AutoYast

Anas Nashif

AutoYast
by Anas Nashif

Revision History

Revision $Id: autoyast1.sgml,v 1.4 2001/09/17 16:57:27 nashif Exp nashif $ 2001-09-14
Version 0.99

Table of Contents

List of Tables

List of Examples

Chapter 1. Introduction

This document describes automated installation of SuSE Linux using YaST. This requires a specially prepared boot-disk or boot image and one of several installation media. The boot-disk can be simply generated from the normal SuSE boot-disk. Currently CD-ROM, Hard-drive and NFS are possible media for the automated installation. The automated installation media can be created from the normal CDs as distributed by SuSE.

Chapter 2. Automated installation

2.1. Automated Boot disk

The automated boot-disk can be generated from the standard boot-disk. You simply copy the image of the standard boot-disk from the first CD onto a disk.

```
dd if=/media/cdrom/disks/bootdisk of=/dev/fd0
```

The boot disk contains the following files:

```
-rwxr-xr-x   1 root     root     631335 Mar 31 17:29 initrd
-r-xr-xr-x   1 root     root       8920 Mar 13 07:40 ldlinux.sys
-rwxr-xr-x   1 root     root     816685 Mar 13 07:40 linux
-rwxr-xr-x   1 root     root       2424 Mar 13 07:40 message
-rwxr-xr-x   1 root     root        774 Mar 13 07:40 syslinux.cfg
```

The file 'linux' is a kernel image, which supports the hard-disks onto which you are installing. If a SCSI system is installed, the corresponding SCSI controller has to be compiled into the kernel. If the installation is to be carried out via NFS, support for the networking cards has also to be compiled into the kernel. If you want to be able to install a wider range of hardware from the same boot-disk, support for all SCSI controllers and networking cards which could potentially be used have to be compiled into the kernel, or you can use modules as described below.

In order to load kernel modules during installation, you have to add the modules to be loaded in the `info` file on the boot-disk.

The file `syslinux.cfg` contains this in the second line:

```
append initrd=initrd rw ramdisk_size=65536
```

To make this disk capable of doing automatic installation you need to add *linuxrc=auto* to the second line. It should then read as follows:

```
append initrd=initrd rw ramdisk_size=65536 linuxrc=auto
```

The instructions for the automated installation are kept in the file `/suse/setup/descr/info`, which needs to be created on the diskette. The contents of this file are described in "*Control File*" of this document. This file may also be located in the root directory of the boot floppy or in the root directory of the initial ramdisk (initrd).

2.2. Automated network installation

Instead of using a floppy disk to initiate the auto-installation process, a bootp file can be created to start the installation over the network. This is useful when installing a large set of machines which might have no floppy disks or CD-ROM drivers.

The only difference between the automated floppy installation and the network installation is that the kernel and the init ramdisk are available on the installation server instead of the floppy.

To prepare for a network installation, you need to copy the linux kernel image and the initial ramdisk from the standard boot floppy to the transfer directory used by tftpd (Usually `/tftpboot`). Linuxrc looks for a file `info` in three places:

1. in the current root directory (that is, within the initrd),
2. in the root directory of the floppy (if any) and

3. in `/suse/setup/descr/` on the floppy.

To create the files needed for the network installation you can use the floppy disk available with every SuSE distribution or use the image found on first CD in the directory called `disks` (file name is `bootdisk`).

Please follow the following steps to extract the initrd and the kernel images from the boot disk image:

1. Mount the bootdisk image on the loop device and copy files to the `/tftpboot` directory:

```
mount -oloop <path to image>/bootdisk /mnt
mkdir -p /tftpboot
cd /tftpboot
cp /mnt/initrd .
cp /mnt/linux .
cp /mnt/syslinux.cfg .
umount /mnt
```

2. Copy the control file (info) to the initrd image:

```
cd /tftpboot
zcat initrd > initrd.tmp
mount -oloop initrd.tmp /mnt
cp <path to info file>/info /mnt
```

3. Copy kernel modules you might need for network and other devices to the modules directory in the initrd image:

   You can either copy these modules from the system you are running directly (kernel version and extra-version must be the same) or you can copy them from the modules disk image also available on the first CD-ROM by mounting it on i.e. `/mnt2`. All modules go to `/mnt/modules`.

4. After copying all needed files, unmount the `/mnt` directory and compress the initrd image:

```
umount /mnt
gzip -9 initrd.tmp
mv initrd.tmp.gz initrd
```

2.3. Creating images for various network boot methods

2.3.1. netboot

Netboot can be used with clients equipped with a network interface having an EPROM or a floppy disk containing the image of such an EPROM. Such a floppy can be created using the Etherboot package, which is available in SuSE distribution.

Install the netboot package using either YaST or the CD-ROM directly (It's available in the Beowulf series).

To create a boot image which can initiate an auto-installation process, please issue the following command in the `/tftpboot` directory:

```
mknbi-linux -a "linuxrc=auto  rw ramdisk_size=65536" -r initrd linux netboot
```

Please consult the mknbi-linux documentation for more options and information about this command.

2.3.2. PXE

For PXE support, please install PXELINUX from the SYSLINUX distribution and consult the documentation available with the package. Basically, you should keep the files initrd and linux in the tftpboot directory and change the command line variables in the file (The following is part of the pxelinux.doc file available in the SYSLINUX distribution).

PXELINUX operates in many ways like SYSLINUX. If you are not familiar with SYSLINUX, read `syslinux.doc` first, since this documentation only explains the differences.

On the TFTP server, create the directory `/tftpboot`, and copy the following files to it:

- pxelinux.0 - from the SYSLINUX distribution
- any kernel or initrd images you want to boot

Finally, create the directory `/tftpboot/pxelinux.cfg` configuration file (equivalent of `syslinux.cfg` -- see `syslinux.doc` for the options here) will live in this directory. Because more than one system may be booted from the same server, the configuration file name depends on the IP address of the booting machine. PXELINUX will search for its config file on the boot server in the following way:

First, it will search for the config file using its own IP address in upper case hexadecimal, e.g. 192.0.2.91 -> C000025B (you can use the included program "gethostip" to compute the hexadecimal IP address for any host.)

If that file is not found, it will remove one hex digit and try again. Ultimately, it will try looking for a file named `default` (in lower case).

As an example, for 192.0.2.91, it will try C000025B, C000025, C00002, C0000, C000, C00, C0, C, and default, in that order. (See also the section on special DHCP options.)

It should be noted that all filename references are relative to the directory where `pxelinux.0` lives in (usually `/tftpboot`). PXELINUX generally requires that filenames (including any relative path) are 63 characters or shorter in length.

PXELINUX does not support MTFTP, and I have no immediate plans of doing so. It is of course possible to use MTFTP for the initial boot, if you have such a setup. MTFTP server setup is beyond the scope of this document.

2.4. Configure a DHCP Server for network installation

The following is an example of how the DHCP server should be configured (SuSE 7.3):

2.4.1. netboot

This is a dhcp 3.0 configuration file, remove the first line if you have an older version.

```
# dhcpd.conf
#
ddns-update-style ad-hoc;
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 "netboot";
}
```

Don’t forget to enable the tftp service either in the inetd.conf or by using the module available with YaST2.

2.4.2. PXE

For PXE, there exist many configuration options and possibilities, depending on your NIC and software environment. For information about the various methods of configuring PXE/DHCP and tftp, please consult the pxelinux.doc file available with the syslinux package and the Intel PXE Specification.

For this example, pxelinux from the syslinux package is used:

```
# dhcpd.conf
#
   ddns-update-style ad-hoc;
   allow booting;
   allow bootp;
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 "pxelinux.0";
   option vendor-class-identifier "PXEClient";
   option vendor-encapsulated-options 09:0f:80:00:0c:4e:65:74:77:6f:72:6b:20:62:6f:6f:74
  next-server 192.168.1.1;
}
```

Chapter 3. Media

3.1. Installation via NFS

To create an automated installation environment on an NFS server, you have two choices for setting up the NFS directory. You can either place all of the source files in multiple directories which are named after the CDs from which they originated or place all of the source files into a single directory.

Here is a description of how to create one single directory:

1. Choose a base directory (e.g. `/usr/local/AUTOINSTALL`)

   ```
   mkdir /usr/local/AUTOINSTALL
   cd /usr/local/AUTOINSTALL
   ```

2. Copy the contents of the SuSE-CDs into this directory

   ```
   mount /cdrom
   cp -av /cdrom/* .
   cp /cdrom/.S* .
   ```

3. Repeat the above process for each of the CD's

Now you can start a NFS-installation from the directory `/usr/local/AUTOINSTALL`. The process is controlled by a few files including the file `/usr/local/AUTOINSTALL/suse/setup/descr/info`. This file is described in more detail in "*Control File*".

3.2. Installation from CD-ROM

An automated installation from CD-ROM really only makes sense if you use CDs created for this purpose. Basically the complete tree of a NFS server installation can be put onto a CD, providing you with automated installation-CD.

3.3. Installation from Hard drive

The hard drive can be configured in a number of ways. In short, the normal CDs need to be copied to the hard drive, and a few additional files need to be added, to make the automatic installation work.

To do an automated installation from Hard disk, you will need to place the sources of the SuSE version that you want to install on the hard drive. In addition, a few additions to these raw sources should be made. For example, it is necessary to create an info file and place it on the hard drive. Additionally, you may want to write some custom scripts, custom configuration files, custom RPMS, or package selection files for your installation.

Here are the steps that can accomplish the task.

1. Determine which packages you need to install on your system. This is a required step, as it will determine how big of a partition you will need to create on the hard drive to hold the sources.

   In this example, I will install the default system (without Star Office). Since all of these packages are included on the 1st CD of SuSE 7.3, I will need create a 1 Gig partition on the hard drive to hold the sources. The 1 GB partition is conservative and will allow some extra space for additional files.

2. Install SuSE on the target system, and create a partition at the end of the hard drive for the SuSE Sources.

   Although it is not required to have the sources in the last partition, it is used in this case. The reason that an installation is performed, is to verify that we have the correct partitions on the hard drive.

   Because the automatic installation will install from sources that are on a partition on the hard drive, it will not be possible to repartition the system during installation. The partitions must exist on the hard drive.

3. Copy the SuSE sources to the last partition of the hard drive.
4. Once the installation described above is complete, you need to add the SuSE sources to the hard drive. It is assumed that you created a partition, and file-system on the partition in question, and that it is the fourth primary partition on a SCSI hard drive.

   Mount the partition:

   ```
   mount /dev/sda4 /mnt
   ```

   Place the first SuSE CD in the carriage, and mount it:

   ```
   mount /cdrom
   ```

   create a subdirectory on the hard drive partition:

   ```
   mkdir /mnt/7.3
   ```

   Copy the sources from the CD to the hard drive

   ```
   cp -avr /cdrom/. /mnt/7.3
   ```

5. Add the necessary info file

   The info file must be placed in the directory `suse/setup/descr/` which, in the current example, can be found in `/mnt/7.3/suse/setup/descr`.

6. Add the necessary custom scripts, and optional configuration files. It is recommended that you create a directory for all of your custom scripts and configurations. In this example, we create the ADD_FILES directory for this.

   ```
   mkdir /mnt/7.3/ADD_FILES
   ```

   Add your custom scripts here. There are a few points during the installation which can be used to call the scripts, depending on what you need them to do.

7. Unmount the partition.

   At this point, your setup should be complete, and you can try an auto install.

Chapter 4. Control File

4.1. Description of the control file

This section describes the various entries in the various control files. Several of these control files may exist. There are at least two: one on the boot-disk and one on the installation medium. In terms of their format, the files are identical, although some entries only really make sense on the boot-disk (See: "*Specific entries for the boot-disk only*"). If the info file contains path names which refer to the installation medium itself, the wildcard "$I:" is used. This path cannot be specified exactly, because YaST only decides at runtime where the installation medium is to be mounted during installation. This means that the same info files can be used for NFS-, Hard-disk and CD-ROM-based automated installations.

4.2. Specific entries for the boot-disk only

The file `/suse/setup/descr/info` on the boot-disk contains the information which Linuxrc needs to set up language and keyboard and to gain access to the installation medium. These entries of course only make sense in the info file on the boot-disk. The individual entries have to following meaning:

Language

Description

Chooses the language of the installation.

Default

german

Valid entries

german and english

Display

Description

Determines, whether a color or monochrome monitor will be used.

Default

color

Valid entries

"color" and "mono"

Keytable

Description

Determines the keyboard table which is to be loaded.

Default

de-lat1-nd

Valid entries

"de-lat1-nd", "us", "fr-latin1", "it".

Bootmode

Description

Determines whether the installation medium is a CD or an NFS server.

Default

CD

Valid entries

Options available: "Harddisk", "CD" and "Net".

The following entries only make sense, when Bootmode has been set to "Net" They contain the information necessary for Linuxrc to access the installation medium via the network.

Netdevice

Description

Determines the names of the network devices.

Default

eth0

Valid entries

"eth0", "tr0", "fddi0".

Server

Description

Determines the IP-address of the NFS-server.

Default

none

Valid entries

the IP address of the NFS server.

Serverdir

Description

Contains the NFS installation directory on on the Harddisk or on the NFS server. This is the directory containing the sub-directory called `suse`. In the case of the last example the directory `/usr/local/AUTOINSTALL` should be entered here.

Default

none

Valid entries

A directory path.

IP

Description

Determines the IP-address of the machine that you are currently installing. If this keyword has not been set, the system tries to query a bootp or DHCP server for the IP address and all successive values.

Default

none

Valid entries

IP address of client or empty for BOOTP request.

Netmask

Description

Determines the netmask of the network.

Default

none

Valid entries

Valid netmask

Gateway

Description

If the NFS server is not located in the same sub-net as the machine to be installed, the IP-address of the gateway computer has to be entered here.

Default

none

Valid entries

Valid IP address of gateway

Nameserver

Description

This is where the Nameserver can be specified.

Default

none

Valid entries

Valid IP address of nameserver

insmod

Description

Determines the drivers that are necessary (SCSI and network card) for the installation. Should the automated installation be possible on a variety of SCSI controllers or network adapters you have to provide multiple 'insmod' statements.

Default

none

Valid entries

Valid kernel module names

Example

```
insmod aic7xxx
insmod ncr53c8xx
```

Similarly, one could add support for multiple network drivers as follows:

```
insmod 3c59x
insmod tulip
```

For this to work correctly, the modules have to be on the boot medium. Because of space limitations, it is not possible to include all of the modules on a single bootdisk. We will only be concerned with the modules that are necessary for the installation.

Example of an info file for an installation via NFS server:

```
Language: german            # "german" or "english"
Display: color              # "color" or "mono"
Keytable: us                # Keytable for us

Bootmode:   Net             # Needed for network installation

IP:         192.168.103.2
Netmask:    255.255.255.0
Gateway:    192.168.103.1
Server:     192.168.102.10
Serverdir:  /usr/local/AUTOINSTALL       # From this example
Nameserver: 192.168.102.1   # as the name says
Netdevice:  eth0

insmod aic7xxx
insmod ncr53c8xx
insmod rtl8139
insmod tulip
insmod 3c59x
```

Example 4-1. info file for an installation via NFS server

In addition to the keywords described above, the info file on the boot-disk can contain all the entries of a normal info file. These entries will be described later in this document.

4.3. Multiple control files

Multiple info files can be used for installation of different type of hardware and functionalities of the automatically installed client/server.

To be able to select between different control files, you should create a file info.lst in the root directory of the boot medium which contains a list of the location of the different control files with their description, separated by a colon.

```
/tmp/info.scsi:SCSI Systems
/tmp/info.ide:IDE Systems
```

Example 4-2. An info.lst file

If such a file is found, linuxrc will show a menu displaying the information above where you can select what control should be read. (linuxrc actually copies the file to `/info`).

4.4. Hierarchy of the various control files

As with an automated installation, several info files can be used to control the process and most entries can be in any info file. We now need to clarify which entry takes precedence in case several entries contradict each other.

The hierarchy works as follows (higher precedence first):

1. Info-file on the bootdisk
2. Class-specific info files
3. Info-file on the installation medium

Should several files contain the same entry, the entry from the file is valid, that is the topmost in the above list. This means that, if necessary, all entries on the installation medium can be disabled with a suitable bootdisk.

4.5. General configuration options

FAST_INSTALL

Description

The entry serves to centrally disable all features of an automated installation. If this keyword is set, a manual installation of the SuSE distribution can be executed from this installation medium.

Default

none

Valid entries

- 0: No automated installation.
- 1: The user is asked to confirm if the automated installation should be carried out.
- 2: Fully automated installation.

Example

```
FAST_INSTALL    2
```

AUTO_LILO

Description

This entry determines whether Lilo-configuration is to happen automatically. Lilo-entries called linux or linux.old , which refer to the kernels `/boot/vmlinuz` or `/boot/vmlinuz.old`, are assigned automatically. If a DOS-partition is present, this is also entered into Lilo.

Default

none

Valid entries

- 0: No automatic Lilo-configuration
- 1: At the end of the automatic Lilo-configuration, Lilo-output is displayed to the user. In case of problems, the user can repeat the Lilo-configuration manually.

- 2: Automatic Lilo-configuration as standard. Lilo-output is stored under `/var/adm/inst-log/lilo.inst` on the installed system.

Example

```
AUTO_LILO   2
```

LILO_DOS_NAME

Description

The automatic Lilo-configuration creates an entry for booting a DOS-partition, if this should be present on the system. This entry determines the Lilo-name for booting the DOS-partition in the Lilo-configuration.

Default

none

Valid entries

DOS or windows name to appear in lilo

Example

```
LILO_DOS_NAME   win98
```

AUTO_NET

Description

If networking is to be installed, the network configuration settings (Netdevice, IP-address, net mask, gateway) may automatically be adopted for the system that is being installed.

Default

none

Valid entries

- 0: No automatic network configuration
- 1: Network data is automatically configured for the system during installation.

Example

```
AUTO_NET   1
```

AUTO_NAME

Description

If an installation is carried out via bootp- or DHCP-server and if this server supplies a nameserver, the machine name specified by the nameserver can automatically be adopted for the system that is being installed.

Default

none

Valid entries

- 0: Machine name is not being adopted.
- 1: The machine name, that was determined by the nameserver for the IP address, is being adopted.

Example

```
AUTO_NAME   1
```

AUTO_NAMESERVER

Description

If an installation is carried out via bootp- or DHCP-server and if this server supplies a nameserver, the machine name specified by the nameserver can automatically be adopted for the system that is being installed.

Default

none

Valid entries

- 0: Nameserver is not adopted.
- 1: Nameserver is adopted for the system being installed.

Example

```
AUTO_NAMSERVER   1
```

AUTO_SERVICES

Description

Determines whether the user will be queried which network services are to be available, or not.

Default

none

Valid entries

- 0: Query concerning network services to be started.
- 1: Do not query concerning network services to be started.

Example

```
AUTO_SERVICES   1
```

AUTO_INSTALL

Description

This entry specifies the selection file that is to be used for automatic installation. The selection file contains the names of all packages on the SuSE distribution that are being installed. A path for the selection file can be entered. Within this path the string *$I:* stands for the path to the installation medium.

NOTE: It is best to copy the format of the existing selection files. One example is the `Minimal.sel` which is located on CD1 in `/suse/setup/descr/` directory. The same applies for the *ADD_INSTAL*L variable described below.

Default

none

Valid entries

Path to package selection file

Example

```
AUTO_INSTALL   $I:/suse/ADD_FILES/AUTO.sel
```

ADD_INSTALL

Description

With this entry additional selection files can be installed. Several entries containing the keyword *ADD_INSTAL* may be present. All selection files will be installed. In the context of class-specific installation this entry opens up the possibility of installing different sets of packages on certain classes of machines.

Default

none

Valid entries

Path to package selection file

Example

```
ADD_INSTALL   $I:/suse/ADD_FILES/Tex.sel
```

INSTALL_WAIT

Description

Once the packages have been installed, the user can check the installation logs. This entry serves to suppress the wait for user input. Installation logs are stored in `/var/adm/inst-log`

Default

none

Valid entries

- 0: Do not wait for user input after package installation.
- 1: Wait for user input after package installation.

Example

```
INSTALL_WAIT   0
```

AUTO_KERNEL

Description

Contains the name of the kernel to be installed. Since SuSE 6.4 you can specify either a kernel RPM or the name of a self compiled kernel. The self-compiled kernel has to be present on the installation medium under `suse/images/<Kernel name>.ikr`. at least the appropriate SCSI support has to be compiled into the kernel, if the installations happens on a SCSI system. The old method of installing self compiled kernels is still supported. Note that since 6.4 the SuSE CDs only contain the kernel in RPM format.

Default

none

Valid entries

Kernel name

Example

```
AUTO_KERNEL      k_deflt.rpm
```

default kernel RPM in `suse/images`

```
AUTO_KERNEL      mykernel
```

Self-compiled kernel in `suse/images/mykernel.ikr`

CDROM_DEVICE

Description

If the installation happens via NFS, this is where the device name for the CD-ROM-drive can be specified.

Default

none

Valid entries

CDROm device name

Example

```
CDROM_DEVICE   /dev/scd0
```

NO_ASK_SWAP

Description

This entry controls whether the question concerning to use of a swap partition will be asked.

Default

none

Valid entries

- 0: Question concerning use of swap-partition will be asked.
- 1: Question concerning use of swap-partition will not be asked.

Example

```
NO_ASK_SWAP   1
```

END_MESSAGE

Description

This entry controls whether the user will be asked at the end of the installation whether he wants to boot the system he has just installed.

Default

none

Valid entries

- 0: System will be booted automatically
- 1: The user will be asked whether the system should be booted now.

Example

```
END_MESSAGE   0
```

END_STARTUP

Description

This entry determines whether or not the message stating that the system is now installed appears at the end of the first start-up.

Default

none

Valid entries

- 0: Message is not displayed.
- 1: Message is displayed.

Example

```
END_STARTUP   0
```

CHECK_DEPENDENCY

Description

This entry controls whether the installation should be interrupted, while the automatic dependency check of the packages installed detects unresolved dependencies.

Default

none

Valid entries

- 0: No interruption in case of unresolved dependencies.
- 1: Interruption in case of unresolved dependencies.

Example

```
CHECK_DEPENDENCY   0
```

NEVER_STOP

Description

Determines that all queries will be answered with the default-value. This prevents windows waiting for user input from being displayed during automatic installation in case of problems. When this entry is set to '1' the installation will run smoothly and without interruption. It cannot be guaranteed however that a system installed in this way is usable.

Default

none

Valid entries

- 0: In case of unexpected problems the user will be queried.
- 1: The user is never queried. Installation will run through.

Example

```
NEVER_STOP   0
```

RC_CONFIG_0

Description

These entries can determine the contents of the file `/etc/rc.config`. Following the keyword *RC_CONFIG_0* the entries contain the name of the entry in `/etc/rc.config`. The rest of the line is taken up by the value set for this name in `/etc/rc.config`. The info-files read can contain any number of such entries. They are all incorporated into `/etc/rc.config`. If an entry with the same variable name is present several times in `/etc/rc.config`, the value for this variable is determined according to the rules

of hierarchy in "*Hierarchy of the various control files*". Note that you can also use this variable to configure a variable that will be in a file located in `/etc/rc.config.d/` .

Default

none

Valid entries

<Variable name in rc.config> <Value>

Example

```
RC_CONFIG_0      START_GPM  yes
```

RC_CONFIG_1

Description

Entries in `/etc/rc.config`, which can only be incorporated once the packages from the additional CDs have been installed. This entry only makes sense when an automatic installation from several CD-ROMs takes place. In the case of an installation via NFS or from a single CD, *RC_CONFIG_1* is functionally equivalent to *RC_CONFIG_0* and should therefore not be used.

Default

none

Valid entries

<Variable name in rc.config> <Value>

Example

```
RC_CONFIG_1      DBROOT  /opt/adabas
```

Apart from the entries for automatic installation the info-file on the installation medium also contains entries which describe the system to be installed itself. They are primarily used and analyzed when a system is updated using YaST. These entries only exist in the info-file on the installation medium and should not be edited.

This refers to the following entries:

```
ELF
MIN_YAST_VERSION
DIST_STRING
DIST_IDENT
MIN_DIST_VERSION
HAS_LIVE_CD
```

4.6. Entries controlling partitioning and formatting

This section covers the entries in the info file which are to do with partitioning and formatting the system to be installed.

AUTO_FDISK

Description

Determines how automatic partitioning is performed.

Default

none

Valid entries

0,1,2

- 0: No automatic partitioning
- 1: Automatic partitioning will be executed. Before the partitions are entered on the disks, the partitioning plan is displayed to the user who has to explicitly confirm the partitioning. It can also be edited manually at this stage.
- 2: Automatic Partitioning is carried out without user approval.

Example

```
AUTO_FDISK    1
```

AUTO_FDISK_DISK

Description

Contains a list of disk-device names which can be used for automatic installation. The installation will be performed on the disk corresponding to the first device name from the list. The partitioning of the disk is controlled through the files named `part_NNNNN` in the directory `suse/setup/descr` on the installation medium (where *NNNNN* is a five-digit number, which represents the size of a hard drive in MB, more on this below). The files named `part_NNNN` each contain partitioning plans, which are used during automated installation. The plans specify which partitions will be created and how large they should be. The files are designed to give flexibility to the system administrator in the event that automatic installations are being done on various sized hard drives.

The various `part_NNNNN` files which have been created with various partitioning plans are dynamically selected during the installation. Which `part_NNNNN` file gets used will depend on the size of the hard drive that is being partitioned. The file is selected by comparing the *NNNNN* number with the size of the hard drive. The highest *NNNNN number* will be chosen that is below the size of the hard drive in Megabytes.

Default

none

Valid entries

Device name representing the hard disk to be partitioned

Example

If you are planning to do automatic partitioning (the variable *AUTO_FDISK* is set to 2) of a 20 GB hard drive. There are two files `part_00000` and `part_25000` in the directory `suse/setup/descr`. The partitioning plan in the file `part_00000` will be used. In the above example, if you had a 40 GB hard drive, then the partitioning plan from the file `part_25000` would have been used.

```
AUTO_FDISK_DISK   /dev/hda /dev/sda
```

Example 4-3. Using *AUTO_FDISK_DISK*

Since SuSE 6.3 there is an alternate format for with a partitioning plan. This format allow more detailed specification about the partitioning and the created file-systems.

```
SWAP        256
/           2000
/var        3000
/usr        8000
/home       0
```

Example 4-4. Example file containing a partitioning plan (old format)

The file above would create the following partitioning on a SCSI-disk (Device `/dev/sda`): As first partition (`/dev/sda1`) a swap-partition of at least 256 MB will be created. The next partition will be the root-partition (`/dev/sda5`) of 2000 MB minimum. The next partitions will be /var (`/dev/sda6`) and /usr (`/dev/sda7`) with at least 3000 and 8000 MB each. The last partition to be created will be `/home`, which takes up the whole of the remaining disk space. The sizes mentioned are minimum sizes. The partitions can in fact be bigger, as only complete cylinders go into a partition. Several entries can have the keyword SWAP, so that several swap partitions can be created. If the last line lists the size 0 (as shown in the example above), the remainder of the hard drive will be allocated to this partition.

```
/      size=2000 id=83 bs=1024 err=c maxm=20 idns=4096 respc=5
SWAP   size=256  id=82
/var   size=3000 id=83 bs=1024 err=c maxm=20 idns=4096 respc=5
/usr   size=8000 id=83 bs=1024 err=c maxm=20 idns=4096 respc=5
/home  size=0   id=83 bs=1024 err=c maxm=20 idns=4096 respc=5
```

Example 4-5. A 'part_NNNNN' file using an alternate format

The above example will create the same partitioning scheme as the previous plan in the old format. By changing the values of the keywords you have more control over the various partitioning and filesystem options. The keyword are defined as follows:

Keyword	Example	Description
size=nnnn	size=400	Size in megabyte for the partition.
num=nn	num=3	Number of the partition. Be careful with the use of this keyword. Make sure that you do not destroy an existing partition. You are responsible for the results.
id=xx	id=8E	Partition Id written in Hex
bs=nnnn	bs=4096	Blocksize of the created ext2 filesystem.
idns=nnnn	idns=4096	Inode density for the partition. The value represents the average size of a file on the filesystem.
maxm=nnnn	maxm=999	Maximal mount count. After the partition has been mounted nnnn times, a filesystem check will be forced.
err=[c|r|p]	err=c	Reaction to errors in the filesystem (c=continue, r=remount-ro, p=panic)

Keyword	Example	Description
respc=nnnn	respc=2	Percentage of blocks reserved for root (respc and resbl may not be configured simultaneously).
resbl=nnnn	resbl=10000	Number of blocks reserved for root (respc and resbl may not be configured simultaneously).
fsys=reiser	fsys=reiser	The created filesystem is not ext2. Instead a reiser filesystem is created (available since SuSE 6.4). For a reiser filesystem the keywords bs, idns, maxm, err, respc and resbl are ignored.

Table 4-1. Alternate partitioning information

FSTAB_SEARCH

Description

Contains a list of partitions. All specified partitions are searched for the file `/etc/fstab`. If it is found on one of the partitions listed, partitioning does not happen automatically, but the contents of `/etc/fstab` will be read and the installation takes place according to the partitioning plan in this file. If you use this entry in combination with the following keyword (FSTAB_FORMAT) you determine that during automated installation partitions relevant to the system (such as `/var /usr`) will be installed from scratch, where as other partitions which are not relevant to the system (e.g. `/home /space`) are not effected.

Default

none

Valid entries

Device names of partitions to be searched for

Example

```
FSTAB_SEARCH   /dev/sda1 /dev/hda1
```

FSTAB_FORMAT

Description

Contains the mount points of the partitions which are regarded as relevant to the system. This entry really only makes sense in combination with the previous entry (FSTAB_SEARCH). It includes the mount points of those partitions which during automated installation have to be re-initialized after an existing file `/etc/fstab` has been read. All partitions whose mount points are not listed remain untouched during such an installation.

Default

none

Valid entries

Mount point names of partitions to be formatted

Example

```
FSTAB_FORMAT   / /var /usr
```

AUTO_FDISK_TABLE

Description

This entry allows you to assign a specific partitioning file to a particular hard-disk. The entry contains the keyword, the device name of a hard-disk and the path name to the file containing the partitioning plan. Within this path name the wildcard *"$I:"* stands for the path to the medium for installation.

Several entries with the keyword *AUTO_FDISK_TABLE* may exist. If they refer to the same disk, the partitioning file will be handled according to the hierarchy discussed in "*Hierarchy of the various control files*". If the entries refer to different disks, the entries from all partitioning files are taken together and several disks used for installation. It is then up to the user to make sure that these partitioning files fit into a valid filesystem hierarchy. This means that there has to be an entry for the root file system and that no mount point may be assigned several times.

Entries referring to hard-disks without write access will be ignored. If at least one *AUTO_FDISK_TABLE* entry with a writable hard-disk is found, this is used for partitioning and the entries in *AUTO_FDISK_DISK* will be ignored.

Default

none

Valid entries

Mount point names of partitions to be formatted

Example

```
AUTO_FDISK_TABLE   /dev/sdb $I:/suse/ADD_FILES/part_test
AUTO_FDISK_TABLE   /dev/sdc $D:/part_test
```

AUTO_REMOVE_PART

Description

YaST will try to remove the partitions defined with the keyword from the hard-disk <diskname>

Default

none

Valid entries

AUTO_REMOVE_PART <diskname> <list of partitions to remove>

Example

AUTO_REMOVE_PART /dev/sdb 2 3 4 5 6 7

AUTO_KEEP_DISK

Description

YaST tries to format the disk <diskname> according to the information provided in the disk layout description. If there is an entry *AUTO_REMOVE_PART* for the same disk, the partitions are removed first, otherwise the existing partitions in <diskname> are not removed.

Default

none

Valid entries

AUTO_KEEP_DISK <diskname> <path to disk layout description>

Example

AUTO_KEEP_DISK /dev/sdb $I:/suse/ADD_FILES/part_disk2

4.7. Integration of customized scripts and packages

During an automated installation packages can be installed which are not part of the SuSE Linux distribution. The customer can choose these packages according to his own needs and simply integrate them into the automatic installation. It is also possible to execute customized scripts at any point during the installation.

There are three points during the installation which can be used for customized scripts and packages:

1. Before the installation of the first package from the SuSE distribution. These are the entries marked with the keywords *PRE_INSTALL* or *PRE_SCRIPT* in the info-file.
2. After the installation of all packages on the SuSE distribution, which can be installed without a change of media. During an installation from CD-ROM this corresponds to all packages on the first CD. These are the entries marked with the keywords *POST_INSTALL* or*POST_SCRIPT* in the info-file.
3. After the installation of the last package on the SuSE distribution. These are the entries marked with the keywords *LAST_INSTALL* or *LAST_SCRIPT* in the info-file. These entries only make sense when an automatic installation from several CD-ROMs is taking place. If the installation happens via NFS or from a single CD-ROM, LAST is functionally equivalent with POST and should therefore not be used.

The packages should be available in RPM-format. For reasons of backward compatibility tgz-archives can also be installed. As these are not present in the RPM-database however we do not recommend this.

At all three points any number of scripts can be started. As their first parameter the scripts contain the directory where they are located. This is also where you put additional files which are to be copied into the distribution. The second parameter to be passed on is PRE, POST or LAST, depending on when the script was called. The scripts are called up with "sh -x"and all output is directed into `/var/adm/inst-log/<Scriptname>.log`.

When you are creating your scripts, it is important to remember that the script will not understand the variable $I (it represents the directory on the installation medium), which is used in the info file (described above). When you specify a directory on the installation medium from within a script, you will have to use the full path, which will start with `/var/adm/mount/suse`. `/var/adm/mount` is the directory under which the installation medium is mounted during the installation. Additionally the path to the script is provided as first parameter of the script, so every install script can use $1 to access the directory where the script itself is located.

Timing of Post-script execution: Post-scripts are not executed in the installed system. Those scripts executed before the installed system is booted (while the installed partitions are still mounted in `/mnt`).

Thus, any changes required for the installed system should be executed in the directory where the system is mounted (`/mnt`).

AUTO_INIT_CMD

Description

A script that is executed by YaST before anything regarding automated partitioning is done. Here one could for example place a call to GNU-parted to shrink a existing DOS partition.

Default

none

Valid entries

Path to script

Example

```
AUTO_INIT_CMD $I:/suse/ADD_FILES/prepare_disk
```

PRE_INSTALL

Description

Insert the package specified into the list of packages which are to be installed before any packages of the SuSE distribution.

Default

none

Valid entries

Path to rpm package

Example

```
PRE_INSTALL  $I:/suse/ADD_FILES/preinst.rpm
```

POST_INSTALL

Description

Inserts the package specified into the list of packages which are to be installed after all the packages in the SuSE distribution which can be installed without change of media.

Default

none

Valid entries

Path to rpm package

Example

```
POST_INSTALL  $I:/suse/ADD_FILES/postinst.rpm
```

LAST_INSTALL

Description

Inserts the package specified into the list of packages which are to be installed after all packages in the SuSE distribution. This entry only makes sense during an automated installation form several CD-ROMs. If the installation happens via NFS or from a single CD-ROM, *LAST_INSTALL* is functionally equivalent with *POST_INSTALL* and should therefore not be used.

Default

none

Valid entries

Path to rpm package

Example

```
LAST_INSTALL  $I:/suse/ADD_FILES/lastinst.rpm
```

PRE_SCRIPT

Description

Incorporates the script specified into the list of scripts, which are to be executed before the first package from the SuSE distribution is installed.

Default

none

Valid entries

Path to script

Example

```
PRE_SCRIPT  $I:/suse/ADD_FILES/scripts/prepare
```

POST_SCRIPT

Description

Incorporates the script specified into the list of scripts that are to be executed after all the packages from the SuSE distribution, which are installable without change of media, have been installed.

Default

none

Valid entries

Path to script

Example

```
POST_SCRIPT  $I:/suse/ADD_FILES/scripts/add_user
```

LAST_SCRIPT

Description

Inserts the script specified into the list of scripts which are to be executed after all the packages from the SuSE distribution have been installed. This entry only makes sense

when an automated installation from several CD-ROMs is taking place. During an installation via NFS or from a single CD-ROM LAST_SCRIPT is functionally equivalent with POST_SCRIPT and should therefore not be used.

Default

none

Valid entries

Path to script

Example

```
LAST_SCRIPT  $I:/suse/ADD_FILES/scripts/last
```

4.8. Class-specific installation

It is possible to determine classes of machines for installation and to specify deviations from or additions to the standard installation for these classes. For each of these groups an additional info-file can be specified which contains different settings. These settings will overwrite or supplement the entries in the central info-file. The prefix info. is added to the name specified in the class definition. If tex is specified as the name of the supplementing info file in the class definition, the file info.tex from the directory suse/setup/descr on the installation medium is read.

To determine the membership in a class of machines a list of machine names and / or IP-addresses is supplied. For machine names '*' can be used as a wildcard, for IP-addresses ranges can be specified. Therefore specifications such as *.subnet.company.de as well as 192.168.102.20-40 are both possible. In the list several comma-separated areas can be strung together.

A machine can belong to several classes. In this case the info-files of the various classes are read one after one. The definition of machine classes really only makes sense if the IP-addresses are assigned automatically during an installation via a bootp or DHCP server.

```
CLASS    192.168.102.1-20,192.168.103.100-110   tex
CLASS    *.subnet.company.de                    x11
```

Example 4-6. Class configuration

Appendix A. Example control files

The following control file will let YaST perform a fully automated installation without any user interaction required. Note that YaST needs some values to be set so that it won't prompt for user interaction, for example the values concerning mouse and modem.

```
Language: english
Keytable: us
insmod via-rhine
Display: color
Bootmode: Net
IP:
Serverdir: /tftpboot/CDs
CDROM_DEVICE /dev/hda
NEVER_STOP   1
END_STARTUP   0
INSTALL_WAIT   0
FAST_INSTALL   2
END_MESSAGE   0

AUTO_LILO   2

AUTO_FDISK      2
AUTO_FDISK_DISK /dev/hdb

AUTO_FDISK_TABLE  /dev/hdb $I:/part/mytest


CHECK_DEPENDENCY   0
AUTO_KERNEL   k_deflt.rpm
AUTO_INSTALL   $I:/suse/setup/descr/Minimal.sel

AUTO_NET   1
AUTO_SERVICES   1
AUTO_NAMESERVER   1
AUTO_NAME   1
PRE_SCRIPT   $I:/scripts/init.sh
POST_SCRIPT   $I:/scripts/setup.sh
POST_SCRIPT   $I:/scripts/boot.sh
RC_CONFIG_0    START_GPM    no
RC_CONFIG_0    MOUSE    /dev/psaux
RC_CONFIG_0    TIMEZONE    Canada/Eastern
RC_CONFIG_0    SET_ROOT_PW    yes
RC_CONFIG_0    SET_START_PW    8lcNGAWj7Bx9I
RC_CONFIG_0    GMT    -u
RC_CONFIG_0    SMTP_TYPE    YES
RC_CONFIG_0    MODEM
```

Appendix B. Example user scripts

As noted in an early section of this document, user scripts are executed in the installation system meaning that all actions are done while the installed system is still mounted in a temporary directory. To be able to execute scripts in the running system you can add the following scripts as post scripts in the info file.

```
#!/bin/sh

if [ ! -f "/mnt/etc/passwd" ]; then
  echo "Running in installed system, aborting"
  exit
fi

target_mount=/mnt

# Check if there are scripts we want to copy to the installed system

cp /var/adm/mount/scripts/* /mnt/var/adm/scripts/ || echo "Failed copying user scripts"


#
# Try to set a default root password (encrypted)
#
epwd="8lcNGAWj7Bx9I"
   cp $target_mount/etc/shadow $target_mount/etc/shadow.old
   $target_mount/usr/bin/awk -F: ’
      $1 == "root" && $2 == ""  { printf "%s:%s:%s:%s:%s:%s:%s:%s:%s\n",$1,epwd,$3,$4,$5,
      $1 == "root" && $2 != ""  { print $0 }     # never change root passwd, if it is al
ready set
      $1 != "root" { print $0 }’ epwd=$epwd $target_mount/etc/shadow.old > $target_mount/


touch /mnt/var/state/autoyast_run
cat <<EOF >/mnt/etc/init.d/boot.autoyast
#! /bin/sh
#
# Copyright (c) 2000 SuSE
#
# Copied from alice
# /etc/init.d/boot.autoyast

### BEGIN INIT INFO
# Provides:          boot.autoyast
# Required-Start:    \$network portmap route
# Required-Stop:
# Default-Start:     2 3 5
# Default-Stop:
# Description:       Runs the AutoYaST post-(post)-scripts
### END INIT INFO
#
if [ ! -e /var/state/autoyast_run ]; then
  exit
fi
. /etc/rc.config
#
# Tell the world we´re running
#
echo "Running AutoYaST post-post-installation scripts..."

for script in \‘ls /var/adm/scripts/*\‘; do
  echo "Executing \$script"
  sh \$script
done


# Your scripts here
```

```
#
#
#

# End of user scripts


rm /var/state/autoyast_run
echo -n "AutoYaST post-post-installation scripts have finished"
echo -e "\$rc_done"
EOF



#
# Set permissions for boot.alice
#
/mnt/bin/chmod a+x /mnt/etc/init.d/boot.autoyast
#
# Link boot.autoyast into target system´s bootup sequence
#
if [ ${target_mount}/sbin/insserv ]; then
  /mnt/sbin/insserv /mnt/etc/init.d/boot.autoyast
else
  ln -s /etc/init.d/boot.autoyast \
        /mnt/etc/init.d/rc2.d/S09boot.autoyast
  # also create a link for runlevel 3 (if the installation reboots
  # (cause of kernel change) beginning with 7.1 the runlevel is 3
  # and not 2 like before)
  ln -s /etc/init.d/boot.autoyast \
        /mnt/etc/init.d/rc3.d/S09boot.autoyast
fi
mkdir -p /mnt/var/state
touch /mnt/var/state/autoyast_run
```

This script will first set root password (root password "blank" in this example) and then create a boot script which is executed in the specified run level to allow various configuration tasks to take place.

You can either include you configuration scripts into this file, where inidicated, or you can create other post-install scripts which are not executed in the installation system. Those scripts will later be copied using the above script and executed in the running system. The following is an example of such a configuration and setup script:

```
#!bin/sh
# Verfiy we are not running in the installation system!
if [ -f "/mnt/etc/passwd" ]; then
  echo "Still installing, this script will be executed later"
  exit
fi

touch /tmp/testing_autoyast
```