SuSE

SuSE Linux

Enterprise Server 8

Installation

for both x86 and
AMD64 systems

Introduction

This book describes the procedure for installing SuSE Linux Enterprise Server on either x86 or AMD64 systems. It provides all information needed for preparing an installation on the BIOS and the firmware side of your system as well as the installation procedure of SuSE Linux Enterprise Server itself. When possible, links are provided to more specific documentation both on the net or as part of your installed Linux system.

Document Structure

Basically, this installation guide is divided in two parts.

Preparation This part provides information about the requirements on the hardware and software side that need to be met to successfully install SuSE Linux Enterprise Server on your machine. You will be informed which basic preparations needed to be performed on the BIOS or firmware.

Installing SuSE Linux Enterprise Server This part covers the complete installation procedure including the preparation of the installation system, network setup, and a complete in-depth description of the YaST2 install procedure. A summary of all supported network connection types and some short examples are also provided.

The appendix contains additional chapters (see *Overview* on page 1).

A README on the SuSE Linux Enterprise Server installation CD also contains last minute changes and additional information on the installation procedure.

Required Background

We have made several assumptions concerning your background knowledge when designing this document.

- You are familiar with the aspects of BIOS or firmware settings and their terminology.

- You have a basic understanding of Linux system administration.

- You have a good knowledge of the devices attached to your system, such as printers, network interfaces, and hard drives.

Typographical Conventions

The following typographic conventions are used in this book:

Example	Meaning
YaST	programs
/etc/passwd	files or directories
⟨*placeholder*⟩	replace the character string `placeholder` (including the angle brackets) with the actual value
PATH	an environment variable called `PATH`
192.168.1.2	the value of a variable
ls	commands
user	users
earth:~ # **ls**	enter the command `ls` in the shell of the user `root` in his home directory on the host "Earth"
newbie@earth:~ > **ls**	enter the command `ls` in the shell of the user `newbie` in his home directory on the host "Earth"
C:\> **fdisk**	at the DOS prompt, enter the command fdisk

Table 1: *continued overleaf…*

(Alt)	press this key; keys to press sequentially are separated by spaces
(Ctrl) + (Alt) + (Del)	keys to press simultaneously are connected with a ' + '
"Permission denied"	system messages
'System update'	menu item or button text 'System update'
boot	reference to an entry in the glossary in the appendix

Acknowledgments

The history of Linux is a success story about countless developers all around the world contributing to what originally started as a "one-man-show" by Linus Torvalds. Thanks to all of them for their tremendous efforts.

Thanks especially go to:

- the developers
- the test team at SuSE
- all beta-testers and proofreaders

Thanks for making SuSE Linux Enterprise Server for x86 and AMD64 systems possible.

Nuremberg, 26th January 2004

Your SuSE team

Contents

Overview

Before you install SuSE Linux Enterprise Server, read the instructions in the chapters covering the requirements and the preparation of the installation to obtain the following information about the hardware and software requirements, preinstallation tasks, and the actual installation:

- The hardware and software requirements that must be met in order to install SuSE Linux Enterprise Server.

- Preparation of the partitions and selection of a suitable installation medium. You can also control the graphical installation by way of a network connection (VNC).

- The actual installation of SuSE Linux Enterprise Server on your system with YaST2.

The appendix provides additional information in chapters such as *AutoYast2: Automatic Linux Installation and Configuration with YaST2*, *YaST2 in Text Mode (ncurses)*, *LVM - The Logical Volume Manager*, *File Systems in Linux*, and *Support Services*.

Requirements

Hardware

x86

Generally, the installation of SuSE Linux Enterprise Server on x86 systems is rather easy. The following hardware equipment is recommended:

- At least 256 MB of main memory.

- The hard disk should be larger than 4 GB.

AMD64 Systems

We recommend the following minimum hardware equipment for this brand-new system:

- At least 256 MB of main memory.

- The hard disk should be larger than 10 GB.

  The installation can be performed with both IDE and SCSI systems.

Certification

Nowadays the production cycles of hardware products are very short. Certifications are necessary to give customers the guarantee that the product can be deployed on various platforms. Formerly, this service was not available

for Linux. For this reason, SuSE Linux AG offers hardware manufacturers the
possibility to have their products certified in the SuSE development center.

For more information about certified hardware, refer to
`http://www.suse.de/de/services/certified_hardware/index.html`.

Firmware and BIOS

All current firmware variants work smoothly. However, make sure booting from
CD is supported.

Preparation

Partitions

Hard disk partitioning is often essential. Even if it is not absolutely necessary, it is useful for several purposes. For example, you may want to install multiple operating systems, keep your data in various file systems regardless of the operating system (see page 95), or expand your partition (even across multiple hard disks) at a later date without having to format the hard disks and lose data (see page 87). The partitions should be set up in consideration of the following four important factors:

- Hard links — The use of hard links is limited to the used file system.

- Mount options — When transferring individual directories to separate partitions, the security approach deserves close attention.

- Performance — Setting up a partition for your data prevents the available hard disk space from being consumed by log files.

- Scalability — If there are many partitions, you may not be able to resize some of the partitions easily. Changing the size of partitions in a running system is difficult. Make sure to create large enough partitions during the installation.

Take a look at the following two examples:

File server For a file server, the space for saving data should be assigned to a special partition apart from the operating system. The performance will drop when a file system is full (when little space is left on a hard disk).

Database server It is not advisable to install your database server in the partition containing the operating system. The advantage of installing the database server in a separate partition is that the configuration of the database server will not change and no data will be lost even if the operating system is reinstalled from scratch.

The hardware should consist of two main systems: a computer system (server) with one or two internal hard disks, a RAID controller, and an external storage unit with a lot of space, which is usually connected by means of an SCSI cable or a glass fiber cable.

We recommend that you use at least two internal hard disks in the server unit, connecting these with a RAID 1 (mirroring) configuration in the RAID controller of the server to be protected in case one of the drives fails. Set up three partitions: a swap partition (virtual memory), a partition for the operating system, and a partition for the database installation.

Preparing a VNC Client for the Installation

Introduction to VNC

VNC "Virtual Network Computing" is a Client Server solution to control a remote X-Server via a small and simple to use client. This client is available for many operating systems, including various Microsoft Windows versions, Apple's MacOS and Linux.

What is VNC used for?

SuSE Linux Enterprise Server uses VNC to enable the use of the graphical user interface of the installation software, YaST2. It is needed for systems, that cannot provide a graphical console or if the administrator cannot access the graphical console (e.g. a rack mounted system without a monitor).

The VNC client vncviewer is used to enable the display and control of YaST2 during the installation process. Before booting for the installation, you need to prepare a remote computer, which is connected via network to the system you want to install on.

Preparing a SuSE Linux Client

SuSE Linux Enterprise Server can also be used as a control client for the installation. The VNC client vncviewer is part of the Package vnc. Use the YaST2 'Install/Remove Software' module to install the package.
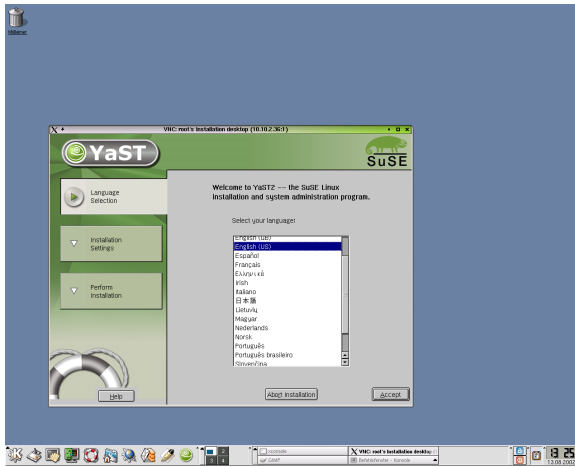
Figure 3.1: *YaST2 on a SuSE LinuxDesktop*

Preparing a Microsoft Windows Client

The VNC client for all Microsoft Windows 32bit versions can be found in the `dosutils/vnc` directory on CD 1. Copy the file vnc-3.3.3r9_x86_win32.tgz to a directory on your harddisk and unpack the file with the tar.exe, which can be found in the `dosutils/gtar109` directory on CD 1.

```
C:\ tmp > tar -xvzf vnc-3.3.3r9_x86_win32.tgz
```

The VNC client vncviewer.exe can be found in the `vnc_x86_win32/vncviewer` directory.

Initiating the Installation for VNC

The boot process for a VNC based installation slightly differs from a text based installation. The process looks as follows:

- Add the necessary kernel parameters to the boot prompt

- Boot the installation kernel

- If necessary, do the basic network configuration on the system console

- Start the vncclient on your client system and start installing the software

vnc=1	This enables the installation via VNC
vnc_password=xyz	This sets the password to access the VNC installation to xyz. Your VNC client will ask you for this password before you can access the installation procedure. Setting this option automatically enables the installation via VNC.
dhcp=1	Setting this option, the installation procedure will automatically do a network configuration via DHCP. You need a working DHCP server on your network.

Table 3.1: *Kernel parameters for VNC installation*

Additional Kernel Parameters

To enable the VNC installation, you need to add at least one parameter to the kernel parameter line. This needs to be done before the kernel is started.

The following table (*Additional Kernel Parameters* on the current page) shows the kernel parameters useful in connection with a VNC installation:

In this manual, the installation via VNC is used on those platforms that do not support a graphical console.

Initiating the VNC Installation

To start an installation via VNC, you need to add the additional kernel parameters at the bootprompt of the installation CD. After the system read the bootloader you are able to add the parameters:

```
boot:   install vnc=1 vncpassword=suse
```

The password is just an example here. Hit (Enter) to initiate the loading of the kernel.

Watch the console showing the kernel messages. If you decided not to do an automatic network configuration via DHCP (dhcp=1), you will be asked for some network parameters:

1. Select the networking device you want to use for the installation.

2. Enter the IP address and subnet mask for this device.

3. Enter the Gateway and DNS information for this network.

4. Select the installation source. Three options are possible:

   CD-ROM Select this option, if you are using the installation media shipped with your SuSE Linux Enterprise Server.

   Harddisk This is useful, if you have the installation data available on a separate harddisk. Enter the devicename with the partition and the path to the installation data in the following dialog (e.g. `/dev/sdb1` and `/suse`).

   Network Installation Source Allows you to access the installation data from an NFS-shared. Enter the Hostname/IP address of the NFS server and the path to the NFS-shared installation data (e.g. `suse-share`).

Finally, a message will appear prompting you to start your VNC client on the remote system.

```
starting VNC server...
a log can be found in /tmp/vncserver.log ...

***
***     You can connect to 10.10.1.154, display :1 now
***

(When YaST2 is finished, close your VNC viewer and return to this window.)
```

Start the VNC client with the parameters shown in the output and enter the VNC password (`suse` in our example). The VNC graphical screen will appear and a few seconds later, YaST2.

Proceed with Chapter *Installation with YaST2* on page to start the actual installation of the software.

Installation

You have the possibility to perform the installation from CD, by means of the network, or from the hard disk. Read the following information regarding the settings for the respective installation method:

- Section *Installing from CD* on the current page to install from CD

- Section *Installing from a Hard Disk* on the facing page to install from the hard disk

- Section *Installing via the Network* on page 12 to install via the network

Installing from CD

To start the installation, insert CD 1 into your CD drive and reboot the computer. The boot loader of the CD appears. After a few seconds, the kernel is started and you can follow the kernel messages on the screen.

If the kernel was loaded successfully, you will see the welcome screen of YaST2, which starts automatically. All settings that need to be performed in YaST2 are described in Chapter *Installation with YaST2* on page 13.

If you can boot your machine from the CD, proceed with the installation using YaST2. Booting from the CD may not be possible for the following reasons:

- Your CD-ROM drive may not be able to read the "boot image" of the first CD. In this case, use CD 2 to boot the system. The second CD contains a conventional 1.44 MB boot image, which can be read even by older drives.

- The boot sequence of the machine may be incorrect. Information about changing the BIOS parameters are provided in the documentation of your motherboard and in the following paragraphs.

  The BIOS is a small utility used to start the basic functionalities of the computer. Motherboard manufacturers provide a BIOS adapted to the characteristics of the motherboard.

  The BIOS setup can only be entered at a specific time. At the system start-up, some hardware tests are performed, such as a memory test. You can see this from the system memory counter. At the same time, the key to press to enter the BIOS setup is displayed at the bottom of the screen. Usually this is done by pressing (Del), (F1), or (Esc). Press the respective key to enter the BIOS setup.

  When the BIOS setup has started, change the boot sequence as follows: In an AWARD BIOS, look for 'BIOS FEATURES SETUP'. Other manufacturers use similar entries, such as 'ADVANCED CMOS SETUP'. Select the respective entry and confirm with (↵).

To change the boot sequence, look for the item defining the start sequence of the drives, usually C, A or A, C. In the first case, the machine first searches for the operating system on the hard disk (C) then in the floppy disk drive (A). Select 'Boot Sequence' and press (Page ↑) or (Page ↓) until the sequence A, CDROM, C is displayed.

Exit the settings by pressing (Esc). To save the changes, select 'SAVE & EXIT SETUP' or press (F10). Confirm your settings with (Y). To undo the settings, press (N).

If you have a SCSI CD-ROM drive, start its BIOS (for example, for an Adaptec host adapter, press (Ctrl) + (A)). Select 'Disk Utilities'. The system checks and displays the connected hardware components. Make a note of the SCSI ID for your CD-ROM drive. Exit the menu with (Esc) then enter 'Configure Adapter Settings'. Under 'Additional Options', select 'Boot Device Options' and press (↵). Enter the ID of the CD-ROM drive and press (↵). Press (Esc) twice to return to the start screen of the SCSI BIOS. Exit this screen and confirm with 'Yes' to reboot the machine.

- Your CD-ROM drive may not be supported because it is an older model. To check whether your drive is supported, refer to the Hardware Database of SuSE Linux Enterprise Server at http://hardwaredb.suse.de/en/.

Installing from a Hard Disk

You can also install directly from the hard disk. The file system of the partition containing the data from the CDs must be supported by Linux. The advantage of using a hard disk is that it is much faster than the network installation and does not require any additional settings for the network sources.

To start the installation, insert CD 1 into your CD drive and reboot the computer. When the boot loader appears, select 'Manual Installation' and hit (Enter). Proceed as follows to set up this installation method:

1. Select the language for linuxrc.

2. Select 'Start installation / system'.

3. Select 'Start installation/update'.

4. Select 'Hard disk' as the source medium.

5. When your hard disk is displayed, select the device and the partition containing the installation data.

After these steps, YaST2 is loaded and started. Continue reading in Chapter *Installation with YaST2* on the next page.

Installing via the Network

If you want to install SuSE Linux Enterprise Server on several systems, set up an installation source that can be accessed via the network to save time. The advantage of this approach is that you do not need to change the CDs for future installations and you can perform the installation simultaneously on multiple partitions.

A network copy of the installation CDs should be placed on a system in the respective directories. For example, copy the contents of a CD using the following command:

```
# cp -a /media/cdrom /suse-share/
```

To rename the created directory to 'CD1', enter the following command:

```
# mv /suse-share/cdrom /suse-share/CD1
```

Repeat this procedure for the second CD (CD2).

> **Note**
>
> The directory suse-share/ must be exported with NFS and the NFS server must be started.
>
> **Note**

To start the installation, insert CD 1 into your CD drive and reboot the computer. In the boot loader of the CD, select 'Manual Installation'. Subsequently, the kernel starts and linuxrc appears. Proceed as follows:

1. Select the language for linuxrc.

2. Select 'Start installation / system'.

3. Select 'Start installation/update'.

4. Select 'Network' as the source medium.

5. Select your network device when it is displayed.

6. Enter the IP address and the network data.

7. Enter the IP address of your NFS server.

8. Specify the path of the NFS directory.

After these steps, YaST2 is loaded and started. Continue reading in Chapter *Installation with YaST2* on the facing page.

Installation with YaST2

YaST2 Takes Over

Now the actual installation of SuSE Linux Enterprise Server starts with the YaST2 installation program. All YaST2 screens have a common format. All entry fields, lists, and buttons on the YaST2 screens can be accessed with your mouse. If your cursor does not move, your mouse has not been automatically recognized by Linux. You will then need to use your keyboard.

If YaST2 was started in textmode, you cannnot use the mouse. Refer to Chapter *YaST2 in Text Mode (ncurses)* on page 83 for details on the usage.

Selecting a Language

SuSE Linux Enterprise Server and YaST2 are adapted to use the language selected. English is the default setting for the international distribution of SuSE Linux Enterprise Server. These settings can be changed individually. If your mouse does not work, navigate with the arrow keys to the desired language then press (Tab) repeatedly until the 'Next' is selected. Then press (↵).

Installation Mode

Here you can decide if you want to do a new installation, or just boot an already installed system (Figure 4.2 on page 15). This might be necessary in cases like a misconfigured bootloader.
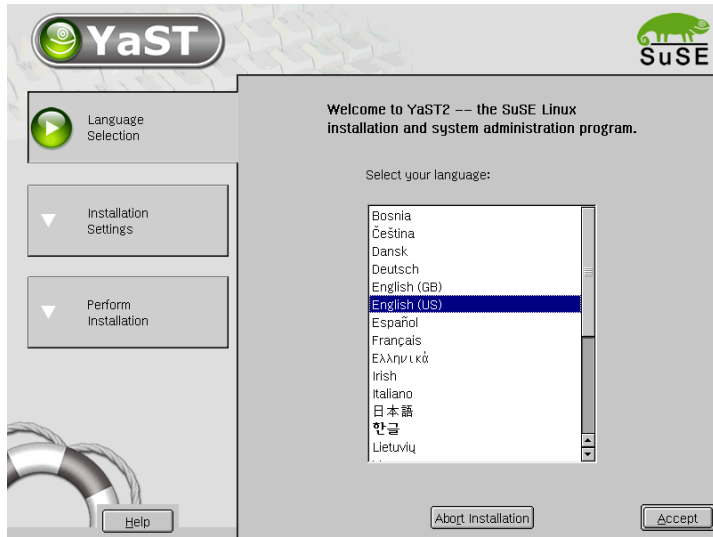
Figure 4.1: Selecting the Language

Installation Suggestions

After the hardware has been detected and the mouse configured, YaST2 provides information about the detected hardware and the suggestions for the installation and the partitioning. After you modify a suggestion, YaST2 returns to the suggestion window. The following sections explain the various configuration settings available.

Mode

Here, change the installation mode selected before the suggestion window appeared if you already have a Linux system on your computer. You can also boot your installed system from here. This is useful if your system is no longer able to boot from the hard disk.
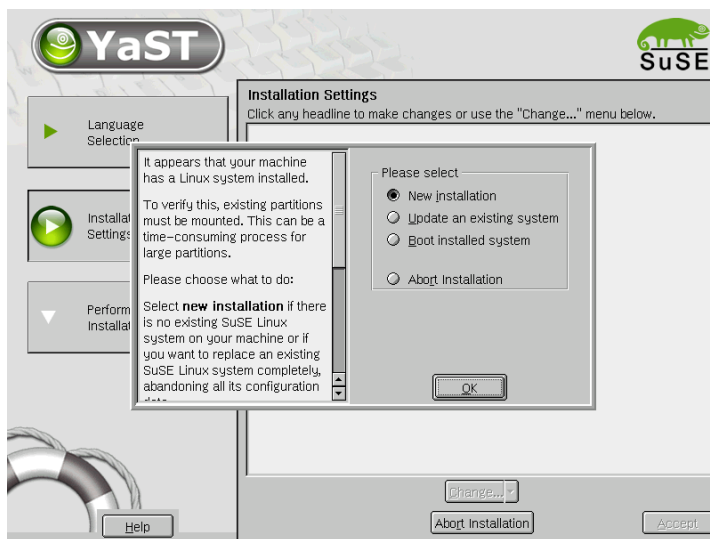
Figure 4.2: Selecting the Installation Type

Keyboard Layout

Select a keyboard layout. By default, the layout corresponds to the language selected. After changing the layout, use the field to test special characters, Y, and Z to make sure the layout is correct. If they are not displayed correctly, the keyboard layout is not correct. Return to the suggestions with 'Next'.

Mouse

If YaST2 did not recognize your mouse type automatically, use $\boxed{\text{Tab}}$ until 'Change' is highlighted. Press $\boxed{\text{Space}}$ and the arrow keys until 'Mouse' is selected. Press $\boxed{\text{↵}}$ to open a mouse type selection screen, illustrated in Figure 4.3 on the next page.

To select your mouse type, use $\boxed{\uparrow}$ and $\boxed{\downarrow}$. Your mouse documentation should include a description of the mouse type. Select the mouse type from the list. Confirm your selection by pressing $\boxed{\text{Alt}}$ + $\boxed{\text{T}}$ or $\boxed{\text{Tab}}$ and $\boxed{\text{↵}}$.

Now, test to see if your mouse is working. If the mouse cursor on the screen follows your mouse movements, your mouse is configured correctly. If the cursor does not move, select a different mouse type and try again.
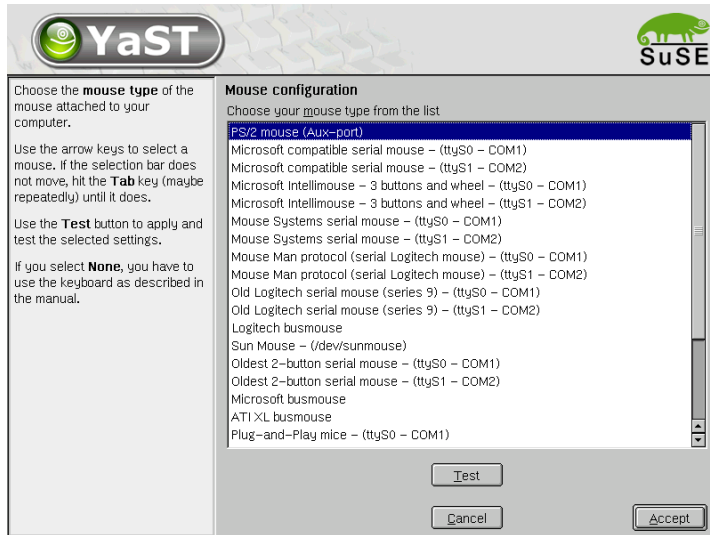
Figure 4.3: Selecting the Mouse Type

Partitioning

During installation you can split up the available disk space into several logical "partitions". This procedure is called "partitioning".

The YaST2 Partitioner

Manual Partitioning

With the "Partitioner" (see Figure 4.4 on the facing page), the partitions of your hard disk can be manually modified. Partitions may be added, removed, or changed.

After selecting 'Partitioning' in the proposal screen and selecting 'Partitioning based on proposal', the partitioner lists all hard disks including the proposal for the partitioning. Disks are listed as devices without numbers (for example, /dev/hda or /dev/sda) including the brand. Partitions are listed as parts of the hard disks (such as /dev/hda1 or /dev/sda1). The most important parameters like size, mount point and type are also listed. The mount point is the location in the Linux file system structure to which this partition is mounted.
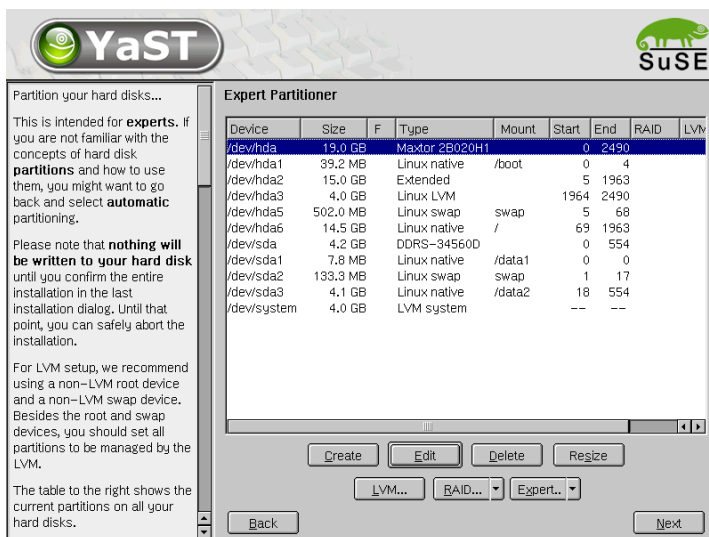
Figure 4.4: *The SuSE Linux Enterprise Server Partitioner*

Creating a Partition

To create a new partition:

- Select 'Create'. If there is more than one hard disk in your system, select the desired hard disk now.

- A dialog appears asking for the type of partition. You can create up to four primary partitions or up to three primary partitions and one extended partition. Within the extended partition, you can create several "logical" partitions.

- Select the file system to use for this partition and, if necessary, a mount point. YaST2 suggests a mount point for each partition. Details of this can be found in the following section.

- Select 'OK' to apply your changes.

The new partition is then listed in the partition table. Selecting 'Next' writes the partition table to disk and formats, if necessary.

Parameters for Partitioning

To add a new partition to the filesystem tree, set some parameters as follows:

1. Select the partition.

2. 'Modify' the partition set the parameters:

   File System ID   When selecting the partition type, select "Linux swap", "Linux", "Linux LVM", or "Linux RAID". Details on LVM and RAID can be found in *LVM - The Logical Volume Manager* on page 87.

   File System   Which file system to choose strongly depends on for what the system will be used. The "ext2" file system is known to be very stable, as it is the oldest Linux file system. If you plan to store a huge amount of data on large disks, using a "Journaling" file system, such as "ext3" or "reiser", should be considered. While "ext3" is very good in handling only a few huge files, "reiser" can manage many small ones. This influences the speed and the amount of space needed for the file system organization. More details can be found in the appendix in *File Systems in Linux* on page 95.

   Mount Point   Sets the directory where the new partition will be mounted in the existing filesystem tree.

3. Select 'Continue' to save the settings for the partition.

More Partitioning Tips

If the partitioning is performed by YaST2 and other partitions are detected on the system, these partitions are also entered in /etc/fstab to enable easy access to this data. View sample in File 1.

```
/dev/sda1        /data1  auto    noauto,user 0 0
/dev/sda8        /data2  auto    noauto,user 0 0
```

File 1: /etc/fstab: Data Partitions

The partitions, regardless of whether they are Linux or FAT partitions, are specified with the options noauto and user. This allows any user to mount or unmount these partitions if needed. For security reasons, YaST2 does not automatically enter the exec option here. However, to run programs from there, you can enter this option yourself. This procedure can wait until you encounter system messages such as "bad interpreter" or "Permission denied".

Software

Here, determine which software to install on your machine. Furthermore, you can select the following Softwarepacks:

'Minimal'   This selection installs a fully functional Linux operating system providing textmode only.

'Minimum graphical system (without KDE)'   This option installs a minimal system including the X-Window-System. You will be able to use FVWM2 or Windowmaker as the windowmanager for the graphical environment.

'UnitedLinux'   This selection contains all the UnitedLinux standard packages.

'Standard'   This is the SuSE Linux Enterprise Server standard installation.

Select 'Details' to do a more detailed software selection.

Here, you can decide to install an LSB conform system. Select 'LSB Runtime Environment'.
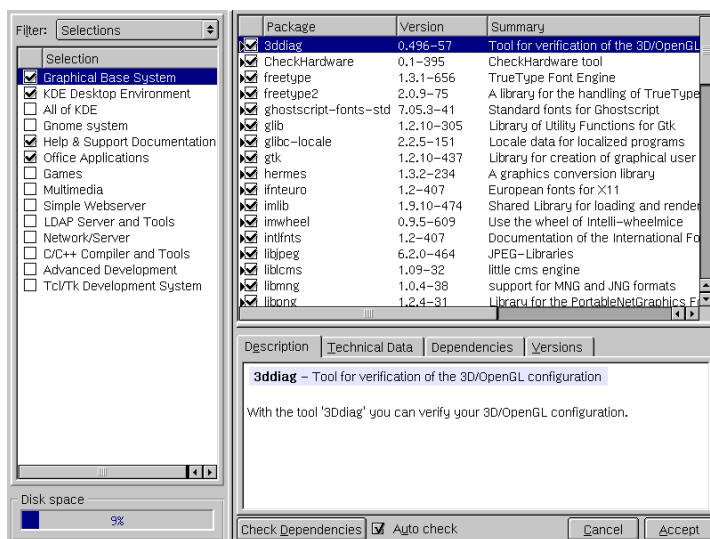
Figure 4.5: *Installing and Removing Software*

Preselection

The module starts with the selection filter. At the top left next to 'Filter', 'Selection' is marked. These selections represent groups of program packages to select for installation or removal by clicking the respective check box. Below, see possible preselection groups of this filter, some of which are already selected as they belong to the default installation of SuSE Linux Enterprise Server.

The right frame displays a list of the individual packages that belong to the selection. Packages selected for installation are checked to the left of the package name. Select and deselect individual packages according to your needs. To do this, click the symbol several times until the desired status is shown.

Other Filters

Click 'Filter' to see a selection of additional filters that can be used to structure the view of the packages. For example, there is a selection according to 'package groups', which is also defined as the default filter when you start the software selection in YaST after the system has been installed. Using this filter, the program packages are displayed according to subjects in a tree structure on the left side. The more you unfold the tree in a package group, the more detailed the selection will be and the smaller the number of related packages in the package list on the right side will be.

Boot Loader — Installation)

To start SuSE Linux Enterprise Server or any other operating system on your computer, you need a program that enables you to decide whether to use SuSE Linux Enterprise Server or another operating system when starting (or booting) the machine. LILO ("**LI**nux **LO**ader") and GRUB ("**GR**and **U**nified **B**ootloader") are two programs that can be used for this purpose.

Up to this version, LILO was the default boot manager of SuSE Linux Enterprise Server. If you updated your old system, LILO was retained as boot manager. On the other hand, if you performed a new installation, your current boot loader is GRUB. However, you will notice no difference whatsoever in terms of utilization and user-friendliness.

By default, YaST2 suggests installing the boot loader to the Master Boot Record (MBR) of the hard disk. If you have more than one hard disk, it will recognize the hard disk used for booting from the BIOS settings.

This is displayed in the "suggestion screen" under 'Booting'. YaST2 will detect whether you have any other operating system apart from SuSE Linux Enterprise Server Depending on the hardware equipment, there are several possibilities for installing the boot loader. The following paragraphs provide an introduction to the boot manager. Enter the boot loader configuration by clicking 'Booting'. The dialog displayed in Figure 4.6 will appear.
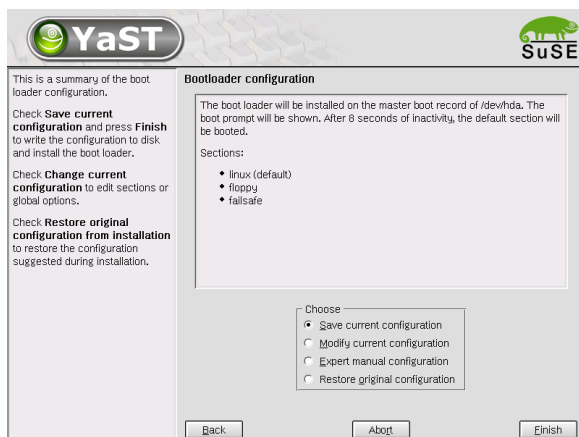
Figure 4.6: *Boot loader – installation*

In this dialog you can determine how and where the boot loader is to be installed. The following options are available:

- Write GRUB to the boot disk ('MBR' of /dev/hda)

- Create a boot floppy

- Do not use GRUB (a different boot manager is required)

- Write GRUB to a different partition

If you want to use an existing boot loader or a different boot loader, please select the option 'Do not use GRUB'.

The first option is usually suitable. This option installs the boot loader in the Master Boot Record (MBR) of your (boot) hard disk. You should also use this option if you intend to use the Linux boot loader as boot manager for multiple operating systems. However, make sure that your operating system can

be booted by LILO or GRUB (applies to MS-DOS, Windows 9x/ME, NT, 2000, XP, and OS/2).

If you have several operating systems about which you are not sure whether they can be booted from your Linux boot loader or in case you do not want to modify the previous boot mechanism, use the option 'Create a boot floppy'. In this way you can boot Linux by inserting the floppy disk in the drive before you switch your computer on. If the floppy disk is not in the drive, the other operating system will start. Later on, you can go to the configuration of the boot loader in YaST2 and modify it in such a way that you can select an operating system at system start-up.

If you have already installed a different boot manager and want to include SuSE Linux Enterprise Server in this boot manager, select 'Do not use GRUB'. Following the installation of SuSE Linux Enterprise Server you have to reconfigure the existing boot manager and include SuSE Linux Enterprise Server in the boot routine.

If your system requires additional kernel parameters for each start-up, enter these parameters in the respective field. Usually nothing needs to be entered here.

Time Settings

In this screen (Figure 4.7 on the next page), choose between `Local Time` and `GMT` in the field marked 'Set hardware clock to'. Your selection depends on the BIOS clock settings for your computer. If the hardware clock is set in the BIOS to `GMT`, SuSE Linux Enterprise Server automatically reflects Standard and Daylight Savings time changes.

Starting the Installation

A click on 'Next' accepts the suggestions and any changes made. A green confirmation screen opens. After clicking 'Yes' here, the installation begins using your settings. The installation usually takes between fifteen and thirty minutes, depending on your machine's performance and your software selection.

Caution

If you chose 'Whole disk' in Section *Manual Partitioning*, all data on your harddisk will be lost.

Caution
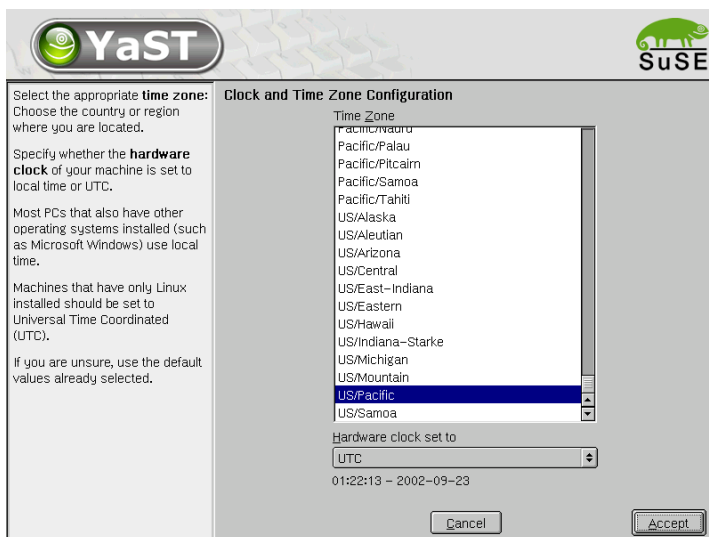
Figure 4.7: *Selecting the Time Zone*

System Configuration

After the installation of your system and selected software is complete, you need to make three more important settings before you can work with SuSE Linux Enterprise Server: define a password for the system administrator `root`, create a normal user, and configure your monitor. The following sections show how this is done.

Root Password

`root` is the name of the superuser, the system administrator. `root` is permitted to do all the things normal users are not permitted to do. The superuser can make changes to the system, such as installing new applications or setting up new hardware. If users forget their passwords or have problems with software, `root` can help them. As a general rule, only log in as root to carry out administrative tasks, such as system maintenance or repairs. `root` is quite risky for everyday use, as root can delete files irreversibly.

For verification purposes, the password must be entered twice as in Figure 4.8 on the following page. Be particularly careful not to forget the `root` password. It cannot be retrieved later.
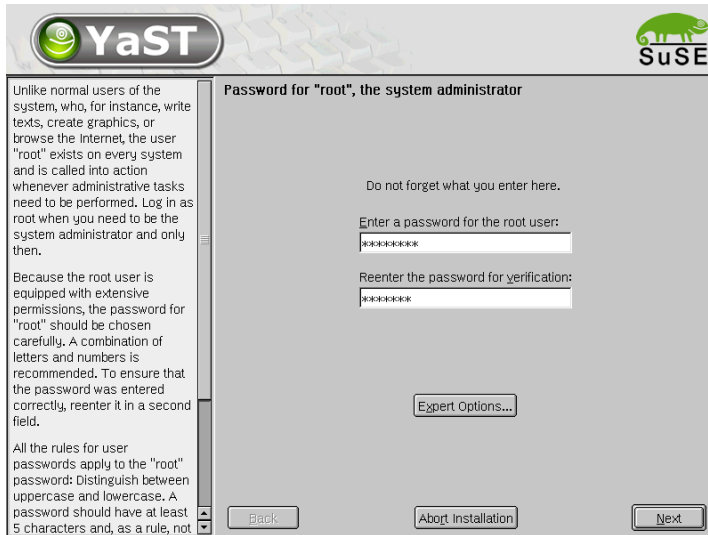
Figure 4.8: Setting the Root Password

User Name and Password

Linux is an operating system that allows several users to work on the same system at the same time. For it to function smoothly, each user needs a user account, which allows them to log in to the system then to log out again when finished. Setting up user accounts provides a strong basis for operating security. Normal users cannot change or delete files needed for the system to work properly. Similarly, a user's personal data may not be accessed, modified, or deleted by other users. Each user can set up his own working environment with preferred applications and settings. When users log in to the Linux system, they find their personal environments unchanged.

Create such a user account yourself using the dialog as shown in Figure 4.9 on the next page. Enter the user's first name and last name. Also

Figure 4.9: *Entering the User Name and Password*

specify the user name (login). If you cannot think of a suitable user name, click 'Suggestion' and the system will automatically generate one for you.

Finally, enter a password for the user, which you must repeat to confirm. The user name tells the system who you are and the password verifies your identity.

Caution

Memorize your user name and password, as you will need this information every time you log in. To provide effective protection, a password should be between five and eight characters long. The maximum length for a password is 128 characters. However, if no special modules are loaded, only the first 8 characters are used to identify the password. Linux distinguishes between lowercase and uppercase letters in the password. Accented characters are not allowed. Special characters (such as *, ., # , ;) and the digits 0–9 may be used.

Caution

Monitor Settings

This shows graphics card and screen with a suitable configuration. In most cases, you can accept the suggestion. However, you can customize color depth, resolution, and the image repetition rate manually.

The settings will be tested once you have accepted the suggestion or entered your changes.

When you click 'Change', you have the option of configuring the graphical interface. For this purpose, the program SaX2 is started.

Hardware Configuration

Figure 4.10: Configuring the System Components

Once your graphics card has been configured, you will see a screen like that shown in Figure 4.10. Now you have the option of configuring your system hardware, such as printer or sound card. The hardware configuration can also be done after the installation. Start the hardware configuration by clicking each component. YaST2 will then automatically detect and configure the hardware. Click 'Finish installation' when completed.

Graphical Login

SuSE Linux Enterprise Server is now installed and configured so you can
log in to your system. Your monitor displays the graphical login, as in Fig-
ure 4.11. Enter the user name specified earlier and the respective password to
log in to your system.
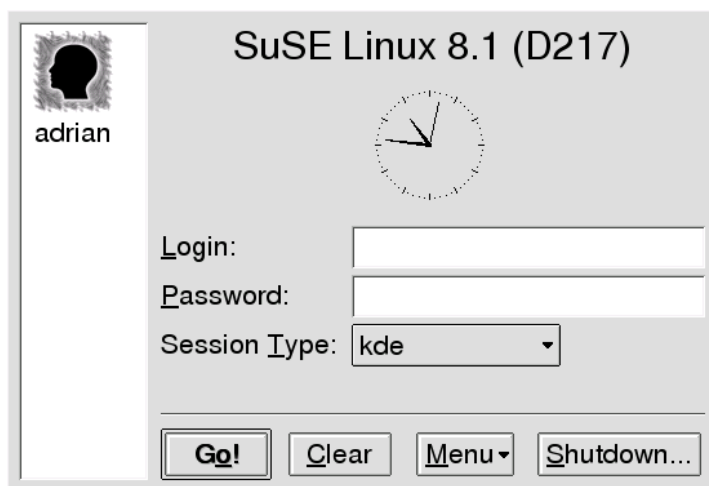
Figure 4.11: Logging In

AutoYast2: Automatic Linux Installation and Configuration with YaST2

To ease and automate Linux installations, SuSE offers AutoYaST2. AutoYaST2 lets the user create a configuration for a system to install and automatically performs the installation if the configuration is provided to during installation.

Introduction

Using AutoYaST2, multiple systems sharing the same environment and hardware performing similar tasks can easily be installed in parallel. A configuration file (the "control file") is created using existing configuration resources and it can be easily tailored for any specific settings.

This chapter guides you through the three steps of autoinstallation:
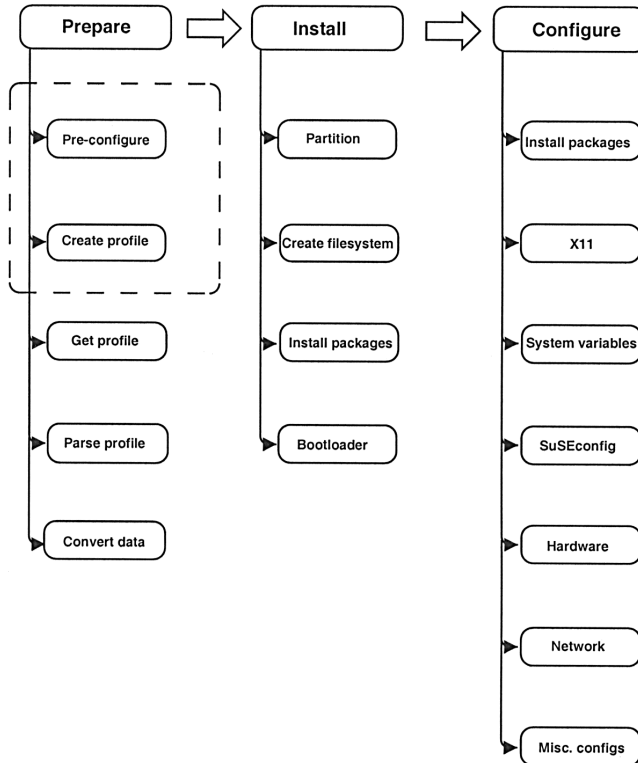
Figure A.1: *The Autoinstallation Process*

Prepare All relevant information about the target system is collected and converted to the appropriate directives of the control file. The control file is transferred onto the target system where its directives will be parsed and transformed to YaST2 conforming data.

Install YaST2 follows the instructions given in the control file and installs a base system.

Configure YaST2 and some user-defined postinstall scripts accomplish the system configuration

---

Tip

When autoinstalling using AutoYaST2, a good knowledge of the YaST2 installation procedure and a basic knowledge of XML will prove helpful. For a detailed reference of the XML syntax and numerous examples, refer to `/usr/share/doc/packages/autoyast2`

Tip

Collect Information

You need to collect information about the machines to install. This includes hardware data and network information. Make sure you know the following about the machines to install:

- Hard disk types and sizes

- Graphical interface and attached monitor, if any

- Network interface and MAC address if known (i.e., when using DHCP)

With these parameters, you are ready to create a profile of your systems to control the autoinstallation process.

Basics on the Control File

The control file is a per-host configuration description. It consists of sets of resources with properties, including support for complex structure representations such as lists, records, trees, and large embedded or referenced objects.

The simplest way to create the control file is by using an editor. Use one of the many XML editors available or your favorite text editor with XML support (such as Emacs and Vim). However, it is not quite optimal to create the control file manually for large sets of machines and it should only be seen as an interface between the autoinstallation engine and the configuration management system (CMS).
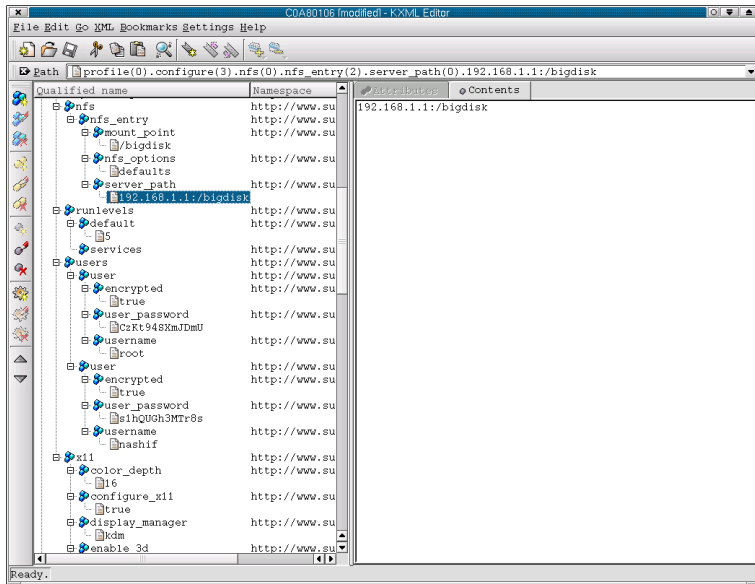
Figure A.2: *Editing the Control File with kxmledit*

Format

The XML configuration format provides a consistent file structure, which is easier to learn and remember when attempting to configure a new system. Using XML, you can eliminate (nearly) all of the control file parsing and error handling — an external XML parser can do that instead — (especially if it is a validating parser). To make sure the control file is well-formatted and the syntax is valid, run the control file through a validating parser before it is actually used for automatic installation. This is especially required if you prefer to edit the profile manually.

The following example shows a control file in XML format:

```xml
<?xml version="1.0"?>
<![CDATA[
<!DOCTYPE control_file SYSTEM
 "/usr/lib/YaST2/include/control-file.dtd">
 <profile
 xmlns="http://www.suse.de/1.0/cfns"
 xmlns:cfg="http://www.suse.de/1.0/cfgns">
 <install>
   <partitioning  config:type="list">
      <drive>
         <device>/dev/hda</device>
         <partition>
            <filesystem>ext2</filesystem>
            <size>520Mb</size>
            <mount>/</mount>
         </partition>
         <partition>
            <filesystem>reiser</filesystem>
            <size>1200Mb</size>
            <mount>/data</mount>
         </partition>
      </drive>
   </partitioning>
 </install>
 <configure>
   <scripts>
    <pre-scripts>
     <script>
        <interpreter>shell</interpreter>
        <filename>start.sh</filename>
         <source>
]]>
            &lt;![CDATA[
                #!/bin/sh
                echo "Starting installation"
                exit 0
            ]]&gt;
<![CDATA[
         </source>
     </script>
    </pre-scripts>
   </scripts>
 </configure>
</profile>
```

Output 1: XML Control File (Profile)

Structure

Below is an example of a basic control file container, the actual content of which is explained later on in this chapter.

```
<?xml version="1.0"?>
<!DOCTYPE control_file SYSTEM
 "/usr/lib/YaST2/include/control-file.dtd">
 <profile
 xmlns="http://www.suse.de/1.0/cfns"
 xmlns:config="http://www.suse.de/1.0/cfgns">

<!-- RESOURCES -->
</profile>
```

Output 2: Control File Container

The profile element (root node) contains one or more distinct resource elements. The permissible resource elements are specified in the DTD.

The root element in the control file can for example contain the following sub-keywords:

installation (Tag ⟨*install*⟩)

- Lilo configuration: lilo device, lilo type (Tag ⟨*bootloader*⟩)
- Partitioning: drives and partition plans (Tag ⟨*partitioning*⟩)
- General: Installation instructions, including all variables related to the client i.e. display, languages, keyboard etc. (Tag ⟨*general*⟩)

Network Network configuration for the client and servers providing services to the target client (Tag ⟨*networking*⟩)

Users user administration, including first user and root. (Tag ⟨*users*⟩)

User scripts: pre-configuration (Tag ⟨*pre-scripts*⟩)

User scripts: post-configuration (Tag ⟨*post-scripts*⟩)

Resources and Properties

A resource element either contains multiple and distinct property and re-source elements or contains multiple instances of the same resource element or is empty. The permissible content of a resource element is specified in the DTD.

A property element is either empty or contains a literal value. The permissible property elements and values in each resource element are specified in the DTD.

An element can be either a container of other elements (a resource) or have a literal value (a property), it can never be both. This restriction is specified in the DTD. A configuration component with more than one value must either be represented as some kind of embedded list in a property value or as a nested resource.

Nested Resources

Nested resource elements allow a tree like structure of configuration components to be built to any level.

```
...
<drive>
  <device>/dev/hda</device>
  <partitions>
    <partition>
       <size>1000</size>
       <mount>/</mount>
    </partition>
    <partition>
       <size>250</size>
       <mount>/tmp</mount>
    </partition>
  </partitions>
</drive>
....
```

Output 3: Nested Resources

In the example above the disk resource consists of a device property and a partitions resource. The partitions resource contains multiple instances of the partition resource. Each partition resource contains a size and mount property.

Although it is specified in the DTD that the partitions resource contains multiple instances, it is still required to specify this to avoid false data typing. Using the example above, imagine having a drive with only one partition. This will result in interpreting the partition resource as a property. To avoid this the following syntax must be used when defining multiple instances. For more information about type attributes, see next section.

```
...
<drive>
  <device>/dev/hda</device>
  <partitions config:type="list">
     <partition>
        <size>1000</size>
        <mount>/</mount>
     </partition>
     <partition>
        <size>250</size>
        <mount>/tmp</mount>
     </partition>
  </partitions>
</drive>
....
```

Output 4: Nested Resources with Type Attributes

Attributes

Global profile attributes are used to define meta-data on resources and properties. Attributes are used to define timestamps, access control, dynamic values and context switching. They are also used for naming and typing properties as shown in earlier sections.

> **Tip**
>
> Profile attributes are in a separate namespace so they don't have to be treated as reserved words in the default namespace. New ones can then be added without having to potentially alter existing profiles.
>
> **Tip**

Profile attributes are defined in the configuration namespace and must always be prefixed with config:. All profile attributes are optional. Most can be

used with both resource and property elements but some can only be used with one type of element which is specified in the DTD.

There are no ordering constraints on attributes and no significance should be interpreted from a specific ordering.

Attribute Names The name of a resource or property element is used for addressing and distinguishing multiple instances of the same element. The name of an element can be defined using the `config:name` attribute. By default for single instance elements the name is the same as the element type. By default for multiple instance elements the name is the index position number of the element. An element can only be addressed by its

Attribute Type The type of an element is defined using the `config:type` attribute. The type of a resource element is always RESOURCE, although this can also be made explicit with this attribute (to ensure correct identification of an empty element for example when there is no DTD to refer to). A resource element cannot be any other type and this restriction is specified in the DTD. The type of a property element determines the interpretation of its literal value. The type of a property element defaults to STRING, as specified in the DTD. The full set of permissible types is specified in the DTD.

The XML Document Type Definition (DTD)

Introduction

The purpose of a DTD is to define the legal building blocks of an XML document. It defines the document structure with a list of legal elements. A DTD can be declared inline in the XML document, or as an external reference.

XML provides an application independent way of sharing data. With a DTD, the application can use a standard DTD to verify that data that the user supplies is valid. A "Valid" XML document is a "Well Formed" XML document which conforms to the rules of a Document Type Definition (DTD).

In AutoYaST2, a DTD should is available to allow users to validate the control files before the installation process is initiated. The DTD can be also used with XML editors while editing the control file to avoid later errors.

An Example DTD

- A ⟨*drive*⟩ resource containing a ⟨*device*⟩ property and a ⟨*partitions*⟩ property represented as a nested resource.

- A ⟨*partitions*⟩ resource containing multiple instances of the ⟨*partition*⟩ property represented as a nested resource.

- A ⟨*partition*⟩ resource containing a ⟨*size*⟩ property and a ⟨*mount*⟩ property.

Below is the XML for an example node view profile for the above tree which includes a DTD which validates it.

```
<?xml version="1.0"?>
<!DOCTYPE profile [
<!ELEMENT profile (install)>
<!ELEMENT install (partitioning)>
<!ELEMENT install (drive+)>
<!ELEMENT drive (name,partitions)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT partitions (partition*)>
<!ELEMENT partition (size,mount)>
<!ELEMENT size (#PCDATA)>
<!ELEMENT mount (#PCDATA)>
]>
<profile>
.....
  <install>
    <partitioning config:type="list">
      <drive>
        <device>
           /dev/hda
        </device>
        <partitions>
          <partition>
            <size>1000mb</size>
            <mount>/</mount>
          </partition>
          <partition>
            <size>250mb</size>
            <mount>/tmp</mount>
          </partition>
        </partitions>
      </drive>
    </partitioning>
  </install>
.....
</profile>
```

Output 5: An Example DTD

Creating a New Control File

To create a control file, either use the configuration management system, which covers most of the features of the autoinstallation system, or use your favorite editor. In some cases, you may need to add some information manually after creating the control file with the configuration management system.

Make sure the configuration management system is installed (package `autoyast2`) and call it using the YaST2 Control Center or call it directly as `root` with the command `/sbin/yast2 autoyast` (make sure the DISPLAY variable is set correctly to start the graphical user interface instead of the text-based one).

Using the Configuration Management System

To create the control file for a specific system, a YaST2-based system is provided. This system depends on existing modules which are usually used to configure a system in regular operation mode — after SuSE Linux Enterprise Server is installed. The configuration management system lets you create control files easily and additionally lets you manage a repository of configurations for use in a networked environment and with multiple clients.

With some exceptions, almost all resources of the control file can be configured using the configuration management system. This system offers more flexibility. Additionally, many resources are configured with the same YaST2 modules as regular system configuration. New interfaces were created for special and complex configuration resources, such as `Partitioning` and `General Options`, to make it easier to access the information.

Using the configuration management system guarantees that the resulting control file is valid and can be used directly to start an automated installation.

Going Through the Configuration Management System

If you intend to create a new configuration, select the respective option from the main screen of the configuration management system. This opens the *File Management* screen where you can start a new or a edit an existing configuration. The files shown on the screen by default are complete configurations that can be used for autoinstallation without any change. These files are stored in the main configuration repository. If you are using classes to create your configuration, switch to the *Templates* view. (Templates are stored in the `templates` directory beneath the configuration repository.)

Figure A.3: The Configuration Management System

To start with the configuration, select an existing file using 'Edit' or start a new configuration by clicking 'New', which opens a pop-up asking for the new file name. This leads directly to the configuration options, which can be browsed in any desired order. You can choose to configure only those resources you need.

Clicking a configuration option shows a summary of the current configuration. It is possible to reset a configuration resource at any time. To configure a resource, click 'Configure'.

Import of Legacy and Foreign Configuration Files

Using AutoYaST2, import ALICE configuration files from previous SuSE releases and foreign autoinstallation systems (Kickstart). Consult Section *Red Hat Kickstart* on page 81 for details.

Using Classes

Using the configuration management system, define a set of classes. The class definition consists of the following variables for each class:

Figure A.4: Configuration Options

- Name: Class name

- Descriptions: Class description

- Order: Order (or priority) of the class in the stack of migration

Create as many classes as required. However, it is recommended to keep the set of classes as small as possible to keep the configuration management system concise. As an example, the following set of classes can be used:

- site: Classes describing a physical location or site.

- machine: Classes describing a type of machine or make.

- role: Classes describing the function of the machine to install.

- group: Classes describing a department or a group within a site or a location.

A file defined in a class can have the same syntax and format as the main profile XML file and represents a subset of the configuration. For example, to

Figure A.5: General Options

create a new profile for a special machine with a specific network interface, only the resource in the profile that controls the configuration of the network is needed. Having multiple network types, you can merge the one needed for a special type of hardware with other class files to create a new control file that corresponds to the defined classes.

Using Templates

Templates are control files that are not complete in their content and belong to one or several classes. To make a template installable, it has to run through a merge process, which sets all needed values according to the data available in the configurations within the classes.

If you define classes in the control file using the configuration management system, the file will not be saved in the repository. Instead, it will be installed in the `templates` directory in the repository.

Creating and Editing a Control File Manually

If you edit the control file manually, make sure it has a valid syntax. An easy way to check the syntax is by using some tools already available on the dis-

Figure A.6: *Defining Classes*

tribution. For example, to verify that the file is well formed, use the utility xmllint available with the libxml2 (`xmllint <control file>`).

If the control file is not well formed, for instance, if a tag is not closed, xmllint will report about the errors. Before continuing with the autoinstallation, fix any errors resulting from such checks. The autoinstallation process cannot be started with poorly formed control files.

A Closer Look at Profile Resources

This section features the most important parts of a control file for standard purposes. For information about the other options available, consult the XML reference and use the configuration management system.

General Options

This is a *required* section of the profile. General options include all the settings related to the installation process and the environment of the installed system. These resources include the following four properties required for almost any installation: `language`, `keyboard`, `time zone`, and `mouse`. If left

out, default values will be used that might not be compatible with how you want the system installed.

```
    <install>
...
      <general>
        <language>de_DE</language>
        <keyboard>
            <keymap>german</keymap>
            <rate>24</rate>
            <delay>24</delay>
            <numlock>on</numlock> <!-- on|off|bios -->
            <scrolllock>on</scrolllock> <!-- on|off -->
            <capslock>on</capslock> <!-- on|off|disables -->
        </keyboard>
        <clock>
          <timezone>US/Eastern</timezone>
          <utc config:type="boolean">true</utc>
          <ntp_servers config:type="list">
             <ntp_server>ntp.example.com</ntp_server>
          </ntp_servers>
        </clock>
        <mouse>
          <id>ps0</id>
          <device>/dev/psaux</device>
          <gpm>
             <protocol>ps2</protocol>
             <parameters></parameters>
          </gpm>
          <wheels config:type="integer">1</wheels>
          <buttons config:type="integer">1</buttons>
          <xemu3 config:type="boolean">true</xemu3>
        </mouse>
        <mode>
          <installation config:type="boolean">true</installation>
          <upgrade config:type="boolean">false</upgrade>
          <confirm config:type="boolean">false</confirm>
          <interactive_boot config:type="boolean">false</interactive_boot>
          <reboot  config:type="boolean">false</reboot>
        </mode>

      </general>
...
    </install>
```

Output 6: General Options

The reboot property in the mode resource is used to force a reboot after initial system setup and before the system is booted for the first time.

By default, the autoinstallation process must be confirmed by the user. The confirmation should be disabled if a full unattended installation is desired. This option is used to control the settings on a target system before anything is changed and can be used when debugging. It is set to `true` by default to avoid recursive installs when the system has to reboot after changing the kernel or if a reboot was requested in the control file.

Reporting

The `report` resource manages three types of pop-ups that may appear during installation.

- Pop-up messages (Normally only noncritical, informative messages)

- Warning messages (If something might go wrong)

- Error messages (In the case of an error)

```
    <install>
...
      <report>
          <messages>
              <show>true</show>
              <timeout>10</timeout>
              <log>true</log>
          </messages>
          <errors>
              <show>true</show>
              <timeout>10</timeout>
              <log>true</log>
          </errors>
          <warnings>
              <show>true</show>
              <timeout>10</timeout>
              <log>true</log>
          </warnings>
      </report>
...
    </install>
```

Output 7: *Reporting Behavior*

Depending on your experience, you can skip, log, and show (with time-out) those messages. It is recommended to show all messages with time-out. Warnings can be skipped in some places, but should not be ignored. The default setting in autoinstallation mode is to show all messages without logging and with a time-out of 10 seconds.

> **⌐ Note ───**
>
> **Critical System Messages**
> Not all messages during installation are controlled by the `report` resource. Some critical messages concerning package installation and partitioning will still show up despite your settings in the `report` section.
>
> **───────────────────────────────────── Note ⌐**

The Boot Loader

If you do not want to install a boot loader, specify this using the `write_bootloader` property (Boolean value). If you choose not to install a boot loader, make sure you create a boot disk or have another way to boot your system (such as a third-party boot loader). The default setting is to write the boot loader.

If you choose to install a boot loader, choose where to install it (the Master Boot Record or the first sector of the `/boot` partition). Install the boot loader on the MBR if you plan to use it as your boot loader. If you are using a different boot loader, install LILO on the first sector of the `/boot` partition and configure the other boot loader to boot SuSE Linux Enterprise Server.

If you need to pass any special parameters to the kernel when the system boots, enter them using the kernel parameters tag. Additionally, choose whether to use `linear` mode and whether to force the use of `lba32` mode.

Partitioning

Automated Partitioning For the automated partitioning to be completed, only the sizes and mount points of partitions are needed. All other data needed for successful partitioning can be calculated automatically.

If no partitions are defined and the specified drive is also the drive where the root partition should reside, the following partitions are created automatically:

/boot   Size of the /boot is determined by the architecture of the target system.

swap   Size of the swap partitions is determined by the amount of memory available in the system.

/(root partition)   Size of the / (root partition) is the space left after creating swap and /boot.

Depending on the initial status of the drive and how it was partitioned before, it is possible to create the default partitioning in the following ways:

Use free space   If the drive is already partitioned, it is possible to create the new partitions on the available space of the hard drive. This requires the availability of enough space for all selected packages in addition to swap.

'Reuse all available space'   This option leads to the deletion of existing partitions.

'Reuse all available Linux partitions'   This option leads to the deletion of existing Linux partitions. All other partitions (such as Windows) will be kept.

'Reuse only specified partitions'   This option leads to the deletion of the specified partitions. The selection of the partitions scheduled for deletion should be started from the last available partition.

If the target system has multiple drives, all drives should be identified with their device names and additional information related to the above-mentioned behavior.

Partition sizes can be given in gigabytes, megabytes, or can be set to a flexible value using the keywords auto and max. max is used to fill a partition to the maximum available space on a drive, which means that the partition is the last one on the drive. auto can be used to determine the size of swap or boot partitions that depend on memory and type of system.

For setting a fixed size, use the format of these examples. 1gb will create a 1 GB partition. 1500mb will create a 1.5 GB partition.

Logical Volume Manager (LVM)   To configure LVM, first create a physical volume using the normal partitioning method described above.

The following example shows how to prepare for LVM in the partitioning resource:

```
....
<partitioning config:type="list">
    <drive>
        <device>/dev/sda</device>
        <use>all</use>
        <partitions config:type="list">
            <partition>
                <size>auto</size>
                <lvm_group>system</lvm_group>
            </partition>
        </partitions>
    </drive>
</partitioning>
.....
```

Output 8: *Creating LVM Physical Volumes*

The last example creates an unformatted partition on device /dev/
sda1 of the type LVM with the volume group system. The partition
created will use all available space on this drive.

The logical volumes should be defined in the lvm resource. Currently,
it is not possible to configure LVM using the configuration management
system. Instead, it is required to add the resource manually as shown
in the following example.

```
....
  <lvm config:type="list">
    <lvm_group>
     <lvm_name>system</lvm_name>
     <pesize>4M</pesize>
     <logical_volumes config:type="list">
       <lv>
         <lv_name>usrlv</lv_name>
         <lv_size>500mb</lv_size>
         <lv_fs>reiser</lv_fs>
         <lv_mount>/usr</lv_mount>
       </lv>
       <lv>
       <lv_name>optlv</lv_name>
```

```
        <lv_size>1500mb</lv_size>
        <lv_fs>reiser</lv_fs>
        <lv_mount>/opt</lv_mount>
      </lv>
      <lv>
        <lv_name>varlv</lv_name>
        <lv_size>200mb</lv_size>
        <lv_fs>reiser</lv_fs>
        <lv_mount>/var</lv_mount>
      </lv>
    </logical_volumes>
    </lvm_group>
  </lvm>
...
```

Output 9: *LVM Logical Volumes*

Software RAID

Using AutoYaST2, you can create assembled software RAID devices. The following RAID levels are supported:

RAID 0 This level increases your disk performace. There is *no* redundancy in this mode. If one of the drives crashes, data recovery will not be possible.

RAID 1 This mode has the best redundancy. It can be used with two or more disks. This mode maintains an exact copy of all data on all disks. As long as at least one disk is still working, no data is lost. The partitions used for this type of RAID should have approximately the same size.

RAID 5 This mode combines management of a larger number of disks and still maintains some redundancy. This mode can be used on three disks or more. If one disk fails, all data is still intact. If two disks fail simultaneously, all data is lost.

Multipath This mode allow access to the same physical device over multiple controllers for redundancy against a fault in a controller card. This mode can be used with at least two devices.

As with LVM, you need to create the RAID partitions first and assign the partitions to the RAID device you want to create and you need to specify whether a partition or device should be configured in the RAID or if it should configured as a spare device.

The following example shows a simple RAID 1 configuration:

```
....
    <partitioning config:type="list">
       <drive>
         <device>/dev/hdc</device>
         <partitions config:type="list">
           <partition>
             <filesystem_id config:type="integer">253</filesystem_id>
             <format config:type="boolean">false</format>
             <raid_name>/dev/md0</raid_name>
             <size>4gb</size>
           </partition>

           <!-- Here come the regular partitions, i.e. / and swap -->

         </partitions>
         <use>all</use>
       </drive>
       <drive>
         <device>/dev/sda</device>
         <use>all</use>
         <partitions config:type="list">
           <partition>
             <filesystem_id config:type="integer">253</filesystem_id>
             <format config:type="boolean">false</format>
             <raid_name>/dev/md0</raid_name>
             <raid_type>raid</raid_type>
             <size>4gb</size>
           </partition>
         </partitions>
       </drive>
    </partitioning>

    <raid config:type="list">
       <device>
         <raid config:type="integer">md0</raid>
         <parity_algorithm>left-asymmetric</parity_algorithm>
         <persistent_superblock>true</persistent_superblock>
         <raid_type>raid1</raid_type>
         <filesystem_id config:type="integer">131</filesystem_id>
         <chunk_size config:type="integer">4</chunk_size>
       </device>
    </raid>
....
```

Software

Package Selections Choose between three different types of package selections:

- Use a predefined package base selection, such as `default`, `development`, or `default+office`, in addition to several `Add-on` selections.

- Custom package selection — package selection of an existing system using the rpm command or similar tools.

- Additional local packages, like custom packages (non-SuSE packages) and packages for initial system setup and configuration.

In the control file, packages and package selections are described as the following:

```
....
<software>
   <base>Minimal</base>
   <addons  config:type="list">
       <addon>Kde</addon>
   </addons>
   <packages  config:type="list">
       <package>apache</package>
       <package>sendmail</package>
   </packages>
</software>
....
```

Output 11: Software Selection in the Control File

A list of possible pre-defined selections can be found on the first CD-ROM in the directory `/suse/setup/descr`. You can install one base selection and additionally one or multiple add-on selections.

Custom package selections In addition to the predefined selections, you can create custom selections by providing a customized selection file in the selection directory. The selection files have a special format and any

AutoYast2: Automatic Linux Installation and Configuration with YaST2

additional selection file must conform to this format. Otherwise, YaST2 will not be able to read it.

For further information on the selection file, consult the documentation in the `yast2-packagemanager-devel` package.

After creating a selection file, add it to the configuration as described above.

```
....
<software>
   <base>My</base>

</software>
....
```

Output 12: Customized Software Selection

The file `My.sel` should have the following format:

```
# SuSE-Linux-Package-Selection 3.0 -- (c) 2002 SuSE Linux AG
# generated on Sat Aug 10 17:55:42 UTC 2002

=Ver: 3.0

# name version release
=Sel: Kde-Desktop <version>

# size in bytes (pkgsize instsize)
=Siz: 123456 1234567

# Summary
...
=Sum.de: KDE Desktop-Umgebung
=Sum.en: KDE Desktop Environment
=Sum.es: Entorno Graf¡fico KDE
=Sum.fr: Environnement de bureau KDE
=Sum.gl: KDE Desktop Environment
=Sum.hu: KDE grafikus munkakrnyezet
=Sum.it: Ambiente Desktop KDE
=Sum.tr: KDE Desktop Environment
...

# selections required for installation
=Req: X11 Basis-Sound
# conflicting selections
=Con: Minimal
```

```
# category, add-on or base
=Cat: addon

# visibility of selection (for user interface)
=Vis: true

# list of packages to install
+Ins:
SDL
aalib
alsa

...
smpppd
unixODBC
wvdial
-Ins:

# list of packages to install if given language is active

+Ins.cs:
kde3-i18n-cs
-Ins.cs:

+Ins.da:
kde3-i18n-da
-Ins.da:

+Ins.de:
kde3-i18n-de
-Ins.de:

...
```

Output 13: A Package Selection File

Installing additional and customized packages In addition to the packages
available for installation on the CDs, you can add non-SuSE packages
including non-SuSE kernels. Customized kernel packages must be com-
patible with the SuSE packages and must install the kernel files to the
same locations.

Unlike earlier versions, to install custom and external packages there is
no need for a special resource in the control file. Instead you need to
re-create the package database and update it with any new packages or
new package versions in the source repository.

A script is provided for this task which will query packages available
in the repository and create the required package database.

The advantage of this method is that you can keep an up-to-date repository with fixed and updated package (i.e. from SuSE ftp server). Additionally this method makes the creation of custom CDs easier.

Services and Runlevels

With the runlevel resource, set the default runlevel and specify in detail which system services to start in which runlevel.

The `default` property specifies the default runlevel of the system. Changes to the default runlevel take effect next time the target system is booted. After the installation is completed, the system has runlevel 5, which is `Full multiuser with network and xdm`. If you have configured a system without X11, it is recommended to reboot the system after the first stage using the `reboot` property in the `general` resource.

Specify the runlevels in which a service should run using a space-delimited list of the runlevels as shown in the following example.

```
<configure>
....
  <runlevels>
   <default>3</default>
   <services config:type="list" >

    <service>
     <service_name>at</service_name>
     <service_start>3 5</service_start>
     <service_stop>2 3 5</service_stop>
    </service>
    <service>
     <service_name>portmap</service_name>
     <service_start>3 5</service_start>
     <service_stop>2 3 5</service_stop>
    </service>
   </services>
  </runlevels>
....
</configure>
```

Output 14: Runlevel Configuration

Network Configuration

Network devices, DNS, and routing Network configuration is used to connect a single SuSE Linux Enterprise Server workstation to an ethernet-based LAN or to configure dial-up connections. More complex configuration (such as multiple network cards and routing) is also provided. With this module, it is possible to configure ethernet controllers and token ring controllers. To configure network settings and activate networking automatically, one global map is used to store the whole network configuration.

```
 <configure>
.....
    <networking>
      <dns>
        <dhcp_hostname config:type="boolean">true</dhcp_hostname>
        <dhcp_resolv config:type="boolean">true</dhcp_resolv>
        <domain>local</domain>
        <hostname>linux</hostname>
      </dns>
      <interfaces config:type="list">
        <interface>
          <bootproto>dhcp</bootproto>
          <device>eth0</device>
          <module>tulip</module>
          <options>options=0</options>
          <startmode>onboot</startmode>
        </interface>
      </interfaces>
      <routing>
        <ip_forwarding config:type="boolean">false</ip_forwarding>
        <routes config:type="list">
          <route>
            <destination>default</destination>
            <device>-</device>
            <gateway>192.168.1.240</gateway>
            <netmask>-</netmask>
          </route>
        </routes>
      </routing>
    </networking>
....
 </configure>
```

Output 15: Network Configuration

NIS The target machine can be set up as an NIS client. Specify multiple servers by using the `list` attribute (`config:type="list"`).

```
<configure>
...
<nis>
  <nis_broadcast config:type="boolean">true</nis_broadcast>
  <nis_broken_server config:type="boolean">true</nis_broken_server>
  <nis_domain>test.com</nis_domain>
  <nis_local_only config:type="boolean">true</nis_local_only>
  <nis_servers config:type="list">
    <nis_server>192.168.1.1</nis_server>
  </nis_servers>
  <start_autofs config:type="boolean">true</start_autofs>
  <start_nis config:type="boolean">true</start_nis>
</nis>
...
</configure>
```

Output 16: Network configuration: NIS

NIS+ If you activate NIS+, the data of the NIS+ server will be added to
/etc/hosts. Keyserv and the NIS+ cache manager will be started
and the NSS and PAM configuration will be modified to use NIS+ and
set the secret key of a user.

```
<configure>
...
        <nisplus>
            <start_nisplus>true</start_nisplus>
            <nisplus_domain>Domain</nisplus_domain>
            <nisplus_address>Address</nisplus_address>
            <start_autofs>true</start_autofs>
        </nisplus>
...
</configure>
```

Output 17: Network configuration: NIS+

LDAP client The installed machine can be set up as an LDAP client to au-
thenticate users with an Open LDAP server. Required data is the name
of the search base (base DN, e.g., dc=mydomain,dc=com) and the IP
address of the LDAP server (e.g., 10.20.0.2). If LDAP is activated,
NSS and PAM will be configured accordingly to use LDAP for user au-
thentication.

```
<configure>
...
        <ldap>
            <start_ldapclient>false</start_ldapclient>
            <ldap_domain>domain</ldap_domain>
            <ldap_address>192.168.1.1</ldap_domain>
        </ldap>
...
</configure>
```

Output 18: Network Configuration: LDAP client

NFS Batch operation of the NFS client module is not currently available (`nfs_write` with options), but the routine `nfs_client_save` can be used for autoinstallation and `/etc/fstab` configuration.

```
<configure>
...
        <nfs config:type="list">
          <entry>
            <server_path>server:/space</server_path>
            <mount_point>/space</mount_point>
            <options>default</options>
          </entry>
        </nfs>
...
</configure>
```

Output 19: Network Configuration: NFS

Mail Configuration (Sendmail or Postfix) For the mail configuration of the client, the existing module for mail configuration available in the running system is used. This module enables very sophisticated mail configuration and should be used instead of editing the mail resource manually.

```
<configure>
...
<mail>
 <mta>sendmail</mta>
 <connection_type>permanent</connection_type>
 <local_domains config:type="list"></local_domains>
 <outgoing_mail_server></outgoing_mail_server>
 <from_header ></from_header>
 <masquerade_other_domains config:type="list"></masquerade_other_domains>
 <masquerade_users config:type="list"></masquerade_users>
 <fetchmail config:type="list"></fetchmail>
 <aliases config:type="list"></aliases>
 <merge_aliases></merge_aliases>
 <virtual_users config:type="list"></virtual_users>
</mail>
...
</configure>
```

Output 20: Mail Configuration

Security Settings

Using the features of this module, change the local security settings on the target system. The local security settings include the boot configuration, login settings, password settings, some user creation settings, and file permissions.

Configuring the security settings automatically corresponds to the 'Custom Settings' in the security module available in the running system, which lets you create your own customized configuration.

Password Setting Options   Change the various password settings. These settings are mainly stored in the /etc/login.defs file.

Use this resource to activate one of the encryption methods currently supported. If not set, DES is configured.

DES, the Linux default method, works in all network environments, but it restricts you to passwords no longer than eight characters. MD5 allows longer passwords, providing more security, but some network protocols do not support this and you may encounter problems with NIS. Blowfish is also supported.

The system can also be set up to check password length and security.

Boot settings   Using the security resource, you can change various boot settings.

- **How to interpret Ctrl + Alt + Del** When someone at the console has pressed the Ctrl + Alt + Del key combination, the system usually reboots. Sometimes, it is desirable to ignore this event, for example, when the system serves both as workstation and server.

- **Shutdown behavior of KDM** Set who is allowed to shut down the machine from KDM.

Login settings Change various login settings. These settings are mainly stored in the /etc/login.defs file.

New user settings (useradd settings) Set the minimum and maximum possible user ID and set the minimum and maximum possible group ID.

Users

The root user and at least one normal user can be added during install using data supplied in the control file. User data and passwords (encrypted or clear text) are part of the configure resource in the control file.

At least the root user should be configured during autoinstallation, which will ensure you will be able to login after installation is finished and ensure others cannot log in to the system (if password is not set).

The two users in the following example are added during system configuration.

```
<configure>
...
   <users config:type="list">
      <user>
         <username>root</username>
         <user_password>password</user_password>
         <encrypted>true</encrypted>
         <forename/>
         <surname/>
      </user>
      <user>
         <username>nashif</username>
         <user_password>password</user_password>
         <encrypted>true</encrypted>
         <forename>Anas</forename>
         <surname>Nashif</surname>
      </user>
   </users>
```

```
...
</configure>
```

Output 21: *User Configuration*

The last example shows the minimal information required for adding users. More options are available for a more customized user account management. The data in /etc/default/useradd is used to determine the home directory of the user to be created in addition to other parameters. Please see the resource reference section for more options.

Custom User Scripts

By adding scripts to the autoinstallation process, customize the installation for your needs and take control in different stages of the installation.

In the autoinstallation process, three types of scripts can be executed:

Preinstall scripts   Executed before YaST2 does any real change of the system (before partitioning and package installation).

Postinstall scripts   These scripts are executed after YaST2 has completed the installation and after it has booted the system the first time.

Chroot environment postinstall scripts   Chroot scripts are executed in the chroot environment before YaST2 boots for the first time and before the boot loader is installed.

All but the preinstall scripts can be written in either shell or perl script language. When added to the control file manually, the scripts have to be included in a CDATA element to avoid confusion with the file syntax and other tags defined in the control file.

Please see the resource reference for more options.

```
  <post-scripts config:type="list" >
   <script>
    <filename>post.sh</filename>
    <interpreter>shell</interpreter>
    <source>
]]>
&lt;![CDATA[
#!/bin/sh
```

```
echo "Do something useful"

]]&gt;
<![CDATA[
    </source>
   </script>
  </post-scripts>
```

Output 22: *Post-script Configuration*

After installation is finished, the scripts and the output logs can be found in the directory /var/adm/autoinstall. The scripts are located in scripts and the output log of the scripts is located in the log directory.

The log is the output resulting when executing the shell scripts using the following command:

```
/bin/sh -x <script_name> 2&> /var/adm/autoinstall/logs/<script_name>.log
```

System Variables (sysconfig)

Using the sysconfig resource, define configuration variables in the sysconfig repository (/etc/sysconfig) directly. Using the sysconfig variables, fine-tune many system components and environment variables exactly to your needs.

Consult the Administration Manual for more details about the many configuration options available in /etc/sysconfig.

The following example shows how a variable can be set using the sysconfig resource. To configure a variable in a sysconfig file, the following syntax is used:

```
<sysconfig config:type="list" >
   <sysconfig_entry>
    <sysconfig_key>XNTPD_INITIAL_NTPDATE</sysconfig_key>
    <sysconfig_path>xntp</sysconfig_path>
    <sysconfig_value>ntp.host.com</sysconfig_value>
   </sysconfig_entry>
   <sysconfig_entry>
    <sysconfig_key>HTTP_PROXY</sysconfig_key>
    <sysconfig_path>proxy</sysconfig_path>
    <sysconfig_value>proxy.host.com:3128</sysconfig_value>
   </sysconfig_entry>
   <sysconfig_entry>
    <sysconfig_key>FTP_PROXY</sysconfig_key>
```

```
  <sysconfig_path>proxy</sysconfig_path>
  <sysconfig_value>proxy.host.com:3128</sysconfig_value>
 </sysconfig_entry>
</sysconfig>
```

Output 23: Sysconfig configuration

Adding Complete Configurations

For many applications and services, you might have prepared a configuration
file that should be copied in a complete form to some location in the installed
system. This is the case, for example, if you are installing a web server and
have a "ready to go" `httpd.conf` file.

Using this resource, embed the file into the control file by specifying the final
path on the installed system. YaST2 will copy this file for you on the speci-
fied location.

```
 <files config:type="list">
     <config_file>
       <file_path>/etc/httpd/httpd.conf</file_path>
       <file_contents>
]]>
&lt;![CDATA[
some content
]]&gt;
<![CDATA[
       </file_contents>
     </config_file>
  </files>
```

Output 24: Dumping Files into the Installed System

Miscellaneous Hardware and System Components

In addition to the core component configuration, like network authentication
and security, AutoYaST2 offers a wide range of hardware and system config-
uration which is available by default on any system installed manually and
in an interactive way. For example, it is possible to configure printers, sound
devices, TV cards and any other hardware components which have a module
within YaST2.

Printer Although Printer configuration, like other configurations can be done manually, it is recommended to use the Configuration Management System to create such a configuration because of the complexity and the range of options offered by such modules.

Using the Configuration Management System will guarantee that the options provided are consistent. The following is an example of a configuration section which was created using the Configuration Management System.

```
 <configure>
....
    <printer>
      <default>lp</default>
      <printcap config:type="list">
        <printcap_entry>
          <cups-state>void</cups-state>
          <ff config:type="boolean">true</ff>
          <info></info>
          <location></location>
          <lprng-state>changed</lprng-state>
          <name>lp</name>
          <options>
            <job-sheets>none,none</job-sheets>
          </options>
          <raw config:type="boolean">true</raw>
          <type>yast2</type>
          <uri>parallel:/dev/lp0</uri>
        </printcap_entry>
      </printcap>
    </printer>
....
 </configure>
```

Output 25: *Printer Configuration*

Sound Devices An example of sound configuration created using the Configuration Management System is shown below.

```
 <configure>
....
    <sound>
      <autoinstall config:type="boolean">true</autoinstall>
      <modules_conf config:type="list">
        <module_conf>
          <alias>snd-card-0</alias>
```

```
            <model>M5451, ALI</model>
            <module>snd-ali5451</module>
            <options>
              <snd_enable>1</snd_enable>
              <snd_index>0</snd_index>
              <snd_pcm_channels>32</snd_pcm_channels>
            </options>
            <unique_key>uniq.virtual</unique_key>
          </module_conf>
        </modules_conf>
        <volume_settings config:type="list">
          <listentry>
            <Master config:type="integer">75</Master>
          </listentry>
        </volume_settings>
      </sound>

  ....
   </configure>
```

Output 26: *Sound Configuration*

Specifying the Source of Installation Data

Autoinstalling a Loose System

The best way to autoinstall one system without any network connection is
by using the standard CDs that come with the SuSE Linux Enterprise Server
box. Using the CDs in combination with a floppy disk can let you get started
with AutoYaST2 quickly without spending much time configuring server and
network environment.

As discussed in the following sections, you need to prepare a floppy disk
with a profile file containing all data needed for YaST2 to complete the au-
toinstallation process.

Create the control file as described in the previous section and name it
`autoinst.xml`. Copy `autoinst.xml` to the floppy by either mounting the
floppy or by using the mtools.

```
mcopy autoinst.xml a:
```

Network Installations

User intervention depends much on how the server side of a network installation is prepared. In a full network installation, the user only needs to turn on the client to initiate an autoinstallation process. This can also be automated using different technologies available today (Remote Power Management) or by using Wake-on-LAN (WOL). More information on WOL can be found at http://www.scyld.com/expert/wake-on-lan.html

Setting up an Installation Repository

The server can serve as a configuration repository. The clients will need to access the server resources to boot, install packages, and so on. To achieve this, various network services must be properly set up.

The installation server exports the SuSE Linux Enterprise Server distribution's files via NFS. Create a directory on a file system with enough free space (several gigabytes) and copy the contents of the CDs into this directory. This directory is then exported via NFS (via an appropriate entry in /etc/exports). The following steps describe how to create an installation repository.

Log in to the machine designated as installation server and create a directory to hold the SuSE Linux Enterprise Server distribution files, for example /usr/local/SuSE/current. In our example /usr/local/SuSE/current is the base directory for the SuSE Linux Enterprise Server distribution. The location of this directory can be specified in the info file or on the command line of the kernel (see below) using the install keyword (i.e., install=nfs://192.168.1.1/usr/local/SuSE/current).

Next, copy the files from the all CDs into the current directory or just copy those CDs required for the installation. Make sure all packages needed for the installation are copied. Make sure the hidden files (.<file name>)files on the CD-ROM are also copied. They serve as identification of the media. Use the following commands to copy the CDs.

```
mount /cdrom
cd /cdrom && cp -va . /usr/local/SuSE/current ; cd -
umount /cdrom
```

Repeat this sequence for all other CDs. The directory can have two different structures that can be used for installation:

- The contents of all CDs are copied into one directory creating a single directory structure with a subdirectory `suse` including all the package groups. This type of structure is recommended as it is easier to manage and will provide a true, single source installation medium.

  To make the directory look like a single medium for the client, modify the package database available on the CDs by replacing the reference to the different CDs to one single CD, which is in our case CD number 1.

  Use the following "one-liner" to modify the installation path in the text-based package database available in the directory `suse/setup/descr`:

  ```
  cd /usr/local/SuSE/current/suse/setup/descr
  perl -pi -e 's/InstPath:\t0[2|3|4|5|6|7]/InstPath:\t01/' common.pkd
  cd -
  ```

- Copy the CDs into subdirectories named after the CD number, such as CD1 and CD2. Using this structure, you will still be able to perform NFS installations, but the single directories will be treated as if they were different media.

After you have copied the CDs into the installation directory, make sure it is exported via NFS. Do that in YaST2 with the NFS server module.

Additionally, modify the following services to start every time the system boots.

- nfsserver

- portmap

Setting up a Configuration Repository

A configuration repository holds the control file for multiple machines. The control files can have any file names, which have to be specified at the boot time of a client. To avoid supplying the profile name for every client, define the directory of the control files. If a directory is specified, the client tries to load a file with a name matching its IP address in hex mode. (See *Control File Retrievable via HTTP* in Section *Invoking the Autoinstallation Process* on page *73*.). This has the advantage that you will be dealing with consistent file names rather than IPs as file names, which might lead to some confusion.

The configuration repository is the same directory defined if you are using the configuration management system for creating control files.

HTTP Repository To use the HTTP protocol to retrieve control file while autoinstalling, you need a working HTTP server on the server side. Install `Apache` or your favorite web server and enable it using `YaST2`. Normally, the web server root directory resides in `/usr/local/httpd/htdocs`, so you need to create a subdirectory below the root directory of the web server as your configuration repository.

NFS Repository Create a directory and make it available via NFS to the clients by exporting it. This directory may, for example, be in the same place as where you have copied the CDs (`/usr/local/SuSE`).

TFTP Repository By default, the TFTP directory is available under `/tftpboot`, which can also contain boot images if you are booting over network. Do not forget to enable tftp in the inetd configuration file (`/etc/inetd.conf`). `Inetd` configuration can be done using `YaST2`.

Boot Management

Booting the Target System

A target system can be booted in different ways. Booting the machine and initiating the autoinstallation process is as important as the installation itself. Depending on how many target systems to install, the following methods are supported:

Floppy Not recommended in complex network environments. A floppy cannot hold all necessary information needed to set up a machine for installation. A floppy should only be used in very special cases and, if it is used, the default boot floppy components supplied with SuSE must be modified to match your exact needs. However, the floppy can be used to store the control file and other information needed for installation.

CD-ROM Use the original SuSE CD-ROMs in combination with other media, for example, with a floppy to hold the control file or with a network where the control file can be located. It is also possible to create customized CD-ROMs to hold only the packages needed in addition to the control file. This method requires creation of CD-ROMs every time you change the configuration.

Network Using the network to boot a target system is the most convenient way to autoinstall. Booting over the network and holding a repository of control files on an accessible server can be very flexible, especially when configuring different types of systems with different roles and hardware.

Booting the Client

There are different methods for booting the client. The computer can boot from its network interface card (NIC) to receive the boot images via DHCP or TFTP or a suitable kernel and an initrd image can be loaded from a floppy or a customized bootable CD-ROM.

Booting from Floppy

For testing and rescue purposes or because the NIC does not have a PROM or PXE, build a boot floppy to use with AutoYaST2. Using a floppy to initiate an autoinstall process is limited due to the amount of data a floppy can contain. However, it is still possible to use floppies when autoinstalling a single, disconnected machine.

Floppies can be used to store the control file, especially when using the original SuSE CD-ROMs. Using the kernel command line, the user can specify the location of the control file on the floppy. (See *Control File on a Floppy* in Section *Invoking the Autoinstallation Process* on page 73).

Even without the special command line option `autoyast`, it is still possible to initiate the autoinstall process by placing a control file on a floppy with a special name (`autoinst.xml`). YaST2 will check for `autoinst.xml` on start-up and, if found, change from interactive to automated installation.

Booting from a Network Card

For administrative purposes, booting from network card (NIC) is much more suitable than booting from floppy. To use this boot method, the client's NIC needs a boot PROM that is able to communicate with a DHCP server to receive communication-related configuration parameters, such as network addresses, and that is capable of communicating with a TFTP server to get a boot image.

Etherboot and Netboot Etherboot and netboot are capable of creating a PROM binary (which must still be programmed onto a PROM) and a corresponding "tagged" TFTP boot image that includes a kernel and an

initial ramdisk. Some tools exist that help test a boot PROM image. In fact ,the support utilities are pretty much common to both etherboot and netboot.

http://etherboot.sourceforge.net/
http://www.han.de/~gero/netboot.httml

Using PXE boot PROMs Another alternative to etherboot and netboot is to use a PXE-compliant boot PROM. PXE (Preboot Execution Environment) is a protocol designed by Intel that allows computers to boot through the network. PXE is stored in the ROM of new generation network cards. When the computer boots, the BIOS loads the PXE ROM in memory and executes it. A menu is displayed, allowing the computer to boot an operating system loaded through the network.

http://developer.intel.com/ial/WfM/Wfmspecs.htm

To install a client via Pre-Boot Execution Environment (PXE) you don't need a PXE server! PXE uses a BOOTP request to get an IP address and other network information and a bootloader program to the client. You can either use a BOOTP server for doing this or a DHCP and TFTP server.

In the following sections, you will find a description of how DHCP and TFTP should be set up to make PXE installations possible.

- **DHCP** Install the DHCP server from ISC (http://www.isc.org/) by using the package available in the SuSE distribution. Configure the DHCP server parameter in /etc/sysconfig/dhcpd and make sure you have a working configuration file /etc/dhcpd.conf.

- **PXE Bootloader** PXE can load a program into the client's memory and start it. The bootloader then loads its configuration file via TFTP from the the server defined in *next-server* (in the dhcpd.conf file example above).

The bootloader configuration file determines whether a client boots from its local hard disk or over the network.

```
default linux
serial 0,9600n8
label linux
kernel linux
append console=ttyS0,9800
console=tty0 load_ramdisk=1
initrd=initrd
autoyast=nfs://nfsserv/file.xml
```

Please note that the "append console ... file.xml" statement has to be written in one line.

Boot from local hard disk (filename default):

```
default linux
label linux
localboot 0
```

Output 28: *Configuration File for PXELINUX Local Boot*

`pxelinux.0` tries to read several configuration files. It uses the first one it finds. The filenames it looks for are determined by the IP address of the client it is running on. It converts the four decimal number parts of an IP address (they are divided by dots) into hexadecimal numbers and concatenates them. Example: IP address 192.168.0.11 gets converted into C0 A8 00 0B (without the spaces).

The search for files starts at *C0A8000B* and proceeds by removing one digit from the right (leaving *C0A8000*) and so forth. When all digits are removed it will try as last resort the filename default.

On your TFTP server, this algorithm can be used to tell each single machine how to boot:

```
/tftpboot/pxelinux.cfg/
C0A8000B -> default.netboot-8.0
C0A8000C -> default.netboot-8.1
default.netboot-8.0
default.netboot-8.1
default
```

Output 29: *PXELINUX Configuration*

This is important if you install a lot of machines at the same time. You can watch the syslog file on your TFTP server and whenever a client got its initial RAM disk transmitted, you can remove the symlink for that machine from the `pxelinux.cfg` directory. This forces the client to load the default configuration which says: "Boot from local disk!" when it reboots after AutoYaST is done.

- **TFTP** PXE requires a special TFTP server. Read `pxelinux.doc` from the already mentioned syslinux package for details.

  The inetd based TFTP server cannot reliably handle much more than 64 clients at a time! With more clients not all of them will get an answer from your TFTP and you will see syslog messages like this:

  ```
  tftpd: read: Connection refused.
  ```

  To overcome this problem you can use `atftp` which is available as a package. This TFTP server can run as stand-alone daemon. The TFTP server directory `/tftpboot` should look like this:

  ```
  /tftpboot/initrd
  pxelinux.0
  linux
  ```

  Output 30: Contents of the Tftpboot Directory

  See next section for how to get the files `linux` and `initrd`.

- **Kernel and Initial Ramdisk** For network booting and other configurations, it is recommended to use the images available on every SuSE CD-ROM in the directory `/suse/images/boot`. The initial ramdisk (initrd) contains all kernel modules needed for successful installation. In special cases, you might need to build you own kernel or use special kernels available on the CD-ROM.

BOOTP and DHCP Options To allow the specification of the source medium location when booting over the network, another DHCP option can be used. This `root-path` option is shown in the next example.

```
subnet 192.168.1.0 netmask 255.255.255.0

      range dynamic-bootp 192.168.1.100 192.168.1.110;
      option broadcast-address 192.168.1.255;
      option routers 192.168.1.1;
      filename "vmlinuz.nbi";
      option root-path "/tftpboot/CDs";

      next-server 192.168.1.1;
```

Output 31: /etc/dhcpd.conf with the root-path Option

One more example shows how the DHCP server can send an image to
the client depending on the type of the requesting client (PXE or Ether-
boot).

```
ddns-update-style none;
allow bootp;
allow booting;


subnet 192.168.1.0 netmask 255.255.255.0
  range dynamic-bootp 192.168.1.100 192.168.1.110;
  option domain-name "cluster.suse.de";
  option routers 192.168.1.240;
  option subnet-mask 255.255.255.0;
  option broadcast-address 192.168.1.255;
  filename "vmlinuz-node.nbi";
  option root-path "/local/CD1";


group
    next-server 192.168.1.240;
    use-host-decl-names on;

    host n1
        hardware ethernet 00:00:1c:b5:6e:71;
        fixed-address n1;
        if substring (option vendor-class-
identifier, 0, 9) = "PXEClient"
            filename "/tulip.lzpxe";
```

```
        else if substring (option vendor-class-
identifier, 0, 9) = "Ether
boot"
            filename "/vmlinuz-node.nbi";


    host n2
        hardware ethernet 00:00:1c:b5:72:ea;
        fixed-address n2;
        if substring (option vendor-class-
identifier, 0, 9) = "PXEClient"
            filename "pxelinux.0";
         else if substring (option vendor-class-
identifier, 0, 9) = "Ether
boot"
            filename "/vmlinuz-node.nbi";
```

Output 32: *DHCP Server Configuration with PXE and Etherboot Options*

Invoking the Autoinstallation Process

Adding the command line variable `autoyast` will make linuxrc start in auto-
mated mode. Linuxrc searches for a configuration file, which should be distin-
guished from the main control file in the following places:

- In the root directory of the initial ramdisk that you use to boot the sys-
  tem

- In the root directory of the floppy

The control file for linuxrc can contain the keywords listed in Table A.1 on the
next page.

Keyword	Value
netdevice	Which network device to use for network setup (Device used for BOOTP or DHCP requests)
server	Which server to contact for source directory (NFS Server)
serverdir	Directory on NFS Server
ip	When empty, client sends BOOTP request. Otherwise, client is configured with entered IP configuration.
netmask	Netmask
gateway	Gateway
nameserver	Name server
insmod	Kernel modules to load
autoyast	Location of the the machine profile to use for the automatic installation, such as `autoyast=http://192.168.2.1/profiles/`
install	Location of the installation directory, such as `install=nfs://192.168.2.1/CDs/`
instmode	Installation mode (e.g., nfs, http, etc.) (Not needed if install is set)

Table A.1: *Keywords for linuxrc*

These variables and keywords bring the system up to the point where YaST2 can take over with the main control file. Currently, the source medium is automatically discovered, which, in some cases, makes it possible to initiate the autoinstall process without giving any instructions to linuxrc.

The traditional linuxrc configuration file (info) should be used only in the preparation phase and has the function of giving the client enough information about the installation server and the location of the sources. In most cases, this file is not needed. It is, however, needed in some special network environments where DHCP and BOOTP are not used or when special kernel modules need to be loaded.

An alternative to supplying the mentioned keywords in the control file is to pass them to linuxrc using the kernel command line. All key and variable combinations can now be passed this way. The command line can, for example, also be set when creating network bootable images or it can be passed to the kernel using a specially configured DHCP server in combination with Etherboot or PXE. The format of the special command line variable autoyast can be used as seen in Table A.2 on the facing page.

Command Line Variable	Description
`autoyast=default`	Default autoinstallation option
`autoyast=file://<path>`	Looks for control file in specified path (relative to source root directory, e.g., `file://autoinst.xml` if in the top directory of a CD-ROM)
`autoyast=floppy://<path>`	Looks for control file in the floppy (Useful when booting from CD)
`autoyast=nfs://<server>:<path>`	Looks for control file on `<server>`
`autoyast=http://<server>/<path>`	Retrieves the control file from a web server using the HTTP protocol.
`autoyast=tftp://<server>:<path>`	Retrieve the control file with TFTP

Table A.2: Command Line Variables for AutoYaST2

Several scenarios for autoinstallation are possible using different types of infrastructure and source media. The simplest way is by using the source media from the SuSE Box. To initiate the autoinstallation process, however, the autoinstallation command line variable should be entered at system boot and the control file should be accessible to YaST2. The following list of scenarios explains how the control file can be supplied and the setup needed for the autoinstallation process to be successful.

- Using SuSE original CDs from SuSE Linux Enterprise Server box: To use the original CDs, the user needs a medium with the control file. The control file can reside on the following locations:

    1. *Floppy*: Control file accessible via the `autoyast=floppy` option. YaST2 also searches upon start-up for a file named `autoinst.xml`. If such a file is found, YaST2 will switch into autoinstallation mode even if no special command line variables were supplied. To use this option, create the control file and copy it to a pre–formatted floppy disk and start the installation as usual.

    2. *Network*: Control file accessible via the `autoyast=nfs://..`, `autoyast=http://..`, or `autoyast=tftp://..` options.

- Using "self-made" CDs:

  In this case, the user can include the control file on the CD-ROM for easy access (using the `autoyast=file://` option) or use one of the above-mentioned methods used with the original SuSE CDs.

- Using NFS and Floppy, Network or CD-ROM for system boot. This option is the most important one due to the fact that installations of PC farms are normally done using NFS servers and other network services like BOOTP and DHCP . The control file can reside in the following places:

  1. *Floppy or CD-ROM*: Control file accessible via the `autoyast=file://..` option.
  2. *Network*: Control file accessible via the `autoyast=http://..`, `autoyast=nfs://..` or `autoyast=tftp://..` options.

Default Behavior

If `autoyast=default` is defined, YaST2 will look for a file named `autoinst.xml` in the following three places:

1. The root directory of the floppy disk.

2. The root directory of the installation medium.

3. The root directory of the initial ram disk used to boot the system.

This is the default that also matches the behavior of linuxrc in earlier releases of SuSE Linux Enterprise Server.

Control File on Boot Media

Define the location of the control file using the file option, which indicates where the control file is located. Depending on the boot method used, YaST2 will look for the control file using the specified path in the root directory of the initial ramdisk (initrd).

Control File on a Floppy

YaST2 looks for the control file on a floppy in the specified directory. This is particularly useful when using a CD-ROM to boot (Original SuSE CD-ROMs).

Control File Retrievable via HTTP

To use HTTP, set the boot command line variable `autoyast` with the control file location as described in Table A.2 on page 75. It is possible to specify the location of the control file using the following methods:

1. Specify the exact location of the control file:

   `autoyast=http://192.168.1.1/control-files/client01.xml`

2. Specify a directory where several control files are located

   `autoyast=http://192.168.1.1/control-files/`

   In this case, the relevant control file is retrieved using the hex digit representation of the IP as described below.

If only the pathprefix variable is defined, YaST2 will fetch the control file from the HTTP server in the following way:

1. First, it will search for the control file using its own IP address in uppercase hexadecimal, for example, `192.0.2.91` is `C000025B`.

2. If that file is not found, it will remove one hex digit and try again. This action is repeated until the file with the correct name is found. Ultimately, it will try looking for a file named `default` (in lowercase).

As an example, for `192.0.2.91`, the HTTP client will try `C000025B`, `C000025`, `C00002`, `C0000`, `C000`, `C00`, `C0`, `C`, and `default`, in that order.

To determine the hex representation of the IP address of the client, use the utility */usr/sbin/gethostip* available with the `syslinux` package.

```
myhost # /usr/sbin/gethostip 10.10.0.1
10.10.0.1 10.10.0.1 0A0A0001
```

Control File Retrievable via TFTP

This option is similar to the former one using HTTP.

Control File on an NFS Server

YaST2 will looks for the control file on the NFS server specified on the command line. In this case, linuxrc will automatically probe the network device and use DHCP to configure the ethernet device. The same behavior dealing with hex IPs is also available here (as with tftp and http).

Starting Autoinstallation

The Autoinstallation Process

After the system has booted and the control file has been retrieved, YaST2 performs configuration of the system according to the information contained in the control file. The configuration is summarized in a window that is shown by default. It should be deactivated if a full automatic installation is needed.

Before displaying the summary of the configuration, YaST2 has only probed hardware and prepared the system for autoinstallation. Nothing has been changed in the system yet, so the process still can be aborted in case of error.

A system should be automatically installable without the need to have any graphic adaptor or monitor. Having a monitor attached to the client machine is recommended for following the process and getting feedback in case of any errors. Choosing between the Qt and the Ncurses interfaces is possible. For headless clients, system messages can be monitored using the serial console.

X11 Interface

This is the default interface while autoinstalling. No special variables are required to activate it.

Serial Console

You can start installing a system using the serial console by adding the keyword console (for example, console=ttyS0) to the command line of the kernel. This will start linuxrc in console mode and, later in the process, YaST2 is also started in serial console mode.

Text-Based YaST2 Installation

This option can also be activated on the command line. This will start YaST2 in Ncurses mode. To start YaST2 in text mode, add textmode=1 on the command line.

Starting YaST2 in text mode is recommended when installing a client with less than 64 MB or when X11 is not being configured at all, especially on headless machines.

System Configuration

The system configuration stage of the autoinstallation can be seen as the most important part of the whole process. Customizing a system to your needs is what makes an autoinstallation system attractive, not the installation part.

As you have seen in the previous chapters, almost anything can be configured automatically on the target system. In addition to the predefined directives, you can use postscripts to change other things in the system. Additionally, you can change any system variable and, if required, copy complete configuration files into the target system.

Postinstall and System Configuration

The postinstallation and the system configuration are initiated directly after the last package is installed in the target system and is continued after the system has booted for the first time.

Before the system is booted for the first time, YaST2 writes all data collected during installation into the system and finally writes the boot loader in the specified location. In addition to these regular tasks, which are also done when performing a normal installation, YaST2 executed the chroot scripts as specified in the control file. These scripts are executed while the system still is not mounted.

If a different kernel than the default is installed, a hard reboot will be required. A hard reboot can also be forced during autoinstallation, independent of the installed kernel. This can be accomplished using the reboot property of the general resource. (See *General Options* in Section *A Closer Look at Profile Resources* on page 43.)

System Customization

Most of the system customization is done in the second stage of the installation. YaST2 provides most of the important resources needed to bring a system into a usable general state. However, you may have other requirements for the installed system. If the required customizations cannot be done using YaST2 resources, the postinstall scripts can be used to accomplish this task. Define an unlimited number of custom scripts in the control file either by editing the control file or by using the configuration system.

Migration from YaST1 and ALICE

ALICE, SuSE's former autoinstallation system, was a system built around the autoinstallation features available with YaST1. For using existing ALICE configuration files and resources, a special option is provided in the configuration management system that enables conversion ALICE system configuration files to control files readable by AutoYaST2.

ALICE uses YaST1 for the installation of the Linux system. This is done by creating a boot medium with the needed control file (info) from a set of configuration files maintained in a CVS repository. The info file has the traditional format used with YaST1.

System Configuration with ALICE

The system configuration is done almost entirely after the system is installed and is initiated as the %post section of the ALICE rpm package.

ALICE Modules

Each of the ALICE modules performs certain tasks and requires one or more configuration files. The modules are shell scripts run after installation of the machine to set up different services on the client.

YaST2 offers an extensive and rich interface to the installed system that replaces most of these modules available with ALICE. Table A.3 on the next page is a list of all ALICE modules, their functions, and how their functionality is provided by YaST2 modules already available or that can be integrated easily.

The modules prepare_alice and make_all provided the base and main scripts of all ALICE modules. The module prepare_alice is called right after the initial installation process — done when YaST1 has finished the installation packages.

With YaST2, the alternative to these modules is a part of the system and not an addition or extension. This means that YaST2 configures the system in one single process and not, as with ALICE, in two different, independent steps.

Module	ALICE	YaST2	Concerned Files
User configuration	Adding users is performed without any consistency check	Using YaST2 users module	/etc/passwd

Group config-uration	Adding users is performed without any consistency check	Using YaST2 users module	/etc/group
Services	X	–	/etc/services
Inetd	X	Using the inetd mod-ule	/etc/inetd.conf
Syslog	X		/etc/syslog.conf
Lilo	X	X	/etc/lilo.conf
SNMP	X	–	/etc/ucdsnmpd. conf
Cron	X	–	/etc/crontab
Hosts	X	X	/etc/hosts
Routing	X	X	/etc/route.conf
Printer	X	X	/etc/printcap
SSH	X	–	/etc/ssh
Kernel mod-ules	X	X	/etc/modules.conf

Table A.3: YaST2 Modules Replacing ALICE

Other Configuration Options with YaST2 and ALICE

ALICE completed most of the configuration using the configuration file /etc/rc.config. This interface to the system no longer exists with newer SuSE products. Instead, configuration options are now available in /etc/sysconfig.

Entries in /etc/sysconfig can be easily set and modified using YaST2. For network services, it is recommended to use the relevant YaST2 modules deal-ing with networking rather than modifying /etc/sysconfig/network di-rectly.

Red Hat Kickstart

The configuration management system offers an option for importing a for-eign autoinstallation configuration file using the Kickstart system. To import a Kickstart file, you only need to enter the path of the configuration file and

decide how to handle the data, by either saving the result into a file directly or by loading the resulting AutoYaST2 compatible control file into the configuration management system to add more configuration options available with SuSE Linux Enterprise Server. The following section describes some limitations and issues to consider when importing from a foreign source.

Software Selections and Packages

It is not possible at the moment to import package selections and single package listings into AutoYaST2. The packages will be added to the control file, but manual selection of the desired packages is needed. The reason is that package groups and package names differ depending on the source distribution.

User Scripts

Kickstart offers two types of scripts, *pre* and *post* scripts. More (>2) scripts are not supported as in AutoYaST2. So it might be needed to split large scripts into smaller components depending on the function to make such scripts manageable.

YaST2 in Text Mode (ncurses)

YaST2 can also be used from a text-based terminal. This is useful when the administrator cannot access the system via a graphical console running X11.

Invocation and Usage

To start YaST2 in text mode, enter yast as root in a terminal.

The usage may be unfamiliar, but is very simple. The whole program can be operated with (Tab), (Alt) + (Tab), (Space), the arrow keys ((↑) and (↓)), and (Enter) as well as with shortcuts. When YaST2 is started in text mode, the YaST2 Control Center appears first, as shown in Figure B.1.

Figure B.1: *Main Window of the YaST2 Control Center*

The window is divided into three areas: The box on the left hand side shows the categories to which the various modules belong. When activated, this category selection is highlighted by a broad white frame. The selected, active category is color-highlighted. The corresponding modules of the active category are listed in a white-framed box on the right hand side of the window. At the bottom, find the buttons for 'Help' and 'Quit'.

After the first start of the YaST2 Control center, the uppermost category 'Software' is automatically selected. Change between categories using (↑) and (↓). Start a module belonging to the selected category by pressing (→). The module selection then appears highlighted by a broad white line. Select a module using (↑) or (↓). Scroll through the module selection by pressing either key continuously. When a module is selected, the title is color-highlighted. A short text describing this module is displayed in the bottom part of the window.

Start the desired module by pressing (Enter) when it is selected. Different buttons or selection fields of the module contain a differently-colored letter (yellow with the standard settings). The combination (Alt) + (yellow letter) selects the corresponding button directly.

Leave the YaST2 Control Center either using the 'Quit' button at the bottom part of the window or by choosing the 'Quit' menu item in the category selection and pressing (Enter).

Restriction of Key Combinations

It is possible that the (Alt) combinations in YaST2 do not work if system-wide (Alt) key combinations are set by a running X server. It is also possible that keys like (Alt) or (⇑ Shift) are captured for the terminal used.

Replacing (Alt) with (Esc): (Alt) shortcuts can be executed with (Esc) instead of (Alt). For example, (Esc) + (H) replaces (Alt) + (H).

Replacement of backward and forward navigation by (Ctrl) + (F) and (Ctrl) + (B): If the (Alt) and (⇑ Shift) combinations are occupied by the window manager or the terminal, the combinations (Ctrl) + (F) (forward) and (Ctrl) + (F) (backward) can be used instead.

Module Operation

In the following, it is assumed that the (Alt) key combinations are functional. Make appropriate substitutions or switch to a pure text console, if needed.

Navigation between buttons and selection lists $\boxed{\text{Tab}}$ and $\boxed{\text{Alt}}$ + $\boxed{\text{Tab}}$ navigates back and forth between buttons and frames containing selection lists and among the frames.

Navigation in selection lists $\boxed{\uparrow}$ and $\boxed{\downarrow}$ always navigate among the single items within an activated frame containing a selection list. These can, for instance, be the single modules of a module group in the control center.

Checking radio buttons and check boxes The selection of buttons with empty square brackets (check boxes) or parentheses (radio buttons) can be done with the $\boxed{\text{Space}}$ or $\boxed{\text{Enter}}$ keys. The buttons at the bottom of the various modules or of the control center are activated with $\boxed{\text{Enter}}$ when selected (colored green) or with the combination $\boxed{\text{Alt}}$ + $\boxed{\text{yellow key}}$ (cf. fig. B.2).
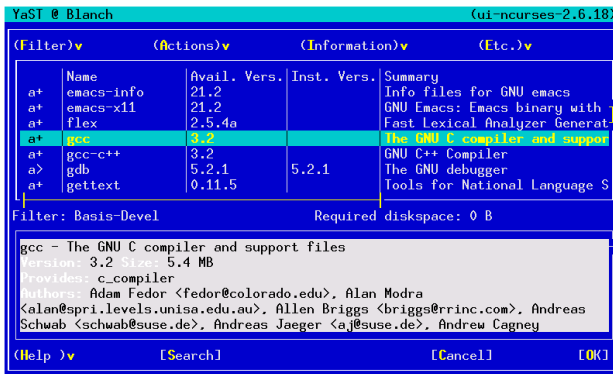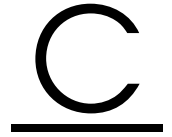

Figure B.2: *The Software Installation Module*

Invoking the Various Modules

Each YaST2 module can also be started directly. The modules can simply be started with yast ⟨*module name*⟩. The network module, for instance, is started with the command yast lan. Access a list of the names of the modules available on a system by running yast -l or yast --list.

LVM - The Logical Volume Manager

This text is based on the "LVM-HowTo" written by Heinz Mauelshagen.

Basics

LVM provides a very sophisticated way to handle disk space. You can build "Logical Volumes" by concatenating physical partitions. The partitions may be spread over different drives.

Example: You planned to use 600 MB for /home but actually you need 1 GB. LVM enables you to simply add another partition with 400 MB for example, to the existing /home "on the fly". Without LVM you would need to have a second partition with at least 1 GB and you would have to unmount the old /home, mount the new Partition to /home and then copy all the data.

The handling of LVM is most simple if you are going to set up a totally new logical volume. This can easily be done using YaST. After building a file system on the new volume, you can use it like any other partition.

If you need to add or remove space to a logical volume containing a running file system, you have to use separate tools like ext2resize after growing the logical volume.

Further options: LVM provides the possibility of "striping". Another interesting feature is the "snapshot", which provides a non persistent backup functionality.

Terms

Lets have a look at the terms used by LVM. Knowing these terms should help to understand the different YaST menus.

`Physical Volume (PV)`

A PV is a physical medium (e.g. `/dev/sda`), which is prepared to be used by LVM. Therefore, some administrative data is added to it. To use a partition with LVM, the partition-type needs to be of the type "8E".

`Physical Extent (PE)`

PEs are like big blocks. A PV is subdivided into PEs. The default PE-size is 4 MB.

`Volume Group (VG)`

A Volume Group contains a number of PEs, provided by one or more PVs. You don´t have to worry about the PEs at this point, since you only have to tell LVM which PVs to use.

`Logical Volume (LV)`

Logical volumes are roughly the same as "partitions" and the linux kernel makes no difference between a regular partition and a logical volume. You can build any kind of supported file system on the LV. The number of LVs is currently limited to 256. This amount is shared among all VGs. The size of a LV is limited by the PE-size you configured at the time of volume group creation. When using the default value (4 MB), each LV is limited to 256 GB in size. If you need a bigger LV, you need to choose another PE-size. For example, LVs containing 1 Terabyte can be achieved by using a PE-size of 16 MB.

How do I Access the LVs?

Assuming you have the two PVs `/dev/sda` and `/dev/sdb` and a VG named "suse" containing the two PVs, the resulting LVs will be found at `/dev/suse/`.
After creating a number of LVs named e.g. `test1`, `test2` ...`testn` you will find a hierarchy like

```
/dev/suse/test1
/dev/suse/test2
...
/dev/suse/testn
```

You can mount these volumes like regular partitions.

Further documentation: Assumed, you have a running SuSE Linux Enterprise Server, you can find some documentation, including the LVM-FAQ, at `/usr/ share/doc/packages/lvm`.

The LVM-HOWTO can be found at `/usr/share/doc/howto/en/ LVM-HOWTO.gz`.

The HTML-Version can be accessed via `/usr/share/doc/howto/en/ html/LVM-HOWTO.html`.

Configuration with YaST2

You can start the LVM configuration by selecting 'LVM...' in the 'Expert Partitioner' dialog.

The partitioner

First, you will reach a dialog where you can change the partitioning of your hard disk. Here, remove or change current partitions as well as create new ones. A partition to use for LVM must have the partition label 8E. These partitions are marked with the text "Linux LVM" in the partition list inside the window.

It is not necessary to individually set each partition designated for LVM to the partition label 8E. When needed, YaST2 will automatically set the label of a partition assigned to an LVM volume group to 8E. If there are unpartitioned areas on your disks, add LVM paritition in this dialog for all these areas. These partitions should immediately be set to the partition label 8E. They do not need to be formatted and cannot be listed as a mount point.
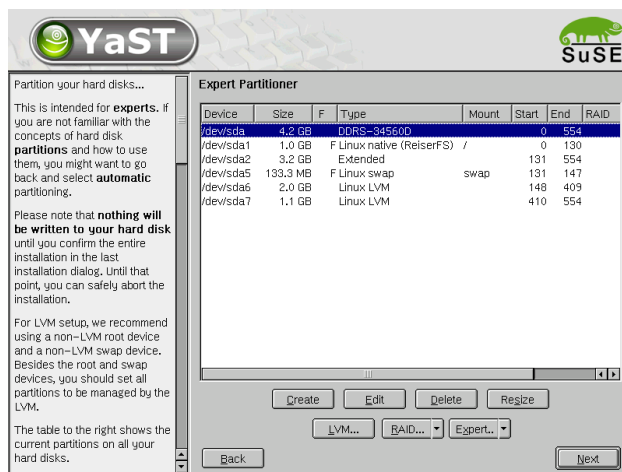
Figure C.1: YaST2: LVM Partitioner

> **Note**
>
> If a valid LVM configuration already exists on your system, it will au-
> tomatically be applied at the start of the LVM configuration. If this
> configuration is activated, no disks containing a partition belonging
> to an activated volume group can be repartitioned. The Linux kernel
> will refuse to detect the modified partitioning of a hard disk as long as
> even a single partition on this drive is being used. Of course, reparti-
> tioning the disks not attributed to an LVM volume group is not a prob-
> lem. If you already have a valid LVM configuration on your system,
> it is usually not necessary to repartition it. In this dialog, configure all
> the mount points not located on the LVM logical volumes.
>
> In YaST2, at least the root file system must be located on a normal par-
> tition. Select this partition from the list and define it as the root file
> system using the 'Edit' button. Due to the great degree of flexibility in
> LVM, we recommend assigning all other file systems to LVM logical
> volumes. After setting the root partition, exit this dialog.
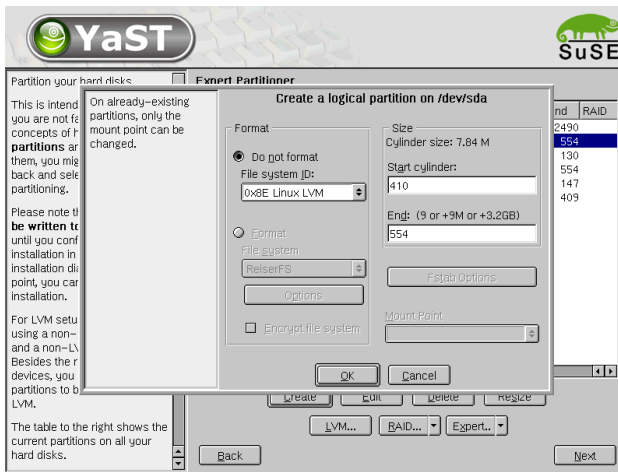>
> **Note**

Figure C.2: YaST2: Creating an LVM partition

Setting up physical volumes

This dialog manages the LVM volume groups (often abbreviated to "VG"). If there is no volume group yet on your system, you will be prompted by a pop-up window to create one. "System" is the name suggested for the volume group where your SuSE Linux Enterprise Server system files are located. The physical extent size (often abbreviated to PE size) defines the maximum size of a physical and logical volume in this volume group. This value is usually set to 4 megabytes. This allows for a maximum size of 256 gigabytes for a physical and logical volume. You should, therefore, only increase the physical extent size (for example, to 8, 16, or 32 megabytes) if you need logical volumes larger than 256 gigabytes.

In the following dialog, all partitions are listed that either have "Linux LVM" or the "Linux native" types. All swap and DOS partitions will not be shown. If a partition is already assigned to a volume group, the name of the volume group will be listed. Unassigned partitions bear the label "–".

The volume group currently edited can be modified in the selection box to the upper left. The buttons to the upper right enable creation of additional volume groups and deletion of existing volume groups. However, only volume groups without any additional partitions assigned to them can be removed. For a standard SuSE Linux Enterprise Server system, you do not need to create more than one volume group. A partition assigned to a vol-
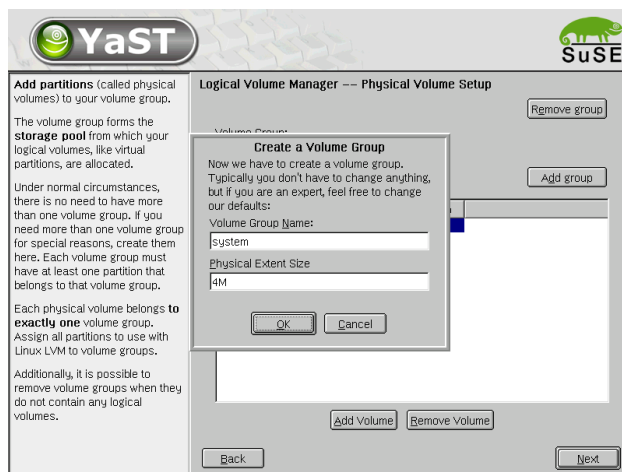
Figure C.3: *YaST2: Creating a Volume Group*

ume group is also called a physical volume (often abbreviated to PV). To add a previously unassigned partition to the volume group you selected, first select the partition and then click on the button 'Add volume' below the selection list. This allows the name of the volume group to be entered next to the partition selected. Assign all partitions intended for LVM to a volume group. Otherwise, the space on the partition will remain unused. Before exiting the dialog, assign at least one physical volume to each volume group.

Logical Volumes

This dialog manages the logical volumes (often just abbreviated to "LV").

Logical volumes are each assigned to a volume group and each have a certain size. Normally, a file system is generated on a logical volume (e.g. reiserfs, ext2), which is also designated a mount point. On an installed system, the files stored on this logical volume can then be found at this mount point.

All the standard Linux partitions assigned a mount point, all swap partitions, and all already existing logical volumes are found in this list. If you have already configured LVM on your system, the available logical volumes should already be listed here. You will, however, still have to assign the appropriate mount point to these logical volumes. If you are configuring LVM on your system for the first time, there will not be any logical volumes yet in this screen and you will have to generate a logical volume for each mount
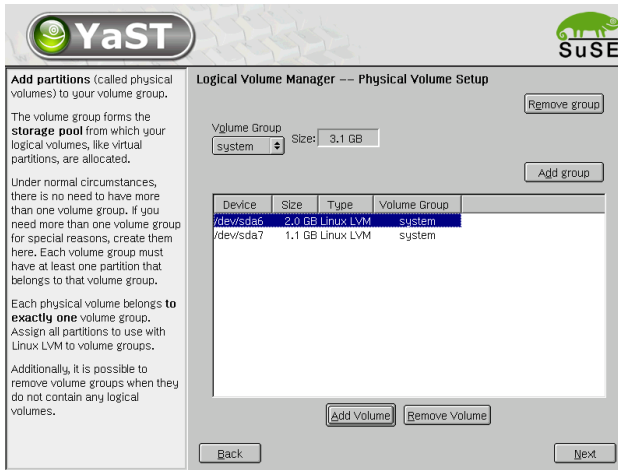
Figure C.4: YaST2: Overview of Partitions

point (with the 'Add' button), as well as define the size, the file system type (e. g., reiserfs or ext2), and the mount point (e. g., /var, /usr, /home).

If you have created several volume groups, switch between the different volume groups in the selection list to the upper left. The new logical volumes are all located in the volume group shown at the upper left. After creating all the logical volumes as required, the LVM configuration is complete. You can then exit the dialog and continue on to software selection if you are currently in the process of installation.

⌐ **Caution** ────────────────────────────────

Implementing the LVM is also associated with increased risk factors such as data loss. Possible dangers are application crashes, power outages, and faulty commands. Secure your data before using LVM and before reconfiguring volumes. Do not work without backups.
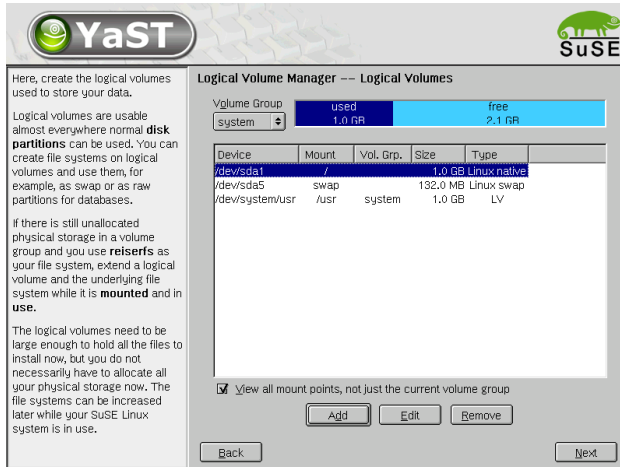
─────────────────────────────── **Caution** ⌐

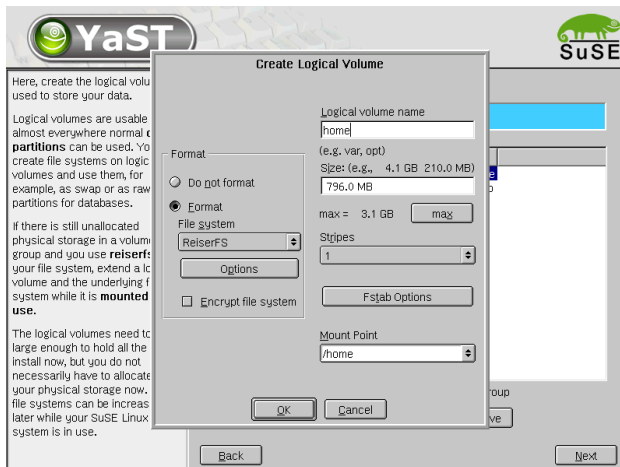*Figure C.5:* YaST2: Management of the Logical Volumes

Figure C.6: YaST2: Creating Logical Volumes

File Systems in Linux

There are a number of file systems supported by Linux. This chapter presents a brief overview of the most popular Linux file systems, elaborating on their design concept, advantages, and fields of application. Some additional information on LFS "Large File Support" in Linux is also provided.

Glossary

metadata   A file system internal data structure that assures all the data on disk is properly organized and accessible. Essentially, it is "data about the data." Almost every file system has its own structure of metadata, which is partly why the file systems show different performance characteristics. It is of major importance to maintain metadata intact, because otherwise the whole file system could be corrupted.

inode   Inodes contain all sorts of information about a file, including size, number of links, date, and time of creation, modification, and access, as well as pointers to the disk blocks where the file is actually stored.

journal   In the context of a file system, a journal means an on-disk structure containing a kind of log where the file system driver enters what it is about to change in the file system's (meta)data. "Journaling" greatly reduces the recovery time of a Linux system as the file system driver does not need to initiate an exhaustive search for corrupt metadata all over the disk. Instead, the journal entries are replayed.

Major File Systems in Linux

Unlike two or three years ago, choosing a file system for your Linux system is no longer a matter of a few seconds ("Ext2 or ReiserFS?"). Kernels starting from 2.4 offer a variety of file systems from which to choose. The following is an overview of how those file systems basically work and which advantages they offer.

It is very important to bear in mind that there may be no file system that best suits all kinds of applications. Each file system has its particular strengths and weaknesses, which have to be taken into account. Even the most sophisticated file system in the world will never substitute for a reasonable backup strategy.

The terms "data integrity" or "data consistency", when used in this chapter, do not refer to the consistency of the user space data (the data your application writes to its files). Whether this data is consistent or not must be controlled by the application itself.

Ext2

The origins of Ext2 go back to the early days of Linux history. Its predecessor, the Extended File System, was implemented in April 1992 and integrated in Linux 0.96c. Extended File System underwent a number of modifications and, as Ext2, became the most popular Linux file system for years. With the creation of journaling file systems and their astonishingly short recovery times, Ext2 became less important.

A brief summary of Ext2's strenghts might help you to understand why it was – and in some areas still is – the favorite Linux file system of many a Linux user.

Solidity   Being quite an "old-timer", Ext2 underwent many improvements and was heavily tested. This may be the reason why people often refer to it as "rock–solid". After a system outage when the file system could not be cleanly unmounted, e2fsck starts to analyze the file system data. Metadata is brought into a consistent state and pending files or data blocks are written to a designated directory (called lost+found). In contrast to (most of) the journaling file systems, e2fsck analyzes the whole file system and not just the recently modified bits of metadata. This takes significantly longer than the checking of the log data of a journaling file system. Depending on file system size, this procedure can take half an hour or more. Therefore, you would not choose Ext2

for any server that needs to be highly available. Yet, as Ext2 does not maintain a journal and uses significantly less memory, it is sometimes faster than other file systems.

Easy upgradability   The code Ext2 is the strong foundation on which Ext3 could become a highly-acclaimed next-generation file system. Its reliability and solidity were elegantly combined with the advantages of a journaling file system.

Ext3

Ext3 was designed by Stephen Tweedie. In contrast to all other "next-generation" file systems, Ext3 does not follow a completely new design principle. It is based on Ext2. These two file systems are very closely related to each other. An Ext3 file system can be easily built on top of an Ext2 file system. The most important difference between Ext2 and Ext3 is that Ext3 supports journaling.

Summed up, Ext3 has three major advantages to offer:

Easy and highly reliable file system upgrades from Ext2   As Ext3 is based on the Ext2 code and shares its on-disk format as well as its metadata format, upgrades from Ext2 to Ext3 are incredibly easy. They can even be transformed while your Ext2 file systems are mounted. Unlike transitions to other journaling file systems, such as ReiserFS, JFS, or XFS, which can be quite tedious (making backups of the whole file system and recreating it from scratch), a transition to Ext3 is a matter of minutes. It is also very safe, as the recreation of a whole file system from scratch might not work flawlessly. Considering the number of existing Ext2 systems that await an upgrade to a journaling file system, you can easily figure out why Ext3 might be of some importance to many system administrators. Downgrading from Ext3 to Ext2 is as easy as the upgrade. Just perform a clean unmount of the Ext3 file system and re-mount it as an Ext2 file system.

Reliability and performance   Other journaling file systems follow the "metadata-only" journaling approach. This means your metadata will always be kept in a consistent state but the same cannot be automatically guaranteed for the file system data itself. Ext3 is designed to take care of both metadata and data. The degree of "care" can be customized. Enabling Ext3 in the `data=journal` mode offers maximum security (i.e., data integrity), but can slow down the system as both

metadata and data are journaled. A relatively new approach is to use the `data=ordered` mode, which ensures both data and metadata integrity, but uses journaling only for metadata. The file system driver collects all data blocks that correspond to one metadata update. These blocks are grouped as a "transaction" and will be written to disk before the metadata is updated. As a result, consistency is achieved for metadata and data without sacrificing performance. A third option to use is `data=writeback` which allows data to be written into the main file system after its metadata has been committed to the journal. This option is often considered the best in performance. It can, however, allow old data to reappear in files after crash and recovery while internal file system integrity is maintained. Unless you specify something else, Ext3 is run with the `data=ordered` default.

ReiserFS

Officially one of the key features of the 2.4 kernel release, ReiserFS has been available as a kernel patch for 2.2.x SuSE kernels since SuSE Linux Enterprise Server version 6.4. ReiserFS was designed by Hans Reiser and the Namesys development team. ReiserFS has proven to be a powerful alternative to the old Ext2. Its key assets are better disk space utilization, better disk access performance, and faster crash recovery. However, there is a minor drawback: ReiserFS pays great care to metadata but not to the data itself. Future generations of ReiserFS will include data journaling (both metadata and actual data are written to the journal) as well as ordered writes.

ReiserFS's strengths, in more detail, are:

Better disk space utilization In ReiserFS, all data is organized in a structure called B*-balanced tree. The tree structure contributes in better disk space utilization as small files can be stored directly in the B*tree leaf nodes instead of being stored elsewhere and just maintaining a pointer to the actual disk location. In addition to that, storage is not allocated in chunks of 1 or 4 kB, but in portions of the exact size that is needed. Another benefit lies in the dynamic allocation of inodes. This keeps the file system more flexible in contrast to traditional file systems, like Ext2, where the inode density has to be specified at file system creation time.

Better disk access performance For small files, you will often find that both file data and "stat_data" (inode) information are stored next to each other. They can be read with a single disk IO operation, meaning that only one access to disk is required to retrieve all the information needed.

Fast crash recovery Using a journal to keep track of recent metadata
changes makes a file system check a matter of seconds, even for huge
file systems.

JFS

JFS, the "Journaling File System" was developed by IBM for their AIX systems. The first beta version of the JFS Linux port reached the Linux community in the summer of 2000. Version 1.0.0 was released in 2001. JFS is tailored
to suit the needs of high throughput server environments where performance
is the ultimate goal. Being a full 64-bit file system, JFS supports both large
files and partitions which is another pro for its use in server environments.

A closer look at JFS shows why this file system might prove a good choice
for your Linux server:

Efficient journaling JFS follows a "metadata only" approach like ReiserFS.
Instead of an extensive check, only metadata changes generated by recent file system activity get checked, which saves a great amount of
time in recovery. Concurrent operations requiring multiple concurrent
log entries can be combined into one group commit, greatly reducing
performance loss of the file system through multiple write operations.

Efficient directory organization JFS holds two different directory organizations. For small directories, it allows the directory's content to be stored
directly into its inode. For larger directories, it uses B^+trees, which
greatly facilitate directory management.

Better space usage through dynamic inode allocation For Ext2, you have
to define the inode density in advance (the space occupied by management information), which restricted the maximum number of files or
directories of your file system. JFS spares you these considerations —
it dynamically allocates inode space and frees it when it is no longer
needed.

XFS

Originally intended as file system for their IRIX OS, SGI started XFS development back in the early 1990s. The idea behind XFS was to create a high-performance 64-bit journaling file system to meet the extreme computing

challenges of today. XFS is very good at manipulating large files and performs well on high-end hardware. However, you will find a drawback even in XFS. Like ReiserFS, XFS takes a great deal of care of metadata integrity, but less of data integrity.

A quick review of XFS's key features explains why it may prove a strong competitor for other journaling file systems in high-end computing.

High scalability through the use of allocation groups At creation time of an XFS file system, the block device underlying the file system is divided into eight or more linear regions of equal size. Those are referred to as "allocation groups". Each allocation group manages its own inodes and free disk space. Practically, allocation groups can be seen as "file systems in a file system." As allocation groups are rather independent of each other, more than one of them can be addressed by the kernel simultaneously. This feature is the key to XFS's great scalability. Naturally, the concept of independent allocation groups suits the needs of multiprocessor systems.

High performance through efficient management of disk space Free space and inodes are handled by B^+ trees inside the allocation groups. The use of B^+ trees greatly contributes to XFS's performance and scalability. A feature truly unique to XFS is "delayed allocation". XFS handles allocation by breaking the process into two pieces. A pending transaction is stored in RAM and the appropriate amount of space is reserved. XFS still does not decide where exactly (speaking of file system blocks) the data should be stored. This decision is delayed until the last possible moment. Some short-lived temporary data may never make its way to disk since they may be obsolete at the time XFS decides where to actually save them. Thus XFS increases write performance and reduces file system fragmentation. Since delayed allocation results in less frequent write events than in other file systems, it is likely that data loss after a crash during a write is more severe.

Preallocation to avoid file system fragmentation Before writing the data to the file system, XFS "reserves" (preallocates) the free space needed for a file. Thus, file system fragmentation is greatly reduced. Performance is increased as the contents of a file will not be distributed all over the file system.

Some Other Supported File Systems

Table D.1 summarizes some other file systems supported by Linux. They are supported mainly to ensure compatibility and interchange of data with different kinds of media or foreign operating systems.

cramfs	*Compressed ROM file system*: A compressed read-only file system for ROMs.
hpfs	*High Performance File System*: the IBM OS/2 standard file system — only supported in read-only mode.
iso9660	Standard file system on CD-ROMs.
minix	This file system originates from academic projects on operating systems and was the first file system used for Linux. Nowadays, it is used as a file system for floppy disks.
msdos	*fat*, the file system originally used DOS, used today by various operating systems.
ncpfs	file system for mounting Novell volumes over networks.
nfs	*Network File System*: Here, data can be stored on any machine in a network and access may be granted via a network.
smbfs	*Server Message Block*: used by products, such as Windows, to enable files to be accessed over a network.
sysv	Used on SCO UNIX, Xenix, and Coherent (commercial UNIX systems for PCs).
ufs	Used by BSD, SunOS, and NeXTstep. Only supported in *read-only* mode.
umsdos	*UNIX on MSDOS*: applied on top of a normal fat file system. Achieves UNIX functionality (permissions, links, long file names) by creating special files.
vfat	*Virtual FAT*: extension of the fat file system (supports long file names).
ntfs	*Windows NT file system*, read-only.

Table D.1: *File System Types in Linux*

Access Control Lists

Up to now one of the definite advantages of commercial UNIX systems was support for Access Control Lists (ACLs).

The traditional UNIX file object permission model allows read, write and execute permissions to be assigned to the owner, the owning group, and to the "rest of the world" (i.e., those users which are neither the owner nor members of the owning group). This scheme is sufficient in the majority of cases that occur practically; several workarounds to stretch its limits have been used by system administrators for years. Advanced applications nevertheless occasionally require a more flexible scheme.

In such situations, when the traditional file object permission model is no longer sufficient, Access Control Lists come to the rescue. They allow permissions to be defined for individual users and groups other than the owner or the owning group.

Access Control Lists are a feature of the Linux kernel, currently supported on ReiserFS, Ext2, Ext3, and XFS. With Access Control Lists more sophisticated scenarios can be realized directly, with no need to implement complex permission models in applications.

A popular example of where Access Control Lists marvel is replacing a Windows server by a Linux server. Often some of the workstations continue to run Windows. The Linux machine offers file and print services to the Windows clients via Samba. Because Samba supports Access Control Lists, permissions of users can be set on the Linux server, but also in the Windows GUI (Windows NT or later only). Using the winbindd deamon it is even possible to define permissions for users who only exist in a Windows domain, but who have no accounts on the Linux server.

On the server, Access Control Lists can be manipulated using the getfacl and setfacl utilities.

For further information on these commands refer to the man page for acl (man 5 acl), the man page for getfacl (man 1 getfacl) and the man page for setfacl (man 1 setfacl). Another useful source of information is the POSIX 1003.1e draft 17 specification, which can be obtained from http://wt.xpilot.org/publications/posix.le/ or http://www.suse.de/~agruen/acl/posix/posix.html.

Extended Attributes

File objects have always been associated with a set of attributes like the file owner and owning group, permissions, and time stamps. The attributes that are available are predefined by the file system on which the file is stored. Maybe you sometimes would have liked to attach some additional piece of

information to a file, like the file's character encoding or a short description. With extended attributes this is finally possible.

Extended attributes (EAs) are name/value pairs associated permanently with files and directories, similar to the environment strings associated with a process. They were first introduced to Linux when support for the XFS file system was added. Meanwhile they are not only supported on XFS, but also on ReiserFS, Ext2, and Ext3. These file systems also use extended attributes internally to implement Access Control Lists.

Extended attributes are not seen by any of the standard file operations. They are manipulated using special system calls. The getfattr and setfattr utilities can be used to retrieve and set extended attributes.

Further information on these commands is contained in the man page for attr (man 5 attr), the man page for getfattr (man 1 getfattr) and the man page for setfattr (man 1 setfattr). Table D.2 presents an overview of the file system limits for extended attributes.

File System	per EA	all EAs
ReiserFS	64kB	unlimited
Ext2/Ext3	File system block size (1/2/4/8kB)	File system block size (1/2/4/8kB)
XFS	64kB	unlimited

Table D.2: File System Limitations for Extended Attributes

Large File Support in Linux

Originally, Linux supported a maximum file size of 2 GB. This was enough as long as no one tried to manipulate huge databases on Linux. Becoming more and more important for server computing, the kernel and GNU C library were modified to support file sizes larger than 2 GB when using a new set of interfaces that applications must utilize. Nowadays, (almost) all major file systems offer LFS support allowing you to perform high-end computing.

Table D.3 on the following page offers an overview of the current limitations of Linux files and file systems for Kernel 2.4. Hopefully the figures below will become outdated, with future kernel releases offering support for even larger files and file systems.

File System	File Size Limit	File System Size Limit
Ext2 or Ext3 (1 kB block size)	16448 MB (~16 GB)	2048 GB (2 TB)
Ext2 or Ext3 (2 kB block size)	256 GB	8192 GB (8 TB)
Ext2 or Ext3 (4 kB block size)	2048 GB (2 TB)	16384 GB (16 TB)
Ext2 or Ext3 (8 kB block size) (Systems with 8 kB pages only (like Alpha))	65568 GB (~64 TB)	32768 GB (32 TB)
ReiserFS 3.5	4 GB	16384 GB (16 TB)
ReiserFS 3.6 (as in Linux 2.4)	2^{60} Bytes (1 EB)	16384 GB (16 TB)
XFS	2^{63} Bytes (8 EB)	2048 GB (2 TB) (Linux kernel limitation)
JFS (512 Bytes block size)	4194304 GB (4 PB)	512 TB
JFS (4 kB block size)	33554432 GB (32 PB)	4 PB

Table D.3: Maximum Sizes of File Systems

For More Information

Each of the file system projects described above maintains its own home page where you can find mailing list information as well as further documentation and FAQs.

http://e2fsprogs.sourceforge.net/ext2.html
http://www.zipworld.com.au/~akpm/linux/ext3/
http://www.namesys.com/
http://oss.software.ibm.com/developerworks/opensource/jfs/
http://oss.sgi.com/projects/xfs/

A comprehensive multipart tutorial on Linux file systems can be found at *IBM developerWorks*:

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

For a comparison of the different journaling file systems in Linux, look at Juan I. Santos Florido's article at *Linuxgazette*: `http://www.linuxgazette.com/issue55/florido.html`.

Those interested in an in-depth analysis of LFS in Linux should try follow LFS site: `http://www.suse.de/~aj/linux_lfs.html`.

Support Services

SuSE Support Services offer a complete range of support services in connection with Linux. No matter whether you have questions on the installation of SuSE Linux Enterprise Server products or need to tackle an obstacle in your mission-critical IT solution, our service models adapt to your specific requirements.

For more information in the SuSE support services please contact one of the following SuSE offices or their website:

SuSE Germany
SuSE Linux AG
Business Support
Deutschherrnstr. 15-19
D-90429 Nürnberg

Phone:	+49-911-74053-2330
Fax:	+49-911-74053-489

bsupport@suse.de
`http://www.suse.de/en/services/support`

SuSE UK
SuSE Linux Ltd.
The Kinetic Centre, Theobald Street
Borehamwood, Herts. WD6 4PJ

Phone:	+44-20-8387-4086

solutions@suse.co.uk
`http://www.suse.co.uk`

SuSE USA
SuSE Inc.
318 Harrison, #301

Oakland, CA 94607
Phone: (510) 628 3386
bsupport@suse.com
http://www.suse.com

Maintenance

Benefit from online access to the *SuSE Linux Maintenance Web* and ensure that your system is always up-to-date and in a secure and stable state. Maintenance for one year is already included in SuSE Linux Enterprise Server or your SuSE Linux Business Solution and can be extended to further years.

Register your product at the following URL to get access to the Maintenance Web:

http://support.suse.de/en/register/

After registration you will receive an e-mail describing further proceedings. Once you have a login, you can directly access the *SuSE Linux Maintenance Web* at:

http://support.suse.de/psdb/

Index

Article Number: HB2103-3INST-INT

4 038564 007377