
```

For debugging and troubleshooting, set the log level to **debug2**. After changing the configuration, restart CUPS by entering **rc cups restart**.

Configuration File

The configuration file for CUPS is /etc/cups/cupsd.conf. It has a similar format as the configuration file for the Apache web server.

Various options are used to configure the server itself, filtering, networking aspects, browsing, and access.

Networking, browsing, and access are covered in the next objective.

Objective 4 **Configure and Manage a Print Server**

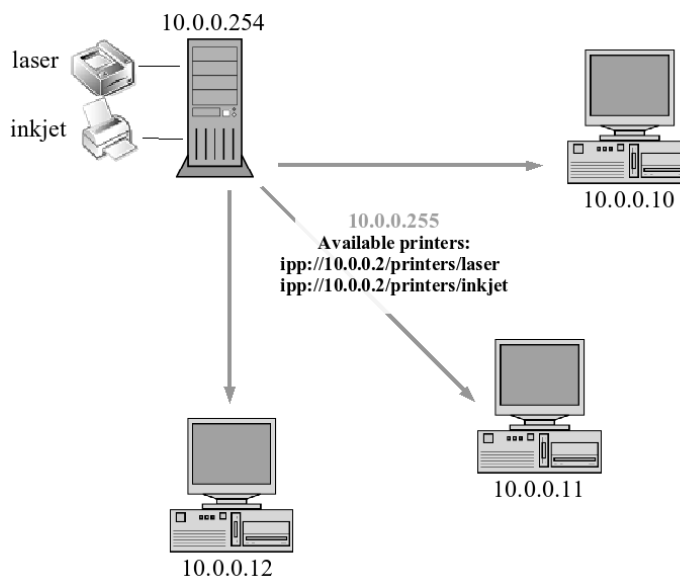
In objective 1, “Configure Local Printing” on page 10-2, you configured printing to work on a local machine. In this objective, you will learn how to control who may use your printer via the network. To be able to do this, you need to understand the following:

- Broadcast Information about Printers to other Computers
- Access Restrictions
- Restrict Access to Printers for Users and Groups
- Restrict Access to the Web Interface

Broadcast Information about Printers to other Computers

CUPS can distribute information about the available printers to all network clients by means of the browsing feature.

Figure 10-18

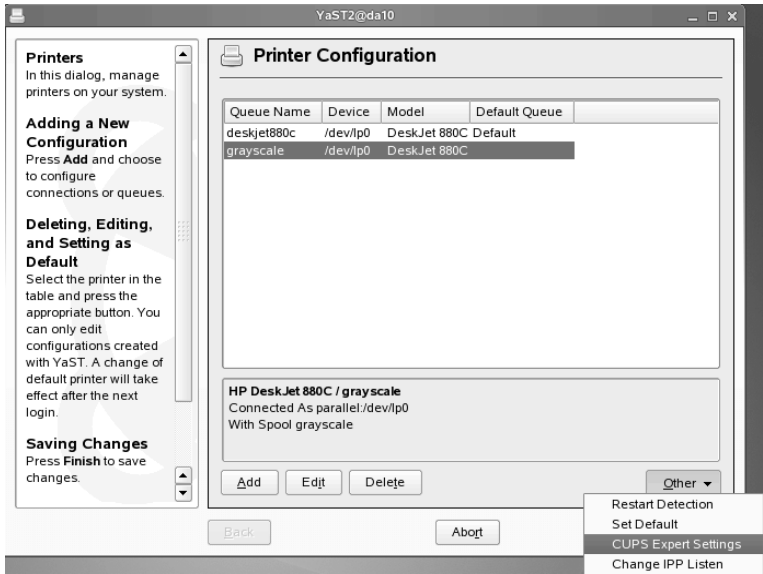


The CUPS server uses broadcast to distribute the printer information. If this function is enabled, the server broadcasts the printer information every 30 seconds. This printer information typically uses only 80 bytes per printer. You can add a large number of servers and printers.

You can configure this either via YaST or directly in the CUPS configuration `/etc/cups/cupsd.conf`.

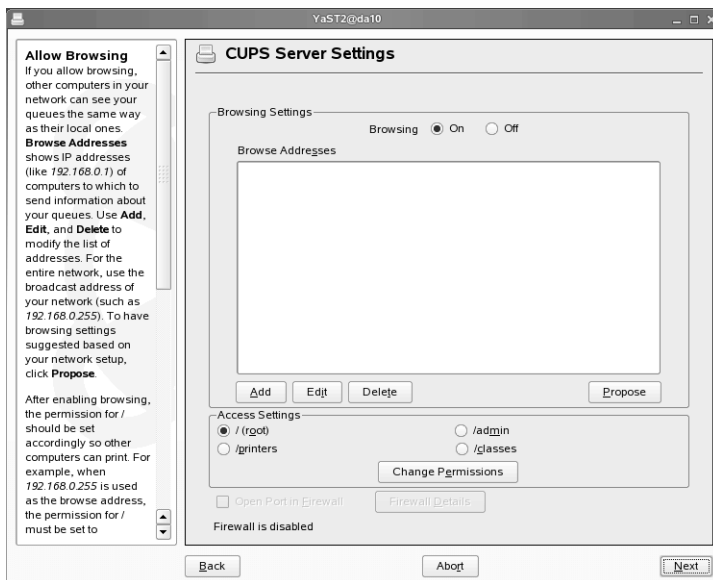
In the YaST Control Center, start the printer configuration **Hardware > Printer** and then select **Other > CUPS Expert Settings**, as in the following:

Figure 10-19



In the dialog that opens, choose **CUPS Server Settings > Next**. This opens the following dialog:

Figure 10-20



By default, browsing is turned on, meaning that the CUPS server will advertise its queues in the network. However, as no browse address is specified yet, browse information is not sent.

To activate browsing, add an IP address, a network/netmask combination or certain keywords that refer, for instance, to the local network. The following excerpt from `/etc/cups/cupsd.conf` shows what you could enter:

```
# BrowseAddress: specifies a broadcast address to be used.  By
# default browsing information is not sent!
#
# Note: Using the "global" broadcast address (255.255.255.255) will
# activate a Linux demand-dial link with the default configuration.
# If you have a LAN as well as the dial-up link, use the LAN's
# broadcast address.
#
# The @LOCAL address broadcasts to all non point-to-point interfaces.
# For example, if you have a LAN and a dial-up link, @LOCAL would
# send printer updates to the LAN but not to the dial-up link.
# Similarly, the @IF(name) address sends to the named network
# interface, e.g. @IF(eth0) under Linux.  Interfaces are refreshed
# automatically (no more than once every 60 seconds), so they can
# be used on dynamically-configured interfaces, e.g. PPP, 802.11, etc.
#
#BrowseAddress x.y.z.255
#BrowseAddress x.y.255.255
#BrowseAddress x.255.255.255
#BrowseAddress 255.255.255.255
#BrowseAddress @LOCAL
#BrowseAddress @IF(name)
```

Usually you would either use the broadcast address of the local network, like `10.0.0.255`, or the local network interfaces, using `@LOCAL`.

Access Restrictions

You can restrict the access to various CUPS resources. The resources are displayed as directories (/printers or /jobs).

Normally, the following resources are available on the CUPS server:

- **/ (root)**. The access restrictions for this resource apply for all subsequent resources if no other restrictions are specified there.
- **/printers**. All printers or queues.
- **/classes**. Available printer classes; for example, all color printers.
- **/jobs**. Print jobs on the CUPS server.
- **/admin**. These settings concern the access to the server configuration.

These resources can be accessed in various ways, for example, with a web browser:

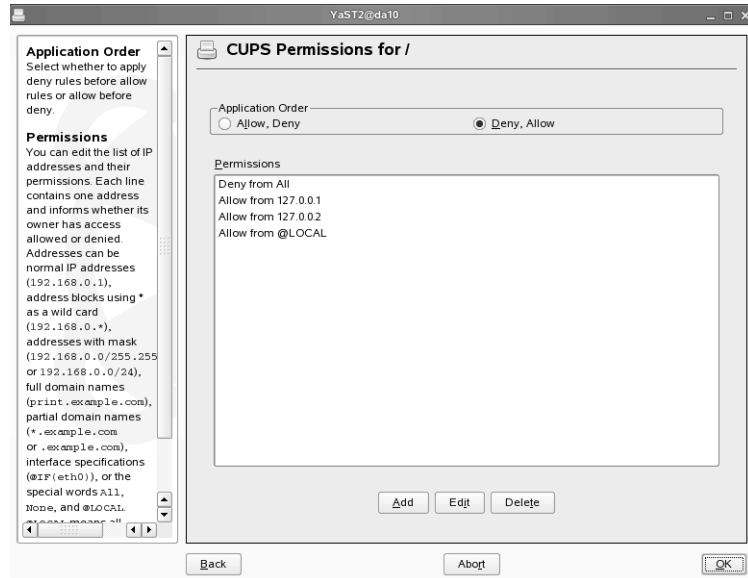
- **<http://localhost:631/printers>**
- **<http://localhost:631/admin>**

You can configure access restrictions by using YaST.

In the CUPS Server Settings dialog, where you configured browsing (Figure 10-20), select the resource and then **Change Permissions**.

You can define the order in which access directives are applied (whether to apply the allow rules first then the deny rules or vice versa), and the default directive in the next dialog.

Figure 10-21



Under Permissions, define the entities to which the access rules should be applied in line with the previously specified order.

Access settings are stored in the configuration file **/etc/cups/cupsd.conf**.



YaST writes most of these entries to the end of the file.

Here is an example:

```
<Location />
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
Allow From 127.0.0.2
Allow From @LOCAL
</Location>

<Location /admin>
...
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
</Location>

<Location /printers>
Deny From All
Allow From 10.0.0.0/24
Allow From 10.0.1.2
Order Allow,Deny
</Location>
```

The following explains the configuration directives:

- **Order.** Defines the order of the rules and the default directive:
 - Allow,Deny.** Allow requests from all systems except for those listed in a Deny directive
 - Deny,Allow.** Allow requests only from those listed in an Allow directive.
- **Deny From All.** All access to the resource is prohibited.
- **Allow From.** Access is permitted.

While the resource `/printers` concerns all queues, you can specify access restrictions on a per queue basis in additional entries that might look like this one:

```
<Location /printers/color>
Deny From All
Allow From 10.0.0.10/24
Order Deny,Allow
</Location>
```

In the above example, all clients belonging to network 10.0.0.0/24 and host 10.0.1.2 can print on all queues in the network (see previous page), with the exception of the color queue, which can only be accessed by the client 10.0.0.10.

The syntax to specify clients is the same as that used to specify browse addresses described in “Broadcast Information about Printers to other Computers” on page 10-43.

Restrict Access to Printers for Users and Groups

You can restrict access to the printers on a user and group basis.

You can configure this by using YaST by selecting a queue in the main ***Printer Configuration*** dialog, and then selecting **Edit > Restrictions Settings**.

You have to choose one of the following:

- **All Users Can Use This Printer**
- **The Following Users Can Use This Printer** (select **Add** to add users (like *tux*) or groups (like *@users*)
- **The Following Users Cannot Use This Printer** (select **Add** to add users and groups as above)

On the command line you can use the command **lpadmin**:

- To permit printing for individual users, enter

lpadmin -p *queue* -u allow:user1, user2

or for a group, enter

lpadmin -p *queue* -u allow:@users



These are not added to the existing users, but replace them.

- To prohibit printing for users or groups, enter

lpadmin -p *queue* -u deny:geeko,@guests

- To permit printing for all, enter

lpadmin -p *queue* -u allow:all or

lpadmin -p *queue* -u deny:none

These access restrictions are written to the `/etc/cups/printers.conf` file, as in the following:

```
<Printer printer>
...
AllowUser user1
AllowUser user2
AllowUser @users
</Printer>
```

Restrict Access to the Web Interface

You can protect resources such as the administration interface by using passwords.

The default configuration entry in `/etc/cups/cupsd.conf` in SLES 10 looks like the following:

```
<Location /admin>
AuthType BasicDigest
AuthClass Group
AuthGroupName sys
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
</Location>
```

The relevant directives are

- **AuthType BasicDigest.** A special CUPS user database (`/etc/cups/passwd.md5`) is used for authentication.

The following types are possible:

- **None.** No authentication should be performed.
- **Basic.** Basic authentication should be performed using the UNIX password and group files.

Note: This authentication type does not work on SLES 10 due to the fact that cupsd runs as user `lp`, and therefore cannot access `/etc/shadow`.

- **Digest.** Digest authentication should be performed using the `/etc/cups/passwd.md5` file.
- **BasicDigest.** Basic authentication should be performed using the `/etc/cups/passwd.md5` file.
- **AuthClass Group.** Access is only possible for valid users who are members of the system (`sys`) group.
- **AuthGroupName.** Name of the system group (in this example, `sys`).

With the above default configuration, CUPS accesses the user database `/etc/cups/passwd.md5`.

This file does not exist by default and is created when the first user is generated using **lppasswd**:

```
da10:~ # lppasswd -a root -g sys
Enter password:
Enter password again:
```

This command creates the user `root` in the group `sys`. Any user name will do, as long it is member of the group `sys`. The user name does not have to exist as a Linux user name.



The password has to be at least six characters long and must contain at least one letter and one number.

Exercise 10-3 Restrict Access

In this exercise, you learn how to administer access to your CUPS server.

You will find this exercise in the workbook.

(End of Exercise)

Objective 5 Use the Web Interface to Manage a CUPS Server

You can access the web interface of the CUPS server by using the URL

`http://IP_Address:631`

The main menu is shown in the following figure.

Figure 10-22



The navigation bar at the top is available on all pages, so it is not necessary to return to the main page to get to the other sections.

To manage printers and jobs or to modify the current settings, you have to authenticate as an administrator of the CUPS server.

By default, no administrator for CUPS is defined. Enabling administrative access using the web interface, is described in “Restrict Access to the Web Interface” on page 10-52.

In the main menu, the following sections are available:

- Do Administration Tasks

- Manage Printer Classes
- On-Line Help
- Manage Jobs
- Manage Printers

Do Administration Tasks

In the Administration module (<http://localhost:631/admin>) you can perform most administration tasks:

- **Add Class.** Add a printer class.
- **Manage Classes.** Edit or delete printer classes.
- **Manage Jobs.** View and manage print jobs.
- **Add Printer.** Configure a new printer.
- **Manage Printers.** Edit or delete a printer configuration.

Figure 10-23



In most cases, you are supported by a wizard and you do not have to enter a lot of information.

Manage Printer Classes

In the Classes module (<http://localhost:631/classes>) you can add printer classes.

If there is at least one class configured, you can also select **Manage Classes** from this module.

Figure 10-24



You are supported by a wizard.

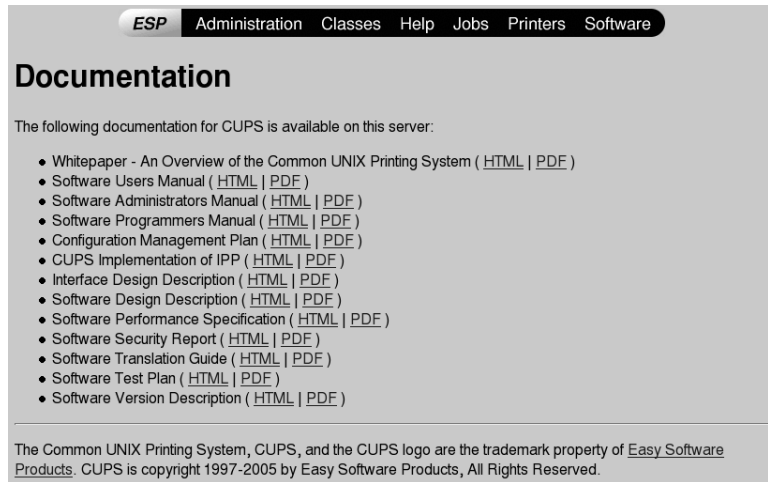
The configuration dialog is the same as the dialog you get when you select **Add Class** or **Manage Classes** in the Administration module.

On-Line Help

There is a lot of documentation installed with the CUPS packages.

You can access them in HTML or PDF format.

Figure 10-25



Manage Jobs

In the Jobs module (<http://localhost:631/jobs>) you can switch between the view of the completed jobs or the view of the active jobs.

To switch between the two views, select **Show Completed View** or **Show Active Jobs**.

If there is one (or more) job in the queue, you can also

- Hold the job.

- Cancel the job.

Figure 10-26



The management dialog is the same as the dialog you get when you select **Manage Jobs** in the Administration interface.

Manage Printers

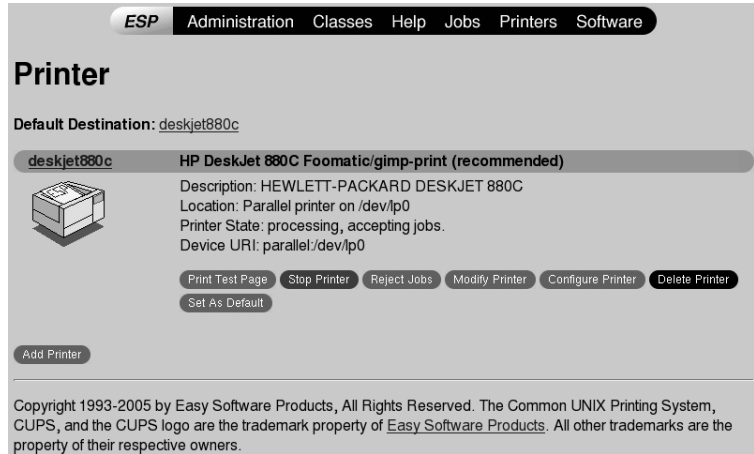
In the Printer module (<http://localhost:631/printers>), you can add a printer.

If there is at least one printer configured, you can also

- Print a test page.
- Stop the printer.
- Reject print jobs.
- Modify the printer configuration.
- Configure the printer (paper size, resolution, and banner).

- Delete the printer configuration.

Figure 10-27



The configuration dialog is the same as the dialog you get when you select **Add Printer** or **Manage Printers** in the Administration module.

Exercise 10-4 *Use the Web Interface to Manage a CUPS Server*

In this exercise you will add a second printer via the web frontend of CUPS, even though a second printer is not available at your workstation.

You will find this exercise in the workbook.

(End of Exercise)

Summary

Objective	Summary
1. Configure Local Printing	<p>CUPS, the Common Unix Printing System is the default printing system used in SLES 10.</p> <p>CUPS can be used to print on local or remote printers. The protocol used is IPP, but various other protocols are supported as well.</p> <p>Printers are addressed by using print queues.</p> <p>The YaST module to configure CUPS can be found at</p> <p>YaST Control Center > Hardware > Printer</p> <p>Any packages needed for the client to function properly are automatically installed by YaST.</p>

Objective	Summary
2. Manage Print Jobs and Queues	<p>The CUPS tools allow you to use commands according to</p> <ul style="list-style-type: none">■ Berkeley style (identical to commands used with the LPRng printing system)■ System V style <p>To generate a print job, enter</p> <ul style="list-style-type: none">■ Berkeley: lpr -P <i>queue file</i>■ System V: lp -d <i>queue file</i> <p>To display information on print jobs, enter</p> <ul style="list-style-type: none">■ Berkeley: lpq -P <i>queue</i>■ System V: lpstat -o <i>queue -p queue</i> <p>To cancel print jobs, enter</p> <ul style="list-style-type: none">■ Berkeley: lprm -P <i>queue jobnumber</i>■ System V: cancel <i>queue-jobnumber</i> <p>Users can obtain the current settings of a local queue by entering</p> <p>lpoptions -p <i>queue -l</i></p>

Objective	Summary
3. Understand How CUPS Works	<p>The main configuration file for CUPS is /etc/cups/cupsd.conf.</p> <p>Information on the print queues is kept in</p> <p>/etc/cups/printers.conf</p> <p>A configuration file for each queue is located in the directory</p> <p>/etc/cups/ppd/</p> <p>These files store settings affecting the printout through the given queue.</p> <p>The file /etc/printcap, which is created and updated automatically, contains an entry for each of the defined queues.</p> <p>CUPS can distribute information about the available printers to all network clients.</p>
4. Configure and Manage a Print Server	<p>A CUPS server can distribute information about the available queues within the network (browsing).</p> <p>Access to resources on the CUPS server can be restricted based on IP addresses, users, groups, or passwords.</p>

Objective	Summary
5. Use the Web Interface to Manage a CUPS Server	<p>You can enter the CUPS web frontend at</p> <p>http://localhost:631</p> <p>or</p> <p>http://IP_Address:631</p> <p>In SLES 10, to enable management via the web frontend, a user (usually root) must be designated as the CUPS administrator and be a member of the CUPS administration group sys.</p> <p>This can be done by entering</p> <p>lppasswd -a root -g sys</p> <p>This command sets the password for the access.</p> <p>The configuration is done mostly with the help of wizards.</p>

CNI USE ONLY-1 HARDCOPY PERMITTED

SECTION 11 Configure Remote Access

In this section you learn how to configure your SUSE Linux Enterprise Server to provide remote access for users and to perform administrative tasks remotely.

Objectives

1. Provide Secure Remote Access with OpenSSH
2. Enable Remote Administration with YaST

Objective 1 **Provide Secure Remote Access with OpenSSH**

In the past, remote connections were established with Telnet, which offers no guards in the form of encryption or other security mechanisms against eavesdropping. There are also other traditional communication channels (such as FTP and some remote copying programs) that provide unencrypted transmission.

The SSH suite was developed to provide secure transmission by encrypting the authentication strings (usually a login name and a password) and all the other data exchanged between the hosts.

With SSH, the data flow can still be recorded by a third party, but the contents are encrypted and cannot be reverted to plain text unless the encryption key is known.

SUSE Linux Enterprise Server 10 installs the package OpenSSH by default, which includes programs such as `ssh`, `scp`, and `sftp` as alternatives to Telnet, `rlogin`, `rsh`, `rcp`, and FTP.

To provide secure remote access on a network with the OpenSSH version of SSH, you need to know the following:

- Cryptography Basics
- SSH Features and Architecture
- Configure the SSH Server
- Configure the SSH Client
- SSH-related Commands
- Public Key Authentication Management

Cryptography Basics

Cryptography deals with procedures and techniques used to encrypt data and prove the authenticity of data. An encryption algorithm is used to convert clear text into cipher text using a key. The key is the information required to encrypt and decrypt data.

There are basically two types of encryption procedures:

- Symmetric Encryption
- Asymmetric Encryption

Symmetric Encryption

With symmetric encryption, the same key is used for encryption and decryption. If this secret key is known, then all data encrypted with that key can be decrypted.

An important feature of an encryption procedure is the length of the key. A symmetric key with a length of 40 bits (109951162776 possibilities) can be broken with brute force methods in a short time. Currently, symmetric keys with 128 bits or more are considered secure.

In other words, the longer the key length, the more secure the data transmission, provided there is no cryptographic flaw in the encryption algorithm.

The following are some of the more important symmetric encryption technologies:

- **DES (Data Encryption Standard).** DES was standardized in 1977 and is the foundation of many encryption procedures (such as UNIX/Linux passwords). The key length is 56 bits.

However, in January 1999 the EFF (Electronic Frontier Foundation) decrypted a text encrypted with DES in 22 hours using brute force (trying one possible key after the other). Therefore, a key with a length of 56 bits is no longer secure, as messages protected with such a key can be decrypted in a short time.

- **Triple-DES.** Triple DES is an extension of DES, using DES three times. Depending on the variant used, the effective key length offered is 112 or 168 bits.
- **IDEA.** IDEA is an algorithm with a key length of 128 bits. This algorithm has been patented in the USA and Europe (its noncommercial use is free).
- **Blowfish.** This algorithm has a variable key length of up to 448 bits. It was developed by Bruce Schneier; it is unpatented and license-free, it can be freely used by anyone.
- **AES (Advanced Encryption Standard).** AES is the successor to DES.

In 1993 the National Institute of Standards and Technology (NIST) decided that DES no longer met today's security requirements, and organized a competition for a new standard encryption algorithm. The winner of this competition was announced on October 2, 2000, and is the Rijndael algorithm which supports key lengths of 128, 192 or 256 bits.

The advantage of symmetric encryption is the fact that it can efficiently encrypt and decrypt data. Its disadvantages are the difficulties associated with key distribution and management.

Asymmetric Encryption

In an asymmetric encryption there are two keys—a private key and a public key. Data that is encrypted with the private key can only be decrypted with the public key, and data encrypted with the public key can only be decrypted with the private key.

The main advantage of asymmetric encryption is the fact that the public key can be distributed freely. A disadvantage of asymmetric procedures is their low speed in data processing. Symmetric procedures are much faster.

Therefore symmetric and asymmetric procedures are often combined. For example, a key for symmetric encryption is transmitted through a channel encrypted asymmetrically. SSH uses a combination of both procedures.

Some important cryptographic procedures in this context are

- **RSA.** The name is derived from the surnames of its developers, Rivest, Shamir, and Adleman. Its security is mainly based on the fact that it is easy to multiply two large prime numbers, but it is difficult to regain the factors from this product.
- **DSA.** Digital Signature Algorithm. It is a US Federal Government standard for digital signatures.
- **Diffie-Hellman.** The Diffie-Hellman key exchange describes a method to establish cryptographic keys securely without having to send the keys across insecure channels. Such a key can then be used as a secret key in symmetric encryption.

Keys for asymmetric encryption are much longer than those used for symmetric procedures. For instance with RSA, the minimum key length currently considered secure is 1024 bit.

SSH Features and Architecture

To understand what SSH can offer as a secure, remote transmission protocol, you need to know the following:

- SSH Features
- SSH Protocol Versions
- SSH Authentication Mechanism Configuration

SSH Features

The secure shell not only provides all the functionality of Telnet, rlogin, rsh and rcp, but even includes some features of FTP.

SSH supports the protection of X11 and any TCP connections, by routing them through a cryptographically secure channel.

The following lists the basic functionality provided by SSH:

- Login from a remote host.
- Interactive or noninteractive command execution on remote hosts.
- Copying files between different network hosts; optional support for compressing data.
- Cryptographically secured authentication and communication across insecure networks.
- Automatic and transparent encryption of all communication.
- Complete substitution of the “r” utilities: rlogin, rsh, and rcp.
- Port forwarding.
- Tunneling.

SSH not only encrypts the traffic and authenticates the client, it also authenticates the involved servers. Various procedures are available for server authentication.

CNI USE ONLY-1 HARDCOPY PERMITTED

In SUSE Linux Enterprise Server the Open Source implementation of SSH (OpenSSH) is used. OpenSSH is available as open source because it does not use any patented algorithms.

By default, the OpenSSH server is not activated when you install SUSE Linux Enterprise Server 10, but it can be easily activated during or after installation.



For more details on OpenSSH functionality, see <http://www.openssh.org>.

SSH Protocol Versions

The following are the versions currently available for the SSH protocol:

- Protocol Version 1 (SSH1)
- Protocol Version 2 (SSH2)

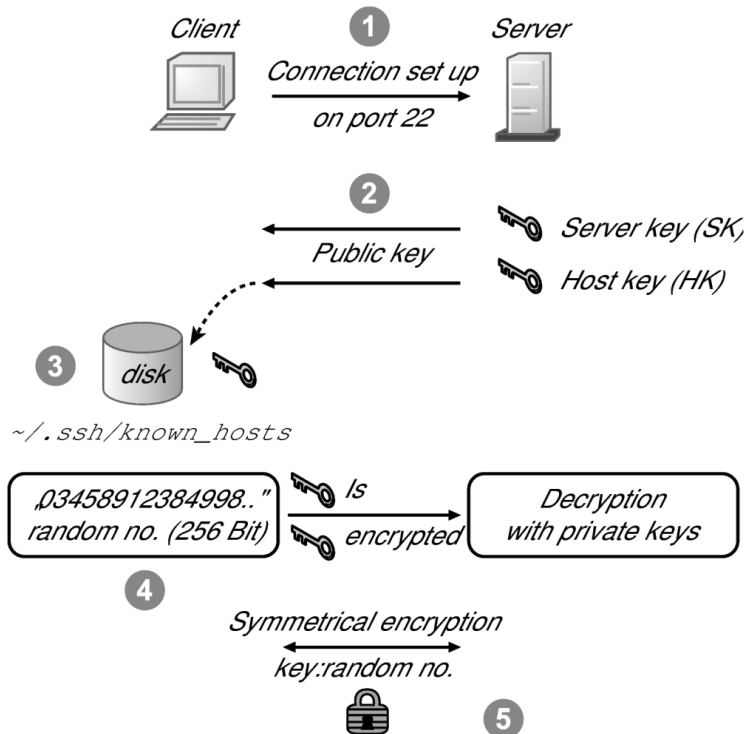


SSH1 and SSH2 are used for convenience in referencing the protocol versions in this section. They are not official designations of the protocol versions.

Protocol Version 1 (SSH1)

The following illustrates the process SSH1 uses to transmit data over a secure connection:

Figure 11-1



The following describes the steps in this process:

1. The client establishes a connection to the server (port 22).
In this phase the SSH client and the server agree on the protocol version and other communication parameters.
2. The SSH server works with the following two RSA key pairs and transmits the public keys to the client:

- ❑ **Long-life host key pair (HK).** This key pair consists of a public host key (/etc/ssh/ssh_host_key.pub) and a private host key (/etc/ssh/ssh_host_key) that identify the computer.

This long-life key pair is identical for all SSH processes running on the host.

- ❑ **Server process key pair (SK).** This key pair is created at the start of each server process that includes a public server key and a private server key that are changed at specific intervals (normally once an hour).

This pair is never stored in a file. These dynamic keys help prevent an attacker from being able to decrypt recorded sessions, even if the attacker can break into the server and steal the long-life key pair.

3. The client checks to see if the public host key is correct.

To do this, it compares the host key with keys in the file /etc/ssh/ssh_known_hosts or ~/.ssh/known_hosts. If these files do not contain the key, depending on the configuration, the connection is terminated or the user is asked how to proceed.

4. The client generates a 256-bit random number, encrypts this using the public keys of the SSH server and sends it to the server.
5. The server is now in a position to decrypt the random number, because it possesses the secret key.
6. This random number is the key for the symmetric encryption that now follows.

The random number is also referred to as the *session key*.

When the user now types his password, it is protected by the encrypted connection.

Protocol Version 2 (SSH2)

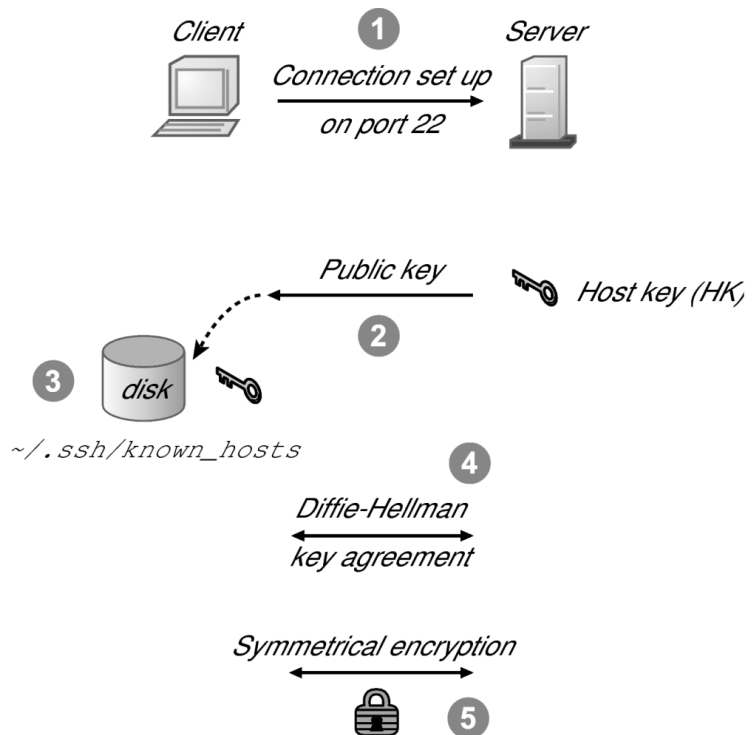
SSH protocol version 1 does not have a mechanism to ensure the integrity of a connection. This allows attackers to insert data packets into an existing connection (an insertion attack).

SSH2 provides features to avoid such attacks. These are referred to as HMAC (Keyed-Hash Message Authentication Code) and are described in detail in RFC 2104.

You should only use SSH1 if SSH2 is not available.

The following illustrates the process SSH2 uses to transmit data over a secure connection:

Figure 11-2



The following describes the steps in this process:

1. A connection is established between the server and client as described for SSH1.
2. The server now contains a key pair (DSA or RSA), the public and private host key.

The private key files are /etc/ssh/ssh_host_rsa_key (RSA) and /etc/ssh/ssh_host_dsa_key (DSA), respectively.
3. As with SSH1, the host key is compared with the keys in the files /etc/ssh/ssh_known_hosts and ~/.ssh/known_hosts.
4. A Diffie-Hellman key agreement then follows, through which client and server agree on a secret session key, without having to send the key across the wire.
5. As with SSH1, communication is ultimately encrypted symmetrically.

The basic difference between SSH1 and SSH2 are mechanisms within the protocol that guarantee the integrity of the connection. A keyed-hash message authentication code (HMAC) is used for this purpose, The mechanism for the session key agreement (Diffie-Hellman) is different as well.

To see which SSH version an SSH server supports, you can log on to port 22 with Telnet. The following shows the potential responses from the server:

Table 11-1

Protocol	Server Response
SSH1 only	SSH-1.5-OpenSSH...
SSH1 and SSH2	SSH 1.99-OpenSSH...
SSH2 only	SSH-2.0-OpenSSH...

The following is an example of a Telnet connection on port 22:

```
da10:~ # telnet da20 22
Trying 10.0.0.20...
Connected to da20.
Escape character is '^]'.
SSH-1.99-OpenSSH_4.2
```

In the server configuration file `/etc/ssh/sshd_config`, the `Protocol` parameter defines which protocol versions are supported.

For example, **Protocol 2,1** in the configuration file means that SSH2 and SSH1 are both supported, but preference is given to SSH2. If SSH2 is not available, then SSH1 is used.

You can also specify the version to use when starting the clients (such as **ssh -1** for SSH1).

SSH Authentication Mechanism Configuration

The SSH server can decrypt the session key generated and encrypted by the client only if it also has the private key. If the server does not do this, the communication ends at that point.

An absolute condition for the security of this procedure is that the client can check if the public host key of the server really belongs to the server.

SSH currently does not use any directory services (such as LDAP) or any certificates (such as with SSL) for public key management. This means that a random key pair can be easily created by anyone, even potential attackers, and included in the authentication dialog.

When first contacting an unknown server, it is possible to “learn” its host key. In this case, the SSH client then writes this key to the local key database.

The following is an example of an SSH connection to a computer whose host key is unknown:

```
geeko@da50:~ > ssh geeko@da10
The authenticity of host 'da10 (10.0.0.10)' can't be established.
RSA key fingerprint is ea:79:90:9a:d4:bf:b6:a2:40:ee:72:56:f8:d9:e5:76.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'da10,10.0.0.10' (RSA) to the list of known
hosts.
```

If you answer the question with “yes,” the host key is saved in the file `~/.ssh/known_hosts`.

Several mechanisms are available on the server side to authenticate clients. The mechanisms allowed by the server are specified in its configuration file `/etc/ssh/sshd_config`.

The following describes the two most important mechanisms with the appropriate configuration parameters for `/etc/ssh/sshd_config` in parentheses:

- **Public Key (RSA/DSA) Authentication**

(`sshd_config`: `RSAAuthentication` for SSH1)

(`sshd_config`: `PubkeyAuthentication` for SSH2)

Authentication through a public key procedure is the most secure method. In this case, the user proves knowledge of her private key (and thus her identity) through a challenge-response procedure, which can be run automatically using the SSH agent.

- **Password Authentication**

(`ssh_config`: `PasswordAuthentication`)

This authentication procedure takes place through a UNIX user password. The transfer of the password is encrypted.

After successful authentication, a work environment is created on the server. For this purpose, environment variables are set (TERM and DISPLAY), and X11 connections and any possible TCP connections are redirected.



The redirection of the X11 connections only works if the DISPLAY variable set by SSH is not subsequently changed by the user. The SSH daemon must appear to the X11 applications as a local X11 server, which requires a corresponding setting of DISPLAY.

In addition the program xauth (used to edit and display the authorization information used in connecting to the X server) must exist. This program is in the package xf86.

The parameter X11Forwarding in the configuration file of the SSH server (/etc/ssh/sshd_config) determines whether or not the graphical output is forwarded when the client requests it.

If you want to use X forwarding, you must set the parameter to Yes, and you must start the SSH client with the option -X.


Configure the SSH Server

The configuration file for the server is /etc/ssh/sshd_config. Some of the more commonly used options include the following:

Table 11-2

Option	Description
AllowUsers	Allows SSH login only for users listed, separated by spaces
DenyUsers	Denies SSH login to users listed, separated by spaces
Protocol	Specifies the protocol versions supported. (Default: 2,1)

Table 11-2 (continued)

Option	Description
ListenAddress	Specifies the local addresses sshd should listen on, <i>IP_address:port</i>
Port	Specifies the port number that sshd listens on. The default is 22. Multiple options of this type are permitted.
PasswordAuthentication	Specifies whether password authentication is allowed. If you want to disable it, set this to no and also UsePAM to no.
UsePAM	Enables the Pluggable Authentication Module interface.
 For additional information on SSH server configuration options, enter man sshd , and man sshd_config .	

Configure the SSH Client

You configure the SSH client by editing the file `/etc/ssh/ssh_config`. Each user can edit his individual settings in the file `~/.ssh/config`.

If a user wants to ensure that only servers are accepted whose keys have been previously added to `~/.ssh/known_hosts` or `/etc/ssh/ssh_known_hosts`, she should set the option `StrictHostKeyChecking` in the client configuration file (`~/.ssh/config`) to **yes**.

This prevents the SSH client from simply adding new keys from unknown servers to `~/.ssh/known_hosts` when connecting to unknown servers. Any new keys have to be added manually using an editor. Connections to servers whose key has changed are refused.

From SSH version 1.2.20 on, three values are allowed for StrictHostKeyChecking: yes, no, and ask. The default setting is ask, which means that before a new key is entered, the user is asked for permission.

The precedence of configuration options is as follows:

- 1. Command line options
- 2. ~/.ssh/config
- 3. /etc/ssh/ssh_config



For additional information on SSH client configuration options, enter **man ssh_config**.

SSH-related Commands

The following are commonly used SSH-related client commands:

Table 11-3

Command	Description
ssh	This is the SSH client. SSH can be a replacement for rlogin, rsh, and Telnet. sslogin is a symbolic link to ssh . Every user should use ssh consistently instead of Telnet.
scp	This command copies files securely between two computers using ssh, and replaces rcp and FTP (for pure file transfer).
sftp	This command offers an interface similar to a command line ftp client. You can view files on the remote machine with ls and transfer files using put and get .

Table 11-3 (continued)

Command	Description
ssh-keyscan	A utility for gathering the public ssh host keys from a number of SSH servers. The keys gathered are displayed on the standard output. This output can then be compared with the key in the file /etc/ssh/ssh_known_hosts and be included in the file.
ssh-keygen	This command generates RSA keys.
ssh-agent	This command can handle private RSA keys, to respond to challenges (challenge response) from the server. This simplifies authentication.
ssh-add	This command registers new keys with the ssh-agent.

The basic syntax for ssh is:

ssh options host command

The basic syntax for scp is:

scp options sourcefile destinationfile

The following are some examples of using the ssh and scp:

- `geeko@da10:~> ssh da20.digitalairlines.com`
In this example, the user geeko logs in to the computer **da20.digitalairlines.com** as user geeko.
- `geeko@da10:~> ssh -l tux da20.digitalairlines.com`
or
`geeko@da10:~> ssh tux@da20.digitalairlines.com`
In these examples, the user geeko on the computer da10 logs in as the user tux on the computer da20.digitalairlines.com.

- **geeko@da10:~> ssh root@da20.digitalairlines.com shutdown -h now**

In this example, the user geeko shuts down the computer da20.digitalairlines.com.

- **geeko@da10:~> scp da20.digitalairlines.com:/etc/HOSTNAME ~**

In this example, the user geeko copies the file /etc/HOSTNAME from the computer da20.digitalairlines.com to his local home directory.

- **geeko@da10:~> scp /etc/motd da20.digitalairlines.com:**

In this example, the user geeko copies the local file /etc/motd to his home directory on the computer da20.digitalairlines.com.

- **geeko@da10:~> ssh -X da20.digitalairlines.com**

In this example, the user geeko logs on to the host da20.digitalairlines.com from da10 via SSH. The connection is established with a graphical X11 tunnel, allowing X11 applications started on da20.digitalairlines.com to be displayed on da10.

- **geeko@da10:~> ssh-keyscan da50**

In this example, the host key is read from the computer da50:

```
geeko@da10:~> ssh-keyscan da50
# da50 SSH-1.99-OpenSSH_4.2
da50 1024 35
147630753138877628907212114351828387115609838536239739
003941645933217891796753690402603932260180108759131976
671875861048667320911706379693377112828949660003683832
...
geeko@da10:~> ssh-keyscan -t rsa da50
# da50 SSH-1.99-OpenSSH_4.2
da50 ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEA3Nj0qGKjyGCBhn487sMtAzyRF
q9QPK9ZcPiILSNPugTgbG9Y7+ta68JLAS+Bxp4yZGNhtw5tdnM3sRY
WCj6KbjtzjdibuVUGv9xddrg8tUH18x3y2SY48JA9YozlO57QIT3VP
p/cv5YFYPA1PttnQf0DIbpLkNlNuXTrhbfIsE=
```

CNI USE ONLY-1 HARDCOPY PERMITTED

SSH can also be used to protect unencrypted traffic, like POP3, by tunnelling it through an SSH connection. The following examples illustrate this:

- **geeko@da10:~> ssh -L 4242:da20.digitalairlines.com:110
geeko@da20.digitalairlines.com**

In this example, the user geeko forwards the connection coming in on port 4242 of his local host da10 to port 110 (POP3) of the remote host da20 via an SSH tunnel (port forwarding).

By using port forwarding through an SSH tunnel, you can set up an additional secure channel for connections between the local host and a remote host.



Privileged ports (0–1024) can only be forwarded by root.

- **geeko@da10:~> ssh -R 4242:da10.example.com:110
geeko@da20.digitalairlines.com**

With this command, you forward port queries addressed to a port of a remote host to the port of the local host (reverse port forwarding).

In this example, queries coming in on port 4242 of the remote host da20.digitalairlines.com are reverse-tunneled via SSH to port 110 of the local host da10.

If the host you want to forward to cannot be reached directly through SSH (for example, because it is located behind a firewall), you can establish a tunnel to another host running SSH, as in the following:

- **geeko@da10:~> ssh -L 4242:da20.digitalairlines.com:110
geeko@da30.digitalairlines.com**

In this example, the user geeko forwards incoming connections on port 4242 of her local host da10 to the remote host da30.digitalairlines.com by way of an SSH tunnel.

This host then forwards the packets to port 110 (POP3) of the host da20.digitalairlines.com by using an unencrypted connection.

Exercise 11-1 Practice Using OpenSSH

In this exercise, you learn how to use OpenSSH. You need to work with a partner to practice using the SSH suite of utilities.

You will find this exercise in the workbook.

(End of Exercise)

Public Key Authentication Management

Besides password authentication, a user can also authenticate using a public key procedure. Protocol version 1 only supports RSA keys. Protocol version 2 provides authentication through RSA and DSA keys.

To manage public key authentication, you need to know the following:

- Public Key Authentication Process
- Create a Key Pair
- Configure and Use Public Key Authentication

Public Key Authentication Process

To use public key authentication, the public key of the user has to be stored on the server in the home directory of the user account being accessed. These public keys are stored on the server in the file `~/.ssh/authorized_keys`. The corresponding private key must be stored on the client computer.

With the keys stored in the appropriate places, the following occurs in the public key authentication process:

1. The client informs the server which public key is being used for authentication.
2. The server checks to see if the public key is known.
3. The server encrypts a random number using the public key and transfers this to the client.
4. Only the client is able to decrypt the random number with its private key.
5. The client sends the server an MD5 checksum that it has calculated from the number.

6. The server also calculates a checksum and if they are identical, the user has authenticated successfully.
7. If public key authentication fails and password authentication is allowed, the user is asked for the login password.

The secret key should be protected by a passphrase. Without passphrase protection, simply owning the file containing the private key is sufficient for a successful authentication.

However, if the key is additionally protected with a passphrase, the file is useless if you do not know the passphrase.

Create a Key Pair

You create a key pair with the command `ssh-keygen`. A different key is required for SSH1 than for SSH2. For this reason, you need to create a separate key pair for each version.

You use the option **-t *keytype*** to specify the type of key. **ssh-keygen -t rsa1** generates a key pair for SSH1; **ssh-keygen -t rsa** or **ssh-keygen -t dsa** are used to create key pairs for ssh2.

The keys are stored in the directory `~/.ssh`. For SSH1, the default for these files is `~/.ssh/identity` (private key) and `~/.ssh/identity.pub` (public key). For SSH2 the default files are `~/.ssh/id_rsa` and `~/.ssh/id_dsa`, respectively, plus the corresponding public key files with the `.pub` extension.

The following shows how a key pair for the protocol version 2 is generated using option **-t** (required) to generate a DSA key pair:

```
geeko@da10:~> ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/geeko/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/geeko/.ssh/id_dsa.
Your public key has been saved in /home/geeko/.ssh/id_dsa.pub.
The key fingerprint is:
ef:73:c6:f6:8a:ff:9d:d1:50:01:cf:07:65:c5:54:8b geeko@da10
```

Configure and Use Public Key Authentication

For authentication using RSA or DSA keys, you need to copy the public key to the server, and then append the public key to the file `~/.ssh/authorized_keys`.

For example, you can copy the key to the server with the command **scp**, as in the following:

```
geeko@da10:~> scp .ssh/id_dsa.pub da50:geeko-pubkey
```

The key should then be added to the file `~/.ssh/authorized_keys` in such a way that the existing keys are not overwritten, as in the following:

```
geeko@da10:~> ssh da50
Password:
Last login: Tue May 30 12:03:29 2006 from da10.digitalairlines.com
geeko@da50:~> cat geeko-pubkey >> ~/.ssh/authorized_keys
geeko@da50:~> exit
geeko@da10:~>
```

You can now launch the client to see if authentication with the DSA key works properly, as in the following:

```
geeko@da10:~> ssh da50
Enter passphrase for key '/home/geeko/.ssh/id_dsa':
Last login: Tue May 30 12:03:40 2006 from da10.digitalairlines.com
geeko@da50:~>
```

You can use the option **-i** to enter the file name for a private key with a different name or location.

When authentication is done with keys, the passphrase is required when logging in to the server or when copying with scp. The ssh-agent can be used to avoid having to type this passphrase upon each connection.

When you first start the **ssh-agent**, you need to enter the passphrase. using the command **ssh-add**. After that, the ssh-agent monitors all SSH requests and provides the required private key as necessary.

The ssh-agent serves as a wrapper for any other process (such as for a shell or the X server). The following example shows the start of a bash shell through the ssh-agent:

```
geeko@da10:~> ssh-agent bash
geeko@da10:~> ssh-add .ssh/id_dsa
Enter passphrase for .ssh/id_dsa:
Identity added: .ssh/id_dsa (.ssh/id_dsa)
```

For all **ssh** or **scp** commands entered from this shell (for which a key authentication is configured), the agent will automatically provide the private key.

You can also use the ssh-agent with a graphical login. When you log in to the graphical interface, an X server is started. If you log in by using a display manager, the X server loads the file `/etc/X11/xdm/sys.xsession`.

For the ssh-agent to start automatically when an X server starts, you simply enter the following parameter in the file `sys.xsession`:

usessh="yes"

This entry is already set by default in SUSE Linux Enterprise Server.

After entering the **yes** parameter, the ssh-agent starts automatically the next time the user logs in to the graphical interface. The agent running in the background must be given the passphrase once, as in the following:

```
geeko@dal0:~> ssh-add .ssh/id_dsa
Enter passphrase for .ssh/id_dsa:
Identity added: .ssh/id_dsa (.ssh/id_dsa)
```

For subsequent connections in which authentication takes place with the public key procedure, a passphrase now no longer has to be given. This ssh-agent takes care of the private keys.

When the X server is terminated, the ssh-agent is also closed. The passphrase is never stored in a file, only the private keys are stored in memory by the ssh-agent until the user has logged out again.

Exercise 11-2 *Perform Public Key Authentication*

In this exercise, you practice using SSH with public key authentication. You need to work with a partner in this exercise.

You will find this exercise in the workbook.

(End of Exercise)

Objective 2 **Enable Remote Administration with YaST**

You can enable remote administration of your SUSE Linux Enterprise Server by using the YaST Remote Administration module.

As a matter of fact, this module activates remote access to the entire graphical environment, not just remote administration.

To implement and use this remote connection, you need to know the following:

- VNC and YaST Remote Administration
- Configure Your Server for Remote Administration
- Access Your Server for Remote Administration

A Remote Administration connection is less secure than SSH, which encrypts all data transmitted (including the password). For this reason, we recommend using the remote connection via VNC only when necessary for performing administrative tasks. SSH would be the preferred choice.

VNC and YaST Remote Administration

VNC (virtual network computing) is a client-server solution that allows a remote X server to be managed through a lightweight and easy-to-use client from anywhere on the Internet (though you should limit this to your LAN as the transmitted data are not encrypted).

The two computers don't have to be of the same type. The server and client are available for a variety of operating systems, including Microsoft Windows, Apple MacOS, and Linux.

You can use the YaST Remote Administration module to configure your SUSE Linux Enterprise Server for remote access through VNC from any network computer.

When you activate Remote Administration, xinetd offers a connection that exports the X login through VNC.

With the Remote Administration activated, you connect to the server through a VNC client such as vncviewer (connect to **hostname:5901**), through a VNC connection in Konqueror (**vnc://hostname:5901**), or through a Java-capable web browser (**http://hostname:5801**).

The hostname parameter can be the actual host name (such as **http://da10.digitalairlines.com:5801**) or the host IP address (such as **http://10.0.0.10:5801**).



For additional information on VNC, enter **man vncviewer** or see <http://www.realvnc.com>. Also refer to the documentation in `/etc/xinet.d/vnc`, or enter **netstat -patune** for a list of Internet connections to the server.

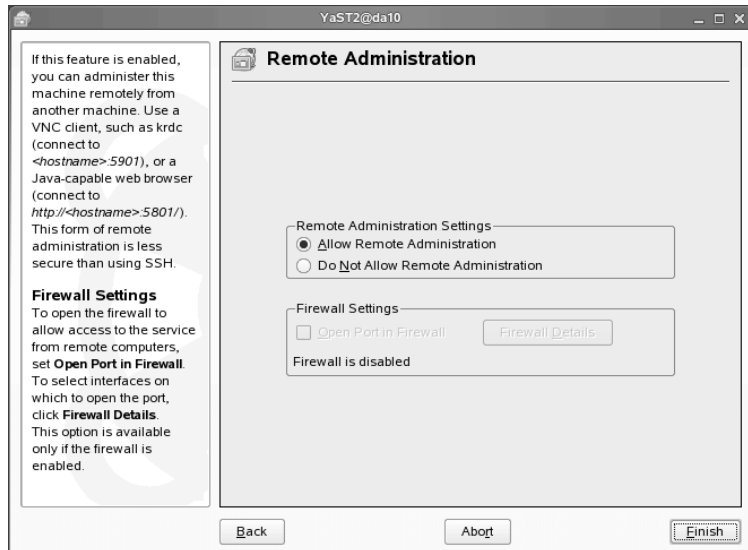
Configure Your Server for Remote Administration

To configure your SUSE Linux Enterprise Server for remote administration, do the following:

Start the YaST Remote Administration module by starting the YaST Control Center; then select **Network Services > Remote Administration**; or open a terminal window, **su -** to root and enter **yast2 remote**.

The following appears:

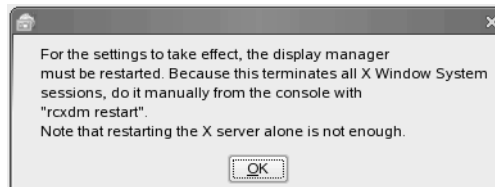
Figure 11-3



Select **Allow Remote Administration**; if your firewall is active, also select **Open Port in Firewall**; then select **Finish**.

The following appears:

Figure 11-4



Close the dialog by selecting **OK**.

As the message says, you need to restart the display manager to activate the remote administration settings. Close any open applications; then display a console pressing **Ctrl+Alt+F2**.

Log in as root with the appropriate password. Restart the display manager by entering **rcxdm restart**. After a few moments, a graphical login is displayed.

Your SUSE Linux Enterprise Server 10 is ready to be accessed remotely for administration.



You can deactivate remote administration on your SUSE Linux Enterprise Server by following the same steps but selecting **Do Not Allow Remote Administration**.

Access Your Server for Remote Administration

To access a SUSE Linux Enterprise Server that has been configured for remote administration, you can use a VNC client or a Java-enabled web browser.

To access the server from a web browser, open the web browser from the computer desktop; then enter the following:

`http://hostname:5801`

where *hostname* is the IP address or host name of the server.

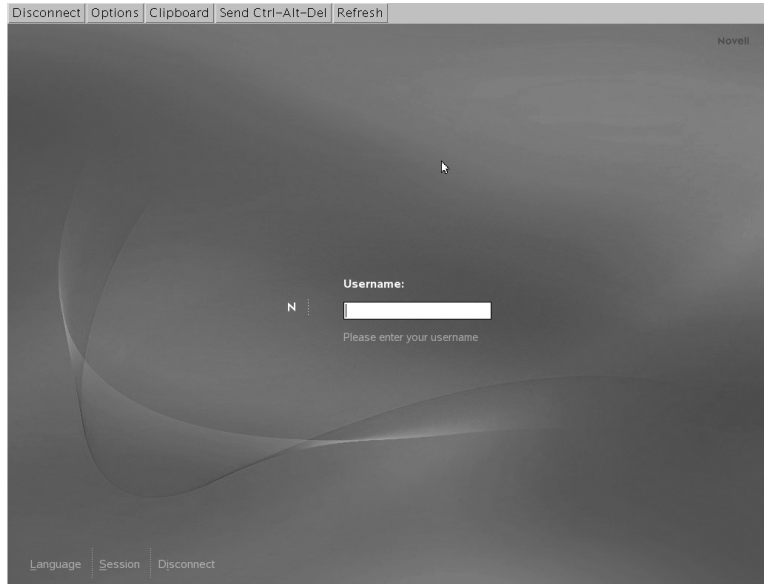
The following appears:

Figure 11-5



Select **OK**, no password is required at this point. The following appears:

Figure 11-6



From the top of the VNC session window, you can select from items such as setting session options and placing items in the clipboard.

From the session window, you can log in to a desktop environment on the server the same as if you were sitting at the physical machine.

Exercise 11-3 Use Remote Administration

In this exercise, you configure remote administration. In this exercise you work with a partner. (If there is no one available with whom to do the exercise, use localhost instead of the partner's computer.)

You will find this exercise in the workbook.

(End of Exercise)

Summary

Objective	Summary
1. Provide Secure Remote Access with OpenSSH	<p>The SSH suite was developed to provide secure transmission by encrypting the authentication strings (usually a login name and a password) and all other data exchanged between the hosts.</p> <p>SUSE Linux Enterprise Server 10 installs the package OpenSSH by default, which includes programs such as ssh, scp, and sftp as alternatives to Telnet, rlogin, rsh, rcp, and FTP.</p>
2. Enable Remote Administration with YaST	<p>SUSE Linux Enterprise Server can be administered remotely via SSH or VNC.</p> <p>You can enable remote administration via VNC by using the YaST Remote Administration module.</p>

Index

Symbols

/etc/at.deny 5-30
 /etc/cron.daily/logrotate 6-23
 /etc/cups/cupsd.conf 10-41
 /etc/cups/ppd/ 10-26, 10-36
 /etc/cups/printers.conf 10-34
 /etc/default/passwd 3-24
 /etc/default/useradd 3-21
 /etc/fstab 2-37, 2-68
 /etc/fstab, options 2-40
 /etc/init.d/cups 10-30
 /etc/inittab 3-26, 7-6, 7-22
 /etc/ld.so.cache 8-23
 /etc/ld.so.conf 8-22
 /etc/login.defs 3-24
 /etc/logrotate.conf 6-24
 /etc/logrotate.d/ 6-25
 /etc/pam.d/ 3-4
 /etc/pam.d/login 6-31
 /etc/passwd 1-45
 /etc/permissions* 3-31
 /etc/permissions.d/ 3-31
 /etc/permissions.local 3-31
 /etc/permissions.paranoid 3-31
 /etc/permissions.secure 3-31
 /etc/printcap 10-36
 /etc/rpmrc 8-5
 /etc/security/pam_pwcheck.conf 3-24
 /etc/shadow 1-45
 /etc/sudoers 3-17
 /etc/sysconfig/kernel 7-5
 /etc/sysconfig/network/ 4-10
 /etc/sysconfig/network/config 4-13
 /etc/sysconfig/network/ifcfg.template 4-13
 /etc/sysconfig/network/routes 4-18
 /etc/sysconfig/syslog 6-13
 /etc/syslog-ng/syslog-ng.conf 6-15
 /etc/syslog-ng/syslog-ng.conf.in 6-15
 /proc/ 6-5
 /root/autoinst.xml 1-49
 /sbin/init 7-4–7-5, 7-22
 /usr/bin/disable 10-25
 /usr/bin/enable 10-25
 /usr/lib/rpm/rpmrc 8-5
 /usr/sbin/accept 10-25
 /usr/sbin/reject 10-25
 /var/adm/rpmconfigcheck 8-10
 /var/lib/rpm/ 8-5
 /var/log/ 6-12, 6-22
 /var/log/boot.msg 6-3
 /var/log/cups/ 10-37
 /var/log/firewall 6-21
 /var/log/messages 6-21
 /var/log/wtmp 3-29
 /var/spool/cups/ 10-32–10-33
 ~/.rpmrc 8-5

CNI USE ONLY-1 HARDCOPY PERMITTED

A

- Access Control Lists 3-34
- ACL command line tools 3-41
- ACL mask 3-37
- ACL types 3-37
- ACL, access 3-37
- ACL, default 3-37, 3-47
- ACL, exercise 3-53
- ACLs 3-34
- ACPI 1-3
- Address 4-2
- administration, remote 11-28
- AES 11-4
- ampersand (&) 5-6
- APIC 1-4
- aquota.group 2-69
- aquota.user 2-69
- at 5-30
- atd 5-30
- automounter 1-47
- Autoyast 1-49

B

- background process 5-5
- backup 9-22
- backup strategy 2-2
- bg 5-6
- BIOS setup 1-2
- block bitmap 2-9
- block group 2-8
- Blowfish 1-31, 11-4
- boot loader 7-7
- boot loader, first stage 7-8
- boot loader, second stage 7-8

- boot manager 7-7
- boot settings 3-26
- boot.quota 2-69
- Broadcast 4-2

C

- CA 1-42
- CA management 1-42
- cancel 10-24
- Certification Intro-3
- certification authority 1-42
- child process 5-3
- chkconfig 7-36
- cipher text 11-3
- CLE 10 Intro-1
- clear text 11-3
- Common UNIX Printing System 10-1
- compatibility 9-26
- compressed 9-19, 9-41
- compression 9-19, 9-24, 9-26, 9-28, 9-41
- configuration 9-7, 9-15
- configuration, network 1-33
- configure 9-2, 9-8, 9-10–9-11, 9-15
- create 9-7, 9-9–9-10, 9-19–9-20, 9-30–9-32, 9-37, 9-41
- cron 5-25
- crontab 5-25, 5-28
- crontab, command 5-28
- crontab, file 5-26
- crontab, format 5-28
- Ctrl+Alt+Del 3-26
- CUPS 10-1
- CUPS Client Only 10-12
- CUPS Network Server 10-11
- CUPS Using Broadcasting 10-13

CUPS, access restrictions 10-47
 CUPS, access_log 10-37
 CUPS, browsing 10-43
 CUPS, configuration file 10-41
 CUPS, error_log 10-39
 CUPS, log level 10-40
 CUPS, page_log 10-39
 CUPS, resources 10-47
 CUPS, web Interface 10-55
 CUPS, web interface 10-52

D

daemon 5-20
 daemon process 5-2
 daemons, interval-controlled 5-21
 daemons, signal-controlled 5-21
 dd 2-60
 Default ACLs 3-47
 default route 4-17
 DES 11-3
 Destination (syslog-ng.conf) 6-20
 Device 4-2
 device 9-15
 Device URI 10-9
 df 2-44
 DHCP 1-36
 Diffie-Hellman 11-5
 directory 9-9, 9-12–9-13, 9-15, 9-19–9-24,
 9-32–9-34, 9-36–9-37, 9-43
 disk quotas 2-67
 dmesg 6-3
 driver 4-2
 DSA 11-5
 du 2-45
 dump 2-38

dumpe2fs 2-48

E

e2fsck 2-48
 edquota 2-70
 encryption key 11-3
 encryption, algorithm 11-3
 encryption, asymmetric 11-4
 encryption, symmetric 11-3
 eth0 4-4
 Ethernet adapter 4-4
 Ethernet devices 4-4
 ethtool 4-13
 Exercise Conventions Intro-10
 ext2 2-3
 ext3 2-4
 extended partition 1-11, 1-21, 2-26

F

facility (syslog) 6-16
 faillog 6-31
 fdisk 2-23, 6-6
 fg 5-8
 file
 system 9-7, 9-12–9-13, 9-19, 9-24, 9-33
 file system 2-2
 file system check 2-38
 file system type 2-37
 file system, create 2-30
 file system, encryption 2-31
 file systems, documentation 2-14
 file systems, internals 2-6
 file systems, journaling 2-4
 file systems, traditional 2-3

filters (syslog-ng.conf) 6-19
finger 6-29
firewall 1-32
foreground process 5-5
format a partition 2-30
forwarding, port 11-6
fsck 2-47
fuser 2-45

G

getfacl 3-35, 3-41
GID 3-15
GID, effective 3-15
gnomesu 3-16
Grand Unified Boot Loader 7-7
graphic card 1-47
Group ID Settings 3-30
group password 3-15
grpquota 2-68
GRUB 7-7
GRUB shell 7-10
grub-md5-crypt 7-20

H

halt 7-41
hardware address 4-5
hardware clock 1-8
hardware information 6-5
hardware RAID 2-63
hdparm 6-6
hostname 1-29
hwdm 6-6

I

IDE controller 2-18
IDEA 11-4
init 5-5, 7-22, 7-40
init=/bin/bash 7-19
initramfs 7-4
initrd 7-4
inode bitmap 2-9
inode table 2-9
insserv 2-69
installation proposal 1-9
installation settings 1-9
installation, configuration 1-29
installation, network 1-31
installation, partitioning 1-10
installation, start 1-28
installation, troubleshooting 1-50
internet connection, test 1-38
Internet Printing Protocol 10-1
iostat 6-6
ip 4-3
IPP 10-1, 10-8
IPv4 4-5

J

job 5-5
job identifier 5-5
Jobs 5-25
jobs 5-7
jobs, user 5-28
John (password cracker) 3-11
journaling 2-13

K

key, private 11-4
key, public 11-4
key, secret 11-3
kill 5-16
kill, signals 5-18
killall 5-16

L

LABEL 2-37
last 6-30
lastlog 6-30
LD_LIBRARY_PATH 8-22
LDAP 1-42
LDAP client 1-46
LDAP Server. address 1-46
ldconfig 8-22
ldd 8-20
libraries, dynamic 8-19
libraries, static 8-19
LILO 2-18, 7-9
LILO (LInux LOader) 7-8
Link 4-2
listener 5-20
local time 1-8
local users 1-45
locate 3-31
locatedb 3-31
Log Path (syslog-ng.conf) 6-20
logical partition 1-11, 1-19, 2-26
logical volume 2-52
Logical volume manager 2-51
login settings 3-28
logrotate 6-23

loopback device 4-4
lost+found 2-46
lp 10-22
lpadmin 10-28
LPD 10-9
lpoptions 10-26
lpq 10-23
lpr 10-22
lprm 10-24
lpstat 10-23
lsof 2-45
lspci 6-7
lvcreate 2-62
lvextend 2-62
LVM 2-51
LVM, features 2-53
lvreduce 2-62
lvscan 2-62

M

Magic SysRq Keys 3-32
Maintenance Intro-5
management 9-8–9-9, 9-13
mask, ACL 3-38
master 9-30
memory test 1-4
metadata 2-13
minix 2-4
mke2fs 2-32
mkfs 2-32
mkfs.ext3 2-32
mkinitrd 7-5
mkreiserfs 2-32, 2-35
mount 2-36, 2-39
mount options 2-38

mount point 2-32, 2-37

MS-DOS/VFAT 2-3

N

named group 3-37

named user 3-37

network 9-9, 9-14

network file systems 2-12

network interface 1-33, 4-2

network interface, setup 4-3

network, configuration 1-31

NetworkManager 4-27

newgrp 3-15

NFS 9-9, 9-14

nice 5-12

nice value 5-10

nm-applet 4-28

nm-tools 4-27

Novell CLE 10 Intro-1

Novell Customer Center Intro-6, 1-39, 1-41

NTFS 2-4

O

Online Resources Intro-7

Online Update 1-40

OpenSSH 11-2

options 9-10–9-11, 9-23, 9-33

P

PAM 3-2

PAM, arguments 3-8

PAM, Configuration Files 3-4

PAM, control flags 3-6

PAM, documentation 3-12

PAM, exercise 3-13

PAM, Illustration 3-3

PAM, module types 3-5

PAM, modules 3-7

pam_tally.so 6-31

parent process 5-4

partition 9-30

partition table 2-16

partition type 2-27

partition, delete 1-22

partition, edit 1-22

partition, new 1-18

partition, resize 1-22

partitioning 2-22

partitioning scheme 2-18

partitions 1-11, 2-16

partitions, naming convention 2-17

partprobe 2-29

password security settings, exercise 3-33

password settings 3-24

password, root 1-29

Password, secure 3-11

Patch RPMs 8-13

path 9-9

pattern 1-26

permissions, effective 3-38

physical extent 2-55

physical volume 2-52

PID (Process ID) 5-3

Pluggable Authentication Modules 3-2

PostScript Printer Description 10-26

poweroff 7-41

Power-On Self Test (POST), 7-7

PPD 10-26

PPID (Parent PID) 5-4

primary partition 1-11, 2-25
print jobs, management 10-21
print queue, new 10-6
print queues 10-34
print queues, management 10-21
printer 1-47
printer, add 10-4
printer, add, CLI 10-19
printer, add, YaST 10-4
printer, change configuration 10-20
printer, directly connected 10-7
printer, network 10-8
printing, groups 10-50
printing, users 10-50
priority (syslog) 6-17
process ID 5-3, 5-5
process priority 5-10
process, management 5-2
program 5-2
property 9-15–9-16
protocol 9-29–9-30, 9-42
pstops 10-32
pstree 5-12
pvcreate 2-61
pvmove 2-61
pvscan 2-61

Q

quota 2-67
quotacheck 2-69
quotaoff 2-70
quotaon 2-70

R

RAID 2-63
RAID 0 2-63
RAID 1 2-63
RAID 5 2-64
RAID 6 2-64
rccups 10-30
reboot 7-41
ReiserFS 2-4, 2-10
ReiserFS format 2-10
reiserfsck 2-48
reiserfstune 2-49
remote administration 11-28
Remote IPP Queue 10-15
renice 5-12
repquota 2-73
Rescue System 1-4
resize_reiserfs 2-49
resize2fs 2-49
Rijndael 11-4
root partition 1-12
root password 1-29
Route 4-2
Routing 4-15
routing table 4-15
RPM 1-26
RPM database 8-2, 8-5
RPM package 1-26
RPM Package Manager 8-2
RPM query options 8-11
rpm, freshen 8-9
RPM, install 8-8
RPM, uninstall 8-10
RPM, upgrade 8-9
RPM, verify 8-12

rpmbuild 8-6
RSA 11-5
runlevel 7-22

S

scp 11-2
SCSI 2-17, 9-6, 9-26–9-27
security 9-2
Security Settings 3-22
server 9-2, 9-4, 9-6–9-7, 9-9, 9-14, 9-34
service 5-20
service, network based 5-2, 5-20
service, time-based 5-2, 5-20
setfacl 3-35, 3-41
sftp 11-2, 11-16
sg 3-15
siga 6-7
sit0 4-5
sitar 6-7
SMB 10-9
socket 10-10
software library 8-18
software RAID 2-63
software, printing 10-3
sound 1-47
source (syslog-ng.conf) 6-18
SSH 1-32, 11-2
ssh-agent 11-25
SSL 1-42
start 9-7, 9-13, 9-17, 9-27
storage 9-2, 9-6–9-7
su 3-14
sudo 3-16
sudo, configuration 3-17
sudo, example 3-18

superblock 2-8
Support Intro-5
swap partition 1-12
SysRq 3-32
system 9-3, 9-5, 9-7, 9-10, 9-12–9-13, 9-19,
9-24, 9-29, 9-33–9-34, 9-37, 9-39,
9-43

T

tail 6-21
time 9-3–9-5, 9-11, 9-13, 9-22, 9-33, 9-39
time zone 1-8
TLS 1-42
top 5-14
top, commands 5-15
transmission 9-34
Triple-DES 11-4
tune2fs 2-49
type 9-10, 9-26–9-27

U

umask 3-42
umount 2-41
Universal Resource Identifier 10-9
updatedb 3-31
upgrade 9-11
URI 10-9
User Access 3-1
user accounts, defaults 3-19
user authentication 1-43
User ID settings 3-29
user process 5-2
usessh 11-26
usrquota 2-68
UTC 1-8

CNI USE ONLY-1 HARDCOPY PERMITTED

UUID 2-37

V

VFS 2-5

vgcreate 2-61

vgexpand 2-61

vgreduce 2-61

vgremove 2-61

Virtual Filesystem Switch 2-5

visudo 3-17

VNC 11-28

VNC client 11-29, 11-31

volume group 2-52

volume group, name 2-55

W

w 6-28

who 6-28

X

XFS 2-4

Y

YaST Expert Partitioner 1-16, 2-21

YaST online update 1-41

YaST, installation 1-1

YaST, software 1-25

YaST, view_anymsg 6-4

yast2 lvm_config 2-59

yast2 remote 11-29

CNI USE ONLY-1 HARDCOPY PERMITTED