

Linux Enterprise Server 11 SP2

www.suse.com

February 1, 2012

SLES 11 SP2: Storage Administration Guide



SLES 11 SP2: Storage Administration Guide

Legal Notices

Copyright © 2006–2012 Novell, Inc. and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled “GNU Free Documentation License”.

All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither Novell, Inc., SUSE LINUX Products GmbH, the authors, nor the translators shall be held liable for possible errors or the consequences thereof.

Trademarks

For Novell trademarks, see the Novell Trademark and Service Mark list [<http://www.novell.com/company/legal/trademarks/tmlist.html>].

Linux* is a registered trademark of Linus Torvalds. All other third party trademarks are the property of their respective owners.

A trademark symbol (®, ™, etc.) denotes a Novell trademark; an asterisk (*) denotes a third party trademark.

Contents

About This Guide	ix
1 Overview of File Systems in Linux	1
1.1 Terminology	2
1.2 Major File Systems in Linux	2
1.3 Other Supported File Systems	11
1.4 Large File Support in Linux	12
1.5 Managing Devices with the YaST2 Partitioner	13
1.6 Additional Information	14
2 What's New for Storage in SLES 11	17
2.1 What's New in SLES 11 SP2	17
2.2 What's New in SLES 11 SP1	18
2.3 What's New in SLES 11	22
3 Planning a Storage Solution	29
3.1 Partitioning Devices	29
3.2 Multipath Support	29
3.3 Software RAID Support	30
3.4 File System Snapshots	30
3.5 Backup and Antivirus Support	30
4 LVM Configuration	33
4.1 Understanding the Logical Volume Manager	34
4.2 Creating LVM Partitions	37
4.3 Creating Volume Groups	39

4.4	Configuring Physical Volumes	41
4.5	Configuring Logical Volumes	43
4.6	Tagging LVM2 Storage Objects	46
4.7	Resizing a Volume Group	55
4.8	Resizing a Logical Volume with YaST	57
4.9	Resizing a Logical Volume with Commands	58
4.10	Deleting a Volume Group	59
4.11	Deleting an LVM Partition (Physical Volume)	60
5	Resizing File Systems	61
5.1	Guidelines for Resizing	61
5.2	Increasing the Size of an Ext2, Ext3, or Ext4 File System	63
5.3	Increasing the Size of a Reiser File System	64
5.4	Decreasing the Size of an Ext2 or Ext3 File System	66
5.5	Decreasing the Size of a Reiser File System	67
6	Using UUIDs to Mount Devices	69
6.1	Naming Devices with udev	69
6.2	Understanding UUIDs	70
6.3	Using UUIDs in the Boot Loader and /etc/fstab File (x86)	71
6.4	Using UUIDs in the Boot Loader and /etc/fstab File (IA64)	74
6.5	Additional Information	75
7	Managing Multipath I/O for Devices	77
7.1	Understanding Multipathing	78
7.2	Planning for Multipathing	78
7.3	Multipath Management Tools	89
7.4	Configuring the System for Multipathing	98
7.5	Enabling and Starting Multipath I/O Services	111
7.6	Configuring Path Failover Policies and Priorities	112
7.7	Configuring Multipath I/O for the Root Device	126
7.8	Configuring Multipath I/O for an Existing Software RAID	131
7.9	Scanning for New Devices without Rebooting	134
7.10	Scanning for New Partitioned Devices without Rebooting	137
7.11	Viewing Multipath I/O Status	139
7.12	Managing I/O in Error Situations	140
7.13	Resolving Stalled I/O	142
7.14	Troubleshooting MPIO	142
7.15	What's Next	143

8	Software RAID Configuration	145
8.1	Understanding RAID Levels	146
8.2	Soft RAID Configuration with YaST	148
8.3	Troubleshooting Software RAIDs	150
8.4	For More Information	151
9	Configuring Software RAID for the Root Partition	153
9.1	Prerequisites for the Software RAID	153
9.2	Enabling iSCSI Initiator Support at Install Time	154
9.3	Enabling Multipath I/O Support at Install Time	155
9.4	Creating a Software RAID Device for the Root (/) Partition	155
10	Managing Software RAIDs 6 and 10 with mdadm	161
10.1	Creating a RAID 6	161
10.2	Creating Nested RAID 10 Devices with mdadm	163
10.3	Creating a Complex RAID 10 with mdadm	168
10.4	Creating a Degraded RAID Array	173
11	Resizing Software RAID Arrays with mdadm	177
11.1	Understanding the Resizing Process	177
11.2	Increasing the Size of a Software RAID	179
11.3	Decreasing the Size of a Software RAID	186
12	iSNS for Linux	193
12.1	How iSNS Works	194
12.2	Installing iSNS Server for Linux	195
12.3	Configuring iSNS Discovery Domains	197
12.4	Starting iSNS	204
12.5	Stopping iSNS	204
12.6	For More Information	205
13	Mass Storage over IP Networks: iSCSI	207
13.1	Installing iSCSI	210
13.2	Setting Up an iSCSI Target	211
13.3	Configuring iSCSI Initiator	222
13.4	Using iSCSI Disks when Installing	229
13.5	Troubleshooting iSCSI	229
13.6	Additional Information	232

14	Fibre Channel Storage over Ethernet Networks: FCoE	233
14.1	Installing FCoE and the YaST2 FCoE Client	236
14.2	Configuring FCoE Interfaces during the Installation	237
14.3	Managing FCoE Services with YaST	238
14.4	Configuring FCoE with Commands	243
14.5	Managing FCoE Instances with the FCoE Administration Tool	245
14.6	Setting Up Partitions for an FCoE Initiator Disk	249
14.7	Creating a File System on an FCoE Initiator Disk	250
14.8	Additional Information	251
15	Volume Snapshots	253
15.1	Understanding Volume Snapshots	253
15.2	Creating Linux Snapshots with LVM	255
15.3	Monitoring a Snapshot	255
15.4	Deleting Linux Snapshots	256
16	Managing Access Control Lists over NFSv4	257
17	Troubleshooting Storage Issues	259
17.1	Is DM-MPIO Available for the Boot Partition?	259
17.2	Issues for iSCSI	259
17.3	Issues for Multipath I/O	260
17.4	Issues for Software RAIDs	260
A	GNU Licenses	261
A.1	GNU General Public License	261
A.2	GNU Free Documentation License	264
B	Documentation Updates	269
B.1	January 18, 2012 (SLES 11 SP2)	270
B.2	July 12, 2011	274
B.3	June 14, 2011	274
B.4	May 5, 2011	275
B.5	January 2011	275
B.6	September 16, 2010	276
B.7	June 21, 2010	277
B.8	May 2010 (SLES 11 SP1)	279
B.9	February 23, 2010	281
B.10	December 1, 2009	282
B.11	October 20, 2009	284

B.12	August 3, 2009	285
B.13	June 22, 2009	286
B.14	May 21, 2009	288

About This Guide

This guide provides information about how to manage storage devices on a SUSE Linux Enterprise Server 11 Support Pack 2 (SP2) server.

Audience

This guide is intended for system administrators.

Feedback

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

Documentation Updates

For the most recent version of the *SUSE Linux Enterprise Server 11 Storage Administration Guide*, visit the SUSE Documentation Web site for SUSE Linux Enterprise Server 11 [<http://www.suse.com/documentation/sles11>].

Additional Documentation

For information about partitioning and managing devices, see “Advanced Disk Setup” [http://www.suse.com/documentation/sles11/book_sle_deployment/data/cha_advdisk.html] in the *SUSE Linux Enterprise Server 11 Deployment Guide* [http://www.suse.com/documentation/sles11/book_sle_deployment/data/pre_sle.html].

Overview of File Systems in Linux

1

SUSE Linux Enterprise Server ships with a number of different file systems from which to choose, including Ext3, Ext2, ReiserFS, and XFS. Each file system has its own advantages and disadvantages.

Professional high-performance setups might require a highly available storage systems. To meet the requirements of high-performance clustering scenarios, SUSE Linux Enterprise Server includes OCFS2 (Oracle Cluster File System 2) and the Distributed Replicated Block Device (DRBD) in the SLES High-Availability Storage Infrastructure (HASI) release. These advanced storage systems are not covered in this guide. For information, see the *SUSE Linux Enterprise 11 SP2 High Availability Extension Guide* [http://www.suse.com/documentation/sle_ha/book_sleha/data/book_sleha.html].

- Section 1.1, “Terminology” (page 2)
- Section 1.2, “Major File Systems in Linux” (page 2)
- Section 1.3, “Other Supported File Systems” (page 11)
- Section 1.4, “Large File Support in Linux” (page 12)
- Section 1.5, “Managing Devices with the YaST2 Partitioner” (page 13)
- Section 1.6, “Additional Information” (page 14)

1.1 Terminology

metadata

A data structure that is internal to the file system. It assures that all of the on-disk data is properly organized and accessible. Essentially, it is “data about the data.” Almost every file system has its own structure of metadata, which is on reason that the file systems show different performance characteristics. It is extremely important to maintain metadata intact, because otherwise all data on the file system could become inaccessible.

inode

A data structure on a file system that contains various information about a file, including size, number of links, pointers to the disk blocks where the file contents are actually stored, and date and time of creation, modification, and access.

journal

In the context of a file system, a journal is an on-disk structure containing a type of log in which the file system stores what it is about to change in the file system’s metadata. Journaling greatly reduces the recovery time of a file system because it has no need for the lengthy search process that checks the entire file system at system startup. Instead, only the journal is replayed.

1.2 Major File Systems in Linux

SUSE Linux Enterprise Server offers a variety of file systems from which to choose. This section contains an overview of how these file systems work and which advantages they offer.

It is very important to remember that no file system best suits all kinds of applications. Each file system has its particular strengths and weaknesses, which must be taken into account. In addition, even the most sophisticated file system cannot replace a reasonable backup strategy.

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

IMPORTANT

Unless stated otherwise in this section, all the steps required to set up or change partitions and file systems can be performed by using YaST.

- Section 1.2.1, “BtrFS” (page 3)
- Section 1.2.2, “Ext2” (page 5)
- Section 1.2.3, “Ext3” (page 6)
- Section 1.2.4, “ReiserFS” (page 8)
- Section 1.2.5, “XFS” (page 9)
- Section 1.2.6, “Feature Comparison” (page 10)

1.2.1 BtrFS

BtrFS (Better File System) is a copy-on-write (COW) file system developed by Chris Mason. It is based on COW-friendly B-trees developed by Ohad Rodeh. BtrFS is a logging-style file system. Instead of journaling the block changes, it writes them in a new location, then links the change in. Until the last write, the new changes are not committed.

IMPORTANT

Because BtrFS is capable of storing snapshots of the file system, it is advisable to reserve more disk space (double) than the standard storage proposal.

BtrFS provides fault tolerance, repair, and easy management features, such as the following:

- Writable snapshots that allow you to easily roll back your system if needed after applying updates, or to back up files.
- Multiple device support that allows you to grow or shrink the file system.
- Compression to efficiently use storage space.

- Different RAID levels for metadata and user data.
- Different checksums for metadata and user data to improve error detection.
- Integration with Linux Logical Volume Manager (LVM) storage objects.
- Integration with the YaST Partitioner on SUSE Linux.

Btrfs creates a default subvolume in its assigned pool of space. It allows you to create additional subvolumes that act as individual file systems within the same pool of space. The number of subvolumes is limited only by the space allocated to the pool.

If Btrfs is used for the root (/) file system, you can cover any subdirectory as a subvolume as you might normally do. You should also consider covering the following subdirectories in separate subvolumes because they contain files that you might prefer not to snapshot for the reasons given:

Path	Reason to Cover as a Subvolume
/opt	Contains third-party add-on application software packages.
/srv	Contains <code>http</code> and <code>ftp</code> files.
/tmp	Contains temporary files.
/var/log	Contains log files.
/var/opt	Contains run-time variable data for <code>/opt</code> .
/var/run	Contains run-time variable data.
/var/spool	Contains data that is awaiting processing by a program, user, or administrator, such as news, mail, and printer queues.
/var/tmp	Contains temporary files or directories that are preserved between system reboots.

1.2.2 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. The 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 short recovery times, Ext2 became less important.

A brief summary of Ext2's strengths might help understand why it was—and in some areas still is—the favorite Linux file system of many Linux users.

- Section “Solidity and Speed” (page 5)
- Section “Easy Upgradability” (page 5)

Solidity and Speed

Being quite an “old-timer,” Ext2 underwent many improvements and was heavily tested. This might 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 journaling file systems, `e2fsck` analyzes the entire file system and not just the recently modified bits of metadata. This takes significantly longer than checking the log data of a journaling file system. Depending on file system size, this procedure can take half an hour or more. Therefore, it is not desirable to choose Ext2 for any server that needs high availability. However, because Ext2 does not maintain a journal and uses significantly less memory, it is sometimes faster than other file systems.

Easy Upgradability

Because 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 very easy.

1.2.3 Ext3

Ext3 was designed by Stephen Tweedie. Unlike 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. In summary, Ext3 has three major advantages to offer:

- Section “Easy and Highly Reliable Upgrades from Ext2” (page 6)
- Section “Reliability and Performance” (page 6)
- Section “Converting an Ext2 File System into Ext3” (page 7)

Easy and Highly Reliable Upgrades from Ext2

The code for Ext2 is the strong foundation on which Ext3 could become a highly-acclaimed next-generation file system. Its reliability and solidity are elegantly combined in Ext3 with the advantages of a journaling file system. Unlike transitions to other journaling file systems, such as ReiserFS or XFS, which can be quite tedious (making backups of the entire file system and recreating it from scratch), a transition to Ext3 is a matter of minutes. It is also very safe, because re-creating an entire 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 see 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 remount it as an Ext2 file system.

Reliability and Performance

Some other journaling file systems follow the “metadata-only” journaling approach. This means your metadata is always kept in a consistent state, but this 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 (data integrity), but can slow down the system because 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 data blocks are 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. Ext3 uses the `data=ordered` option as the default.

Converting an Ext2 File System into Ext3

To convert an Ext2 file system to Ext3:

- 1 Create an Ext3 journal by running `tune2fs -j` as the `root` user.

This creates an Ext3 journal with the default parameters.

To specify how large the journal should be and on which device it should reside, run `tune2fs -J` instead together with the desired journal options `size=` and `device=`. More information about the `tune2fs` program is available in the `tune2fs` man page.

- 2 Edit the file `/etc/fstab` as the `root` user to change the file system type specified for the corresponding partition from `ext2` to `ext3`, then save the changes.

This ensures that the Ext3 file system is recognized as such. The change takes effect after the next reboot.

- 3 To boot a root file system that is set up as an Ext3 partition, include the modules `ext3` and `jbd` in the `initrd`.

3a Edit `/etc/sysconfig/kernel` as `root`, adding `ext3` and `jbd` to the `INITRD_MODULES` variable, then save the changes.

3b Run the `mkinitrd` command.

This builds a new `initrd` and prepares it for use.

- 4 Reboot the system.

1.2.4 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 version 6.4. ReiserFS was designed by Hans Reiser and the Namesys development team. It has proven itself to be a powerful alternative to Ext2. Its key assets are better disk space utilization, better disk access performance, faster crash recovery, and reliability through data journaling.

IMPORTANT

The ReiserFS file system is fully supported for the lifetime of SUSE Linux Enterprise Server 11 specifically for migration purposes. SUSE plans to remove support for creating new ReiserFS file systems starting with SUSE Linux Enterprise Server 12.

- Section “Better Disk Space Utilization” (page 8)
- Section “Better Disk Access Performance” (page 8)
- Section “Fast Crash Recovery” (page 9)
- Section “Reliability through Data Journaling” (page 9)

Better Disk Space Utilization

In ReiserFS, all data is organized in a structure called a B*-balanced tree. The tree structure contributes to better disk space utilization because 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 needed. Another benefit lies in the dynamic allocation of inodes. This keeps the file system more flexible than traditional file systems, like Ext2, where the inode density must be specified at file system creation time.

Better Disk Access Performance

For small files, file data and “stat_data” (inode) information are often stored next to each other. They can be read with a single disk I/O 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.

Reliability through Data Journaling

ReiserFS also supports data journaling and ordered data modes similar to the concepts outlined in Section 1.2.3, “Ext3” (page 6). The default mode is `data=ordered`, which ensures both data and metadata integrity, but uses journaling only for metadata.

1.2.5 XFS

Originally intended as the file system for their IRIX OS, SGI started XFS development in the early 1990s. The idea behind XFS was to create a high-performance 64-bit journaling file system to meet extreme computing challenges. XFS is very good at manipulating large files and performs well on high-end hardware. However, even XFS has a drawback. Like ReiserFS, XFS takes great care of metadata integrity, but less care of data integrity.

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

- Section “High Scalability through the Use of Allocation Groups” (page 9)
- Section “High Performance through Efficient Management of Disk Space” (page 10)
- Section “Preallocation to Avoid File System Fragmentation” (page 10)

High Scalability through the Use of Allocation Groups

At the 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. Because 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. XFS uses *delayed allocation*, which 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 (in file system blocks) the data should be stored. This decision is delayed until the last possible moment. Some short-lived temporary data might never make its way to disk, because it is obsolete by the time XFS decides where actually to save it. In this way, XFS increases write performance and reduces file system fragmentation. Because 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 because the contents of a file are not distributed all over the file system.

1.2.6 Feature Comparison

For a side-by-side feature comparison of the major operating systems in SUSE Linux Enterprise Server, see File System Support and Sizes [<http://www.suse.com/products/server/technical-information/#FileSystem>] on the SUSE Linux Enterprise Server Technical Information Web site [<http://www.suse.com/products/server/technical-information/>].

1.3 Other Supported File Systems

Table 1.1, “File System Types in Linux” (page 11) 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.

Table 1.1 *File System Types in Linux*

File System Type	Description
<code>cramfs</code>	Compressed ROM file system: A compressed read-only file system for ROMs.
<code>hpfs</code>	High Performance File System: The IBM OS/2 standard file system. Only supported in read-only mode.
<code>iso9660</code>	Standard file system on CD-ROMs.
<code>minix</code>	This file system originated from academic projects on operating systems and was the first file system used in Linux. Today, it is used as a file system for floppy disks.
<code>msdos</code>	<code>fat</code> , the file system originally used by DOS, is today used by various operating systems.
<code>ncpfs</code>	File system for mounting Novell volumes over networks.
<code>nfs</code>	Network File System: Here, data can be stored on any machine in a network and access might be granted via a network.
<code>ntfs</code>	Windows NT file system; read-only.
<code>smbfs</code>	Server Message Block is used by products such as Windows to enable file access over a network.
<code>sysv</code>	Used on SCO UNIX, Xenix, and Coherent (commercial UNIX systems for PCs).

File System Type	Description
ufs	Used by BSD, SunOS, and NextStep. Only supported in read-only mode.
umsdos	UNIX on MS-DOS: Applied on top of a standard <code>fat</code> file system, achieves UNIX functionality (permissions, links, long filenames) by creating special files.
vfat	Virtual FAT: Extension of the <code>fat</code> file system (supports long filenames).

1.4 Large File Support in Linux

Originally, Linux supported a maximum file size of 2 GB (2^{31} bytes). Currently all of our standard file systems have LFS (large file support), which gives a maximum file size of 2^{63} bytes in theory. The numbers given in the following table assume that the file systems are using 4 KiB block size. When using different block sizes, the results are different, but 4 KiB reflects the most common standard.

NOTE

In this document: 1024 Bytes = 1 KiB; 1024 KiB = 1 MiB; 1024 MiB = 1 GiB; 1024 GiB = 1 TiB; 1024 TiB = 1 PiB; 1024 PiB = 1 EiB (see also *NIST: Prefixes for Binary Multiples* [<http://physics.nist.gov/cuu/Units/binary.html>]).

Table 1.2, “Maximum Sizes of Files and File Systems (On-Disk Format)” (page 12) offers an overview of the current limitations of Linux files and file systems.

Table 1.2 *Maximum Sizes of Files and File Systems (On-Disk Format)*

File System (4 KiB Block Size)	Maximum File Size	Maximum File System Size
BtrFS	16 EiB	16 EiB
Ext2 or Ext3	2 TiB	16 TiB

File System (4 KiB Block Size)	Maximum File Size	Maximum File System Size
OCFS2 (available in the High Availability Extension)	4 PiB	4 PiB
ReiserFS v3	2 TiB	16 TiB
XFS	8 EiB	8 EiB
NFSv2 (client side)	2 GiB	8 EiB
NFSv3 (client side)	8 EiB	8 EiB

IMPORTANT

Table 1.2, “Maximum Sizes of Files and File Systems (On-Disk Format)” (page 12) describes the limitations regarding the on-disk format. The Linux kernel imposes its own limits on the size of files and file systems handled by it. These are as follows:

File Size

On 32-bit systems, files cannot exceed 2 TiB (2^{41} bytes).

File System Size

File systems can be up to 2^{73} bytes in size. However, this limit is still out of reach for the currently available hardware.

1.5 Managing Devices with the YaST2 Partitioner

You can use the YaST2 Partitioner to create and manage file systems and RAID devices. For information, see “Advanced Disk Setup” [http://www.suse.com/documentation/sles11/book_sle_deployment/data/cha_advdisk.html] in the *SUSE Linux Enterprise Server 11 SP2 Deployment Guide* [<http://>

www.suse.com/documentation/sles11/book_sle_deployment/data/pre_sle.html].

1.6 Additional Information

Each of the file system projects described above maintains its own home page on which to find mailing list information, further documentation, and FAQs:

- *E2fsprogs: Ext2/3/4 File System Utilities* [<http://e2fsprogs.sourceforge.net/>]
- *Introducing Ext3* [<http://www.ibm.com/developerworks/linux/library/l-fs7/>]
- *Using ReiserFS with Linux* [<http://www.ibm.com/developerworks/aix/library/au-unix-reiserFS/>]
- *XFS: A High-Performance Journaling Filesystem* [<http://oss.sgi.com/projects/xfst/>]
- *OCFS2 Project* [<http://oss.oracle.com/projects/ocfs2/>]

A comprehensive multipart tutorial about Linux file systems can be found at IBM developerWorks in the *Advanced File System Implementor's Guide* [<https://www.ibm.com/developerworks/linux/library/l-fs/>].

An in-depth comparison of file systems (not only Linux file systems) is available from the Wikipedia project in *Comparison of File Systems* [http://en.wikipedia.org/wiki/Comparison_of_file_systems#Comparison].

What's New for Storage in SLES 11

The features and behavior changes noted in this section were made for SUSE Linux Enterprise Server 11.

- Section 2.1, “What's New in SLES 11 SP2” (page 17)
- Section 2.2, “What's New in SLES 11 SP1” (page 18)
- Section 2.3, “What's New in SLES 11” (page 22)

2.1 What's New in SLES 11 SP2

In addition to bug fixes, the features and behavior changes in this section were made for the SUSE Linux Enterprise Server 11 SP2 release:

- BtrFS (Better File System). See Section 1.2.1, “BtrFS” (page 3).
- Open Fibre Channel over Ethernet. See Chapter 14, *Fibre Channel Storage over Ethernet Networks: FCoE* (page 233).
- Tagging for LVM storage objects. See Section 4.6, “Tagging LVM2 Storage Objects” (page 46).
- NFSv4 ACLs tools. See Chapter 16, *Managing Access Control Lists over NFSv4* (page 257).

- `--assume-clean` option for `mdadm resize` command. See Section 11.2.2, “Increasing the Size of the RAID Array” (page 182).

2.2 What’s New in SLES 11 SP1

In addition to bug fixes, the features and behavior changes noted in this section were made for the SUSE Linux Enterprise Server 11 SP1 release.

- Section 2.2.1, “Saving iSCSI Target Information” (page 18)
- Section 2.2.2, “Modifying Authentication Parameters in the iSCSI Initiator” (page 19)
- Section 2.2.3, “Allowing Persistent Reservations for MPIO Devices” (page 19)
- Section 2.2.4, “MDADM 3.0.2” (page 19)
- Section 2.2.5, “Boot Loader Support for MDRAID External Metadata” (page 20)
- Section 2.2.6, “YaST Install and Boot Support for MDRAID External Metadata” (page 20)
- Section 2.2.7, “Improved Shutdown for MDRAID Arrays that Contain the Root File System” (page 20)
- Section 2.2.8, “MD over iSCSI Devices” (page 21)
- Section 2.2.9, “MD-SGPIIO” (page 21)
- Section 2.2.10, “Resizing LVM 2 Mirrors ” (page 21)
- Section 2.2.11, “Updating Storage Drivers for Adapters on IBM Servers” (page 22)

2.2.1 Saving iSCSI Target Information

In the *YaSTNetwork Services* *iSCSI Target* function, a *Save* option was added that allows you to export the iSCSI target information. This makes it easier to provide information to consumers of the resources.

2.2.2 Modifying Authentication Parameters in the iSCSI Initiator

In the *YaSTNetwork ServicesiSCSI Initiator* function, you can modify the authentication parameters for connecting to a target devices. Previously, you needed to delete the entry and re-create it in order to change the authentication information.

2.2.3 Allowing Persistent Reservations for MPIO Devices

A SCSI initiator can issue SCSI reservations for a shared storage device, which locks out SCSI initiators on other servers from accessing the device. These reservations persist across SCSI resets that might happen as part of the SCSI exception handling process.

The following are possible scenarios where SCSI reservations would be useful:

- In a simple SAN environment, persistent SCSI reservations help protect against administrator errors where a LUN is attempted to be added to one server but it is already in use by another server, which might result in data corruption. SAN zoning is typically used to prevent this type of error.
- In a high-availability environment with failover set up, persistent SCSI reservations help protect against errant servers connecting to SCSI devices that are reserved by other servers.

2.2.4 MDADM 3.0.2

Use the latest version of the Multiple Devices Administration (MDADM, `mdadm`) utility to take advantage of bug fixes and improvements.

2.2.5 Boot Loader Support for MDRAID External Metadata

Support was added to use the external metadata capabilities of the MDADM utility version 3.0 to install and run the operating system from RAID volumes defined by the Intel Matrix Storage Technology metadata format. This moves the functionality from the Device Mapper RAID (DMRAID) infrastructure to the Multiple Devices RAID (MDRAID) infrastructure, which offers the more mature RAID 5 implementation and offers a wider feature set of the MD kernel infrastructure. It allows a common RAID driver to be used across all metadata formats, including Intel, DDF (common RAID disk data format), and native MD metadata.

2.2.6 YaST Install and Boot Support for MDRAID External Metadata

The YaST installer tool added support for MDRAID External Metadata for RAID 0, 1, 10, 5, and 6. The installer can detect RAID arrays and whether the platform RAID capabilities are enabled. If multipath RAID is enabled in the platform BIOS for Intel Matrix Storage Manager, it offers options for DMRAID, MDRAID (recommended), or none. The `initrd` was also modified to support assembling BIOS-based RAID arrays.

2.2.7 Improved Shutdown for MDRAID Arrays that Contain the Root File System

Shutdown scripts were modified to wait until all of the MDRAID arrays are marked clean. The operating system shutdown process now waits for a dirty-bit to be cleared until all MDRAID volumes have finished write operations.

Changes were made to the startup script, shutdown script, and the `initrd` to consider whether the root (/) file system (the system volume that contains the operating system and application files) resides on a software RAID array. The metadata handler for the array is started early in the shutdown process to monitor the final root file system envi-

ronment during the shutdown. The handler is excluded from the general `killall` events. The process also allows for writes to be quiesced and for the array's metadata dirty-bit (which indicates whether an array needs to be resynchronized) to be cleared at the end of the shutdown.

2.2.8 MD over iSCSI Devices

The YaST installer now allows MD to be configured over iSCSI devices.

If RAID arrays are needed on boot, the iSCSI initiator software is loaded before `boot.md` so that the iSCSI targets are available to be auto-configured for the RAID.

For a new install, Libstorage creates an `/etc/mdadm.conf` file and adds the line `AUTO -all`. During an update, the line is not added. If `/etc/mdadm.conf` contains the line

```
AUTO -all
```

then no RAID arrays are auto-assembled unless they are explicitly listed in `/etc/mdadm.conf`.

2.2.9 MD-SGPIO

The MD-SGPIO utility is a standalone application that monitors RAID arrays via `sysfs(2)`. Events trigger an LED change request that controls blinking for LED lights that are associated with each slot in an enclosure or a drive bay of a storage subsystem. It supports two types of LED systems:

- 2-LED systems (Activity LED, Status LED)
- 3-LED systems (Activity LED, Locate LED, Fail LED)

2.2.10 Resizing LVM 2 Mirrors

The `lvresize`, `lvextend`, and `lvreduce` commands that are used to resize logical volumes were modified to allow the resizing of LVM 2 mirrors. Previously, these commands reported errors if the logical volume was a mirror.

2.2.11 Updating Storage Drivers for Adapters on IBM Servers

Update the following storage drivers to use the latest available versions to support storage adapters on IBM servers:

- Adaptec: `aacraid`, `aic94xx`
- Emulex: `lpfc`
- LSI: `mptas`, `megaraid_sas`

The `mptsas` driver now supports native EEH (Enhanced Error Handler) recovery, which is a key feature for all of the IO devices for Power platform customers.

- qLogic: `qla2xxx`, `qla3xxx`, `qla4xxx`

2.3 What's New in SLES 11

The features and behavior changes noted in this section were made for the SUSE Linux Enterprise Server 11 release.

- Section 2.3.1, “EVMS2 Is Deprecated” (page 23)
- Section 2.3.2, “Ext3 as the Default File System” (page 23)
- Section 2.3.3, “JFS File System Is Deprecated” (page 23)
- Section 2.3.4, “OCFS2 File System Is in the High Availability Release” (page 24)
- Section 2.3.5, “/dev/disk/by-name Is Deprecated” (page 24)
- Section 2.3.6, “Device Name Persistence in the /dev/disk/by-id Directory” (page 24)
- Section 2.3.7, “Filters for Multipathed Devices” (page 25)
- Section 2.3.8, “User-Friendly Names for Multipathed Devices” (page 25)
- Section 2.3.9, “Advanced I/O Load-Balancing Options for Multipath” (page 26)

- Section 2.3.10, “Location Change for Multipath Tool Callouts” (page 26)
- Section 2.3.11, “Change from mpath to multipath for the mkinitrd -f Option” (page 26)
- Section 2.3.12, “Change from Multibus to Failover as the Default Setting for the MPIIO Path Grouping Policy” (page 27)

2.3.1 EVMS2 Is Deprecated

The Enterprise Volume Management Systems (EVMS2) storage management solution is deprecated. All EVMS management modules have been removed from the SUSE Linux Enterprise Server 11 packages. Your EVMS-managed devices should be automatically recognized and managed by Linux Volume Manager 2 (LVM2) when you upgrade your system. For more information, see *Evolution of Storage and Volume Management in SUSE Linux Enterprise* [<http://www.novell.com/linux/volumemanagement/strategy.html>].

For information about managing storage with EVMS2 on SUSE Linux Enterprise Server 10, see the *SUSE Linux Enterprise Server 10 SP3: Storage Administration Guide* [http://www.novell.com/documentation/sles10/stor_admin/data/bookinfo.html].

2.3.2 Ext3 as the Default File System

The Ext3 file system has replaced ReiserFS as the default file system recommended by the YaST tools at installation time and when you create file systems. ReiserFS is still supported. For more information, see *File System Support* [<http://www.novell.com/linux/techspecs.html?tab=2>] on the *SUSE Linux Enterprise 11 Tech Specs* Web page.

2.3.3 JFS File System Is Deprecated

The JFS file system is no longer supported. The JFS utilities were removed from the distribution.

2.3.4 OCFS2 File System Is in the High Availability Release

The OCFS2 file system is fully supported as part of the SUSE Linux Enterprise High Availability Extension.

2.3.5 `/dev/disk/by-name` Is Deprecated

The `/dev/disk/by-name` path is deprecated in SUSE Linux Enterprise Server 11 packages.

2.3.6 Device Name Persistence in the `/dev/disk/by-id` Directory

In SUSE Linux Enterprise Server 11, the default multipath setup relies on `udev` to overwrite the existing symbolic links in the `/dev/disk/by-id` directory when multipathing is started. Before you start multipathing, the link points to the SCSI device by using its `scsi-xxx` name. When multipathing is running, the symbolic link points to the device by using its `dm-uuid-xxx` name. This ensures that the symbolic links in the `/dev/disk/by-id` path persistently point to the same device regardless of whether multipathing is started or not. The configuration files (such as `lvm.conf` and `md.conf`) do not need to be modified because they automatically point to the correct device.

See the following sections for more information about how this behavior change affects other features:

- Section 2.3.7, “Filters for Multipathed Devices” (page 25)
- Section 2.3.8, “User-Friendly Names for Multipathed Devices” (page 25)

2.3.7 Filters for Multipathed Devices

The deprecation of the `/dev/disk/by-name` directory (as described in Section 2.3.5, “`/dev/disk/by-name` Is Deprecated” (page 24)) affects how you set up filters for multipathed devices in the configuration files. If you used the `/dev/disk/by-name` device name path for the multipath device filters in the `/etc/lvm/lvm.conf` file, you need to modify the file to use the `/dev/disk/by-id` path. Consider the following when setting up filters that use the `by-id` path:

- The `/dev/disk/by-id/scsi-*` device names are persistent and created for exactly this purpose.
- Do not use the `/dev/disk/by-id/dm-*` name in the filters. These are symbolic links to the Device-Mapper devices, and result in reporting duplicate PVs in response to a `pvscan` command. The names appear to change from `LVM-pvuuid` to `dm-uuid` and back to `LVM-pvuuid`.

For information about setting up filters, see Section 7.2.3, “Using LVM2 on Multipath Devices” (page 82).

2.3.8 User-Friendly Names for Multipathed Devices

A change in how multipathed device names are handled in the `/dev/disk/by-id` directory (as described in Section 2.3.6, “Device Name Persistence in the `/dev/disk/by-id` Directory” (page 24)) affects your setup for user-friendly names because the two names for the device differ. You must modify the configuration files to scan only the device mapper names after multipathing is configured.

For example, you need to modify the `lvm.conf` file to scan using the multipathed device names by specifying the `/dev/disk/by-id/dm-uuid-.*-mpath-.*` path instead of `/dev/disk/by-id`.

2.3.9 Advanced I/O Load-Balancing Options for Multipath

The following advanced I/O load-balancing options are available for Device Mapper Multipath, in addition to round-robin:

- Least-pending
- Length-load-balancing
- Service-time

For information, see ??? (page 122) in Section “Understanding Priority Groups and Attributes” (page 114).

2.3.10 Location Change for Multipath Tool Callouts

The `mpath_* prio_callouts` for the Device Mapper Multipath tool have been moved to shared libraries in `/lib/libmultipath/lib*`. By using shared libraries, the callouts are loaded into memory on daemon startup. This helps avoid a system deadlock on an all-paths-down scenario where the programs need to be loaded from the disk, which might not be available at this point.

2.3.11 Change from `mpath` to `multipath` for the `mkinitrd -f` Option

The option for adding Device Mapper Multipath services to the `initrd` has changed from `-f mpath` to `-f multipath`.

To make a new `initrd`, the command is now:

```
mkinitrd -f multipath
```

2.3.12 Change from Multibus to Failover as the Default Setting for the MPIO Path Grouping Policy

The default setting for the `path_grouping_policy` in the `/etc/multipath.conf` file has changed from `multibus` to `failover`.

For information about configuring the `path_grouping_policy`, see Section 7.6, “Configuring Path Failover Policies and Priorities” (page 112).

Planning a Storage Solution

Consider what your storage needs are and how you can effectively manage and divide your storage space to best meet your needs. Use the information in this section to help plan your storage deployment for file systems on your SUSE Linux Enterprise Server 11 server.

- Section 3.1, “Partitioning Devices” (page 29)
- Section 3.2, “Multipath Support” (page 29)
- Section 3.3, “Software RAID Support” (page 30)
- Section 3.4, “File System Snapshots” (page 30)
- Section 3.5, “Backup and Antivirus Support” (page 30)

3.1 Partitioning Devices

For information about using the YaST Expert Partitioner, see “Using the YaST Partitioner” in the *SUSE Linux Enterprise Server 11 Installation and Administration Guide*.

3.2 Multipath Support

Linux supports using multiple I/O paths for fault-tolerant connections between the server and its storage devices. Linux multipath support is disabled by default. If you

use a multipath solution that is provided by your storage subsystem vendor, you do not need to configure the Linux multipath separately.

3.3 Software RAID Support

Linux supports hardware and software RAID devices. If you use hardware RAID devices, software RAID devices are unnecessary. You can use both hardware and software RAID devices on the same server.

To maximize the performance benefits of software RAID devices, partitions used for the RAID should come from different physical devices. For software RAID 1 devices, the mirrored partitions cannot share any disks in common.

3.4 File System Snapshots

Linux supports file system snapshots.

3.5 Backup and Antivirus Support

- Section 3.5.1, “Open Source Backup” (page 30)
- Section 3.5.2, “Commercial Backup and Antivirus Support” (page 31)

3.5.1 Open Source Backup

Open source tools for backing up data on Linux include `tar`, `cpio`, and `rsync`. See the man pages for these tools for more information.

- PAX: POSIX File System Archiver. It supports `cpio` and `tar`, which are the two most common forms of standard archive (backup) files. See the man page for more information.
- Amanda: The Advanced Maryland Automatic Network Disk Archiver. See www.amanda.org [<http://www.amanda.org/>].

3.5.2 Commercial Backup and Antivirus Support

Novell Open Enterprise Server (OES) 2 for Linux is a product that includes SUSE Linux Enterprise Server (SLES) 10. Antivirus and backup software vendors who support OES 2 also support SLES 10. You can visit the vendor Web sites to find out about their scheduled support of SLES 11.

For a current list of possible backup and antivirus software vendors, see *Novell Open Enterprise Server Partner Support: Backup and Antivirus Support* [http://www.novell.com/products/openenterpriseserver/partners_communities.html]. This list is updated quarterly.

LVM Configuration

This section briefly describes the principles behind Logical Volume Manager (LVM) and its basic features that make it useful under many circumstances. The YaST LVM configuration can be reached from the YaST Expert Partitioner. This partitioning tool enables you to edit and delete existing partitions and create new ones that should be used with LVM.

WARNING

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

- Section 4.1, “Understanding the Logical Volume Manager” (page 34)
- Section 4.2, “Creating LVM Partitions” (page 37)
- Section 4.3, “Creating Volume Groups” (page 39)
- Section 4.4, “Configuring Physical Volumes” (page 41)
- Section 4.5, “Configuring Logical Volumes” (page 43)
- Section 4.6, “Tagging LVM2 Storage Objects” (page 46)
- Section 4.7, “Resizing a Volume Group” (page 55)

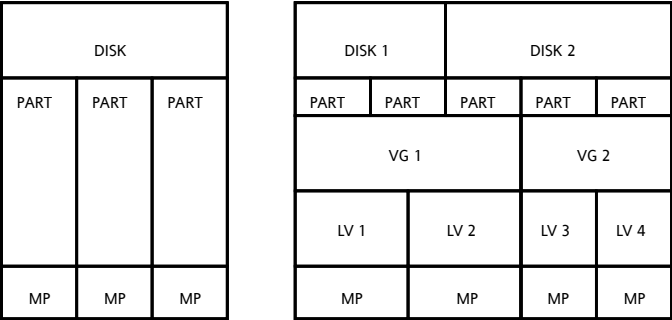
- Section 4.8, “Resizing a Logical Volume with YaST” (page 57)
- Section 4.9, “Resizing a Logical Volume with Commands” (page 58)
- Section 4.10, “Deleting a Volume Group” (page 59)
- Section 4.11, “Deleting an LVM Partition (Physical Volume)” (page 60)

4.1 Understanding the Logical Volume Manager

LVM enables flexible distribution of hard disk space over several file systems. It was developed because the need to change the segmentation of hard disk space might arise only after the initial partitioning has already been done during installation. Because it is difficult to modify partitions on a running system, LVM provides a virtual pool (volume group or VG) of memory space from which logical volumes (LVs) can be created as needed. The operating system accesses these LVs instead of the physical partitions. Volume groups can span more than one disk, so that several disks or parts of them can constitute one single VG. In this way, LVM provides a kind of abstraction from the physical disk space that allows its segmentation to be changed in a much easier and safer way than through physical repartitioning.

Figure 4.1, “Physical Partitioning versus LVM” (page 35) compares physical partitioning (left) with LVM segmentation (right). On the left side, one single disk has been divided into three physical partitions (PART), each with a mount point (MP) assigned so that the operating system can access them. On the right side, two disks have been divided into two and three physical partitions each. Two LVM volume groups (VG 1 and VG 2) have been defined. VG 1 contains two partitions from DISK 1 and one from DISK 2. VG 2 contains the remaining two partitions from DISK 2.

Figure 4.1 *Physical Partitioning versus LVM*



In LVM, the physical disk partitions that are incorporated in a volume group are called physical volumes (PVs). Within the volume groups in Figure 4.1, “Physical Partitioning versus LVM” (page 35), four logical volumes (LV 1 through LV 4) have been defined, which can be used by the operating system via the associated mount points. The border between different logical volumes need not be aligned with any partition border. See the border between LV 1 and LV 2 in this example.

LVM features:

- Several hard disks or partitions can be combined in a large logical volume.
- Provided the configuration is suitable, an LV (such as `/usr`) can be enlarged when the free space is exhausted.
- Using LVM, it is possible to add hard disks or LVs in a running system. However, this requires hot-swappable hardware that is capable of such actions.
- It is possible to activate a *striping mode* that distributes the data stream of a logical volume over several physical volumes. If these physical volumes reside on different disks, this can improve the reading and writing performance just like RAID 0.
- The snapshot feature enables consistent backups (especially for servers) in the running system.

With these features, using LVM already makes sense for heavily used home PCs or small servers. If you have a growing data stock, as in the case of databases, music archives, or user directories, LVM is especially useful. It allows file systems that are larger than the physical hard disk. Another advantage of LVM is that up to 256 LVs can be added. However, keep in mind that working with LVM is different from working with conventional partitions.

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

You can manage new or existing LVM storage objects by using the YaST Partitioner. Instructions and further information about configuring LVM is available in the official *LVM HOWTO* [<http://tldp.org/HOWTO/LVM-HOWTO/>].

4.2 Creating LVM Partitions

For each disk, partition the free space that you want to use for LVM as `0x8E Linux LVM`. You can create one or multiple LVM partitions on a single device. It is not necessary for all of the partitions on a device to be LVM partitions.

You can use the Volume Group function to group one or more LVM partitions into a logical pool of space called a volume group, then carve out one or more logical volumes from the space in the volume group.

In the YaST Partitioner, only the free space on the disk is made available to you as you are creating LVM partitions. If you want to use the entire disk for a single LVM partition and other partitions already exists on the disk, you must first remove all of the existing partitions to free the space before you can use that space in an LVM partition.

WARNING

Deleting a partition destroys all of the data in the partition.

- 1 Log in as the `root` user, then open YaST.
- 2 In YaST, open the *Partitioner*.
- 3 (Optional) Remove one or more existing partitions to free that space and make it available for the LVM partition you want to create.

For information, see Section 4.11, “Deleting an LVM Partition (Physical Volume)” (page 60).

- 4 On the Partitions page, click *Add*.
- 5 Under *New Partition Type*, select *Primary Partition* or *Extended Partition*, then click *Next*.
- 6 Specify the *New Partition Size*, then click *Next*.

New Partition Size

☐ Maximum Size (305.92 MB)

☒ Custom Size

Size (e.g. 9.0 MB or 9.0 GB)

250 MB

☐ Custom Region

Start Cylinder

0

End Cylinder

38

- **Maximum Size:** Use all of the free available space on the disk.
- **Custom Size:** Specify a size up the amount of free available space on the disk.
- **Custom Region:** Specify the start and end cylinder of the free available space on the disk.

7 Configure the partition format:

Formatting Options

☐ Format partition

File system

Ext3

Options...

☒ Do not format partition

File system ID:

0x8E Linux LVM

☐ Encrypt device

1. Under *Formatting Options*, select *Do not format*.
2. From the *File System ID* drop-down list, select *0x8E Linux LVM* as the partition identifier.
3. Under *Mounting Options*, select *Do not mount partition*.

8 Click *Finish*.

The partitions are not actually created until you click *Next* and *Finish* to exit the partitioner.

- 9** Repeat Step 4 (page 37) through Step 8 (page 38) for each Linux LVM partition you want to add.

- 10 Click *Next*, verify that the new Linux LVM partitions are listed, then click *Finish* to exit the partitioner.
- 11 (Optional) Continue with the Volume Group configuration as described in Section 4.3, “Creating Volume Groups” (page 39).

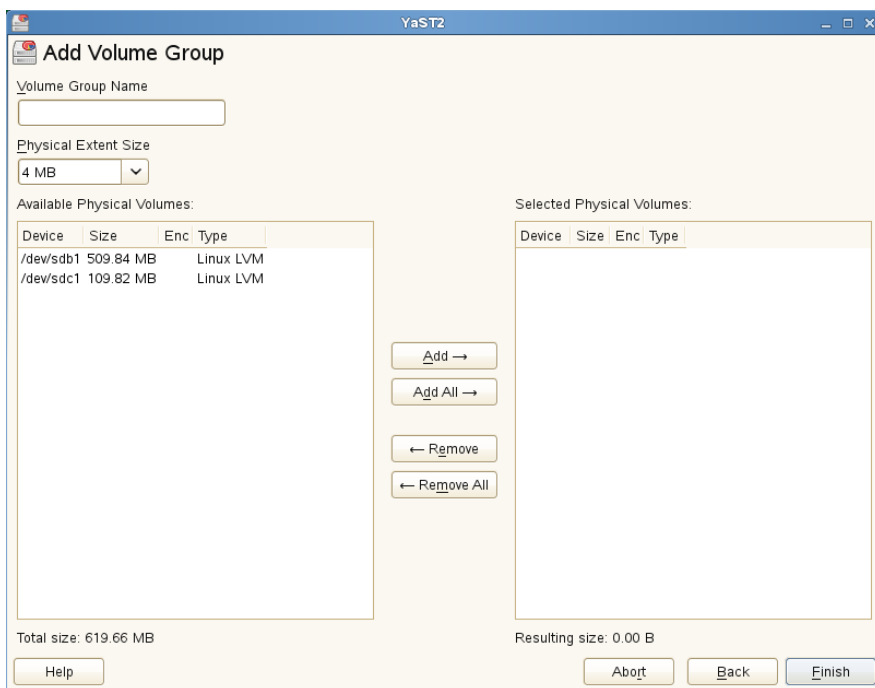
4.3 Creating Volume Groups

An LVM volume group organizes the Linux LVM partitions into a logical pool of space. You can carve out logical volumes from the available space in the group. The Linux LVM partitions in a group can be on the same or different disks. You can add LVM partitions from the same or different disks to expand the size of the group. Assign all partitions reserved for LVM to a volume group. Otherwise, the space on the partition remains unused.

- 1 Log in as the `root` user, then open YaST.
- 2 In YaST, open the *Partitioner*.
- 3 In the left panel, click *Volume Management*.

A list of existing Volume Groups are listed in the right panel.

- 4 At the lower left of the Volume Management page, click *Add Volume Group*.



5 Specify the *Volume Group Name*.

If you are creating a volume group at install time, the name `system` is suggested for a volume group that will contain the SUSE Linux Enterprise Server system files.

6 Specify the *Physical Extent Size*.

The *Physical Extent Size* defines the size of a physical block in the volume group. All the disk space in a volume group is handled in chunks of this size. Values can be from 1 KB to 16 GB in powers of 2. This value is normally set to 4 MB.

In LVM1, a 4 MB physical extent allowed a maximum LV size of 256 GB because it supports only up to 65534 extents per LV. VM2 does not restrict the number of physical extents. Having a large number of extents has no impact on I/O performance to the logical volume, but it slows down the LVM tools.

IMPORTANT

Different physical extent sizes should not be mixed in a single VG. The extent should not be modified after the initial setup.

- 7 In the *Available Physical Volumes* list, select the Linux LVM partitions that you want to make part of this volume group, then click *Add* to move them to the *Selected Physical Volumes* list.
- 8 Click *Finish*.

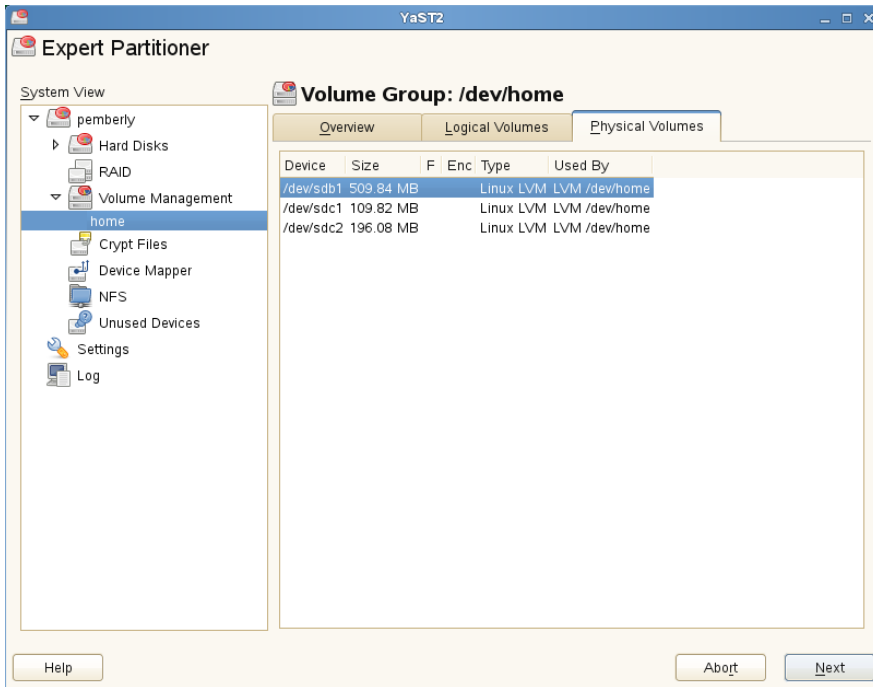
The new group appears in the *Volume Groups* list.

- 9 On the Volume Management page, click *Next*, verify that the new volume group is listed, then click *Finish*.

4.4 Configuring Physical Volumes

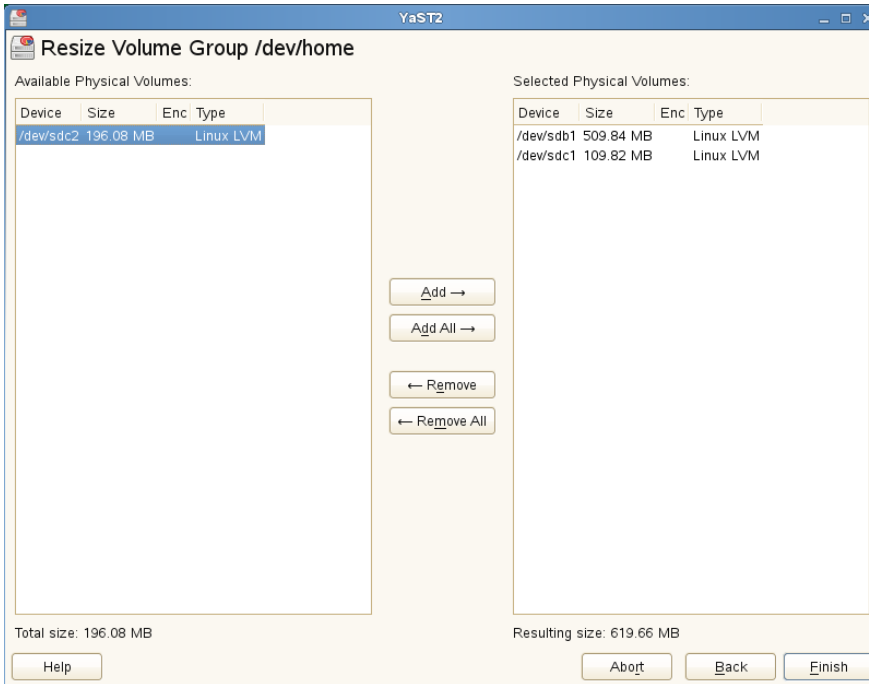
When the Linux LVM partitions are assigned to a volume group, the partitions are then referred to as physical volumes.

Figure 4.2 *Physical Volumes in the Volume Group Named Home*



To add more physical volumes to an existing volume group:

- 1 Log in as the `root` user, then open YaST.
- 2 In YaST, open the *Partitioner*.
- 3 In the left panel, select Volume Management and expand the list of groups.
- 4 Under Volume Management, select the volume group, then click the *Overview* tab.
- 5 At the bottom of the page, click *Resize*.

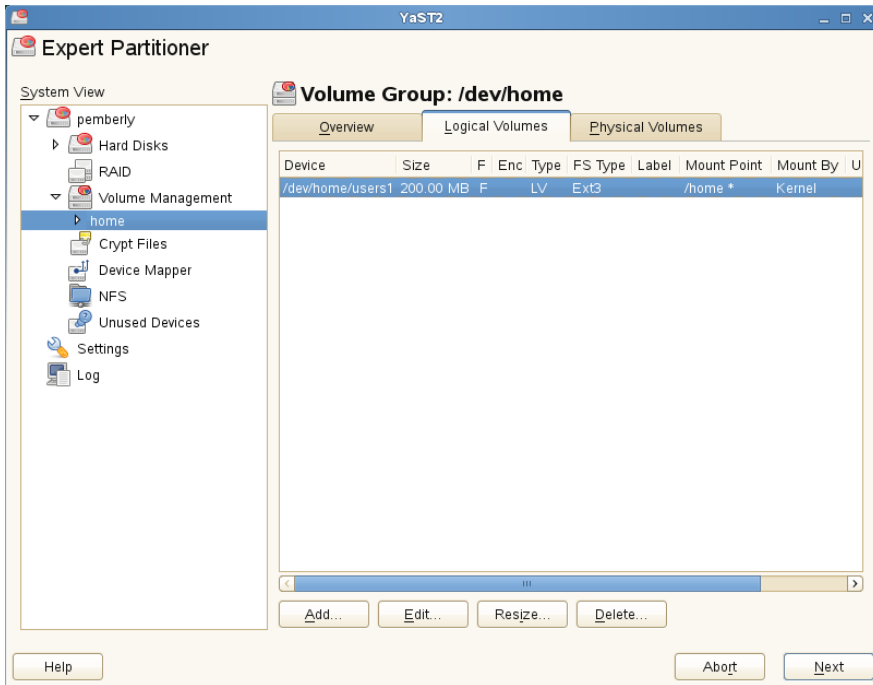


- 6 Select a physical volume (LVM partitions) from the *Available Physical Volumes* list then click *Add* to move it to the *Selected Physical Volumes* list.
- 7 Click *Finish*.
- 8 Click *Next*, verify that the changes are listed, then click *Finish*.

4.5 Configuring Logical Volumes

After a volume group has been filled with physical volumes, use the Logical Volumes dialog (see Figure 4.3, “Logical Volume Management” (page 44)) to define and manage the logical volumes that the operating system should use. This dialog lists all of the logical volumes in that volume group. You can use *Add*, *Edit*, and *Remove* options to manage the logical volumes. Assign at least one logical volume to each volume group. You can create new logical volumes as needed until all free space in the volume group has been exhausted.

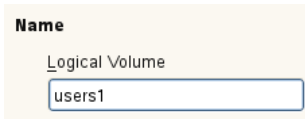
Figure 4.3 *Logical Volume Management*



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

- 1 Log in as the `root` user, then open YaST.
- 2 In YaST, open the *Partitioner*.
- 3 In the left panel, select Volume Management and expand it to see the list of volume groups.
- 4 Under Volume Management, select the volume group, then click the *Logical Volumes* tab.

- 5 In the lower left, click *Add* to open the *Add Logical Volume* dialog.
- 6 Specify the *Name* for the logical volume, then click *Next*.

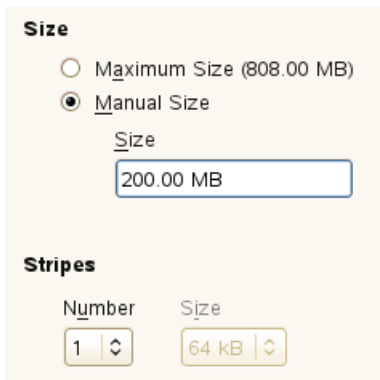


Name

Logical Volume

users1

- 7 Specify the size of the volume and whether to use multiple stripes.



Size

☐ Maximum Size (808.00 MB)

☒ Manual Size

Size

200.00 MB

Stripes

Number	Size
1	64 kB

1. Specify the size of the logical volume, up to the maximum size available.

The amount of free space in the current volume group is shown next to the *Maximum Size* option.

2. Specify the number of stripes.

WARNING

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

- 8 Specify the formatting options for the logical volume:

Formatting Options

☒ **Format partition**

File system

Ext3

Options...

☐ **Do not format partition**

☐ **Encrypt device**

Mounting Options

☒ **Mount partition**

Mount Point

/home

Fstab Options...

☐ **Do not mount partition**

1. Under *Formatting Options*, select *Format partition*, then select the format type from the *File system* drop-down list, such as Ext3.

2. Under *Mounting Options*, select *Mount partition*, then select the mount point.

The files stored on this logical volume can be found at this mount point on the installed system.

3. Click *Fstab Options* to add special mounting options for the volume.

9 Click *Finish*.

10 Click *Next*, verify that the changes are listed, then click *Finish*.

4.6 Tagging LVM2 Storage Objects

A tag is an unordered keyword or term assigned to the metadata of a storage object. Tagging allows you to classify collections of LVM storage objects in ways that you find useful by attaching an unordered list of tags to their metadata.

- Section 4.6.1, “Using LVM2 Tags” (page 47)
- Section 4.6.2, “Requirements for Creating LVM2 Tags” (page 47)
- Section 4.6.3, “Command Line Tag Syntax” (page 48)

- Section 4.6.4, “Configuration File Syntax” (page 49)
- Section 4.6.5, “Using Tags for a Simple Activation Control in a Cluster” (page 51)
- Section 4.6.6, “Using Tags to Activate On Preferred Hosts in a Cluster” (page 51)

4.6.1 Using LVM2 Tags

After you tag the LVM2 storage objects, you can use the tags in commands to accomplish the following tasks:

- Select LVM objects for processing according to the presence or absence of specific tags.
- Use tags in the configuration file to control which volume groups and logical volumes are activated on a server.
- Override settings in a global configuration file by specifying tags in the command.

A tag can be used in place of any command line LVM object reference that accepts:

- a list of objects
- a single object as long as the tag expands to a single object

Replacing the object name with a tag is not supported everywhere yet. After the arguments are expanded, duplicate arguments in a list are resolved by removing the duplicate arguments, and retaining the first instance of each argument.

Wherever there might be ambiguity of argument type, you must prefix a tag with the commercial at sign (@) character, such as @mytag. Elsewhere, using the “@” prefix is optional.

4.6.2 Requirements for Creating LVM2 Tags

Consider the following requirements when using tags with LVM:

- Section “Supported Characters” (page 48)

- Section “Supported Storage Objects” (page 48)

Supported Characters

An LVM tag word can contain the ASCII uppercase characters A to Z, lowercase characters a to z, numbers 0 to 9, underscore (_), plus (+), hyphen (-), and period (.). The word cannot begin with a hyphen. The maximum length is 128 characters.

Supported Storage Objects

You can tag LVM2 physical volumes, volume groups, logical volumes, and logical volume segments. PV tags are stored in its volume group’s metadata. Deleting a volume group also deletes the tags in the orphaned physical volume. Snapshots cannot be tagged, but their origin can be tagged.

LVM1 objects cannot be tagged because the disk format does not support it.

4.6.3 Command Line Tag Syntax

`--addtag <tag_info>`

Add a tag to (or *tag*) an LVM2 storage object.

Example

```
vgchange --addtag @dbl vg1
```

`--deltag <tag_info>`

Remove a tag from (or *untag*) an LVM2 storage object.

Example

```
vgchange --deltag @dbl vg1
```

`--tag <tag_info>`

Specify the tag to use to narrow the list of volume groups or logical volumes to be activated or deactivated.

Example

Enter the following to activate it the volume if it has a tag that matches the tag provided:

```
lvchange -ay --tag @db1 vg1/vol2
```

4.6.4 Configuration File Syntax

- Section “Enabling Hostname Tags in the lvm.conf File” (page 49)
- Section “Defining Tags for Hostnames in the lvm.conf File” (page 49)
- Section “Defining Activation” (page 50)
- Section “Defining Activation in Multiple Hostname Configuration Files” (page 50)

Enabling Hostname Tags in the lvm.conf File

Add the following code to the `/etc/lvm/lvm.conf` file to enable host tags that are defined separately on host in a `/etc/lvm/lvm_<hostname>.conf` file.

```
tags {  
    # Enable hostname tags  
    hosttags = 1  
}
```

You place the activation code in the `/etc/lvm/lvm_<hostname>.conf` file on the host. See Section “Defining Activation” (page 50).

Defining Tags for Hostnames in the lvm.conf File

```
tags {  
  
    tag1 { }  
        # Tag does not require a match to be set.  
  
    tag2 {  
        # If no exact match, tag is not set.  
        host_list = [ "hostname1", "hostname2" ]  
    }  
}
```

Defining Activation

You can modify the `/etc/lvm/lvm.conf` file to activate LVM logical volumes based on tags.

In a text editor, add the following code to the file:

```
activation {
    volume_list = [ "vg1/lvol0", "@database" ]
}
```

Replace `@database` with the your tag. Use `"@"` to match the tag against any tag set on the host.

The activation command matches against *vgname*, *vgname/lvname*, or *@tag* set in the metadata of volume groups and logical volumes. A volume group or logical volume is activated only if a metadata tag matches. The default if there is no match is not to activate.

If `volume_list` is not present and any tags are defined on the host, then it activates the volume group or logical volumes only if a host tag matches a metadata tag.

If `volume_list` is not present and no tags are defined on the host, then it does activate.

Defining Activation in Multiple Hostname Configuration Files

You can use the activation code in a host's configuration file (`/etc/lvm/lvm_<host_tag>.conf`) when host tags are enabled in the `lvm.conf` file. For example, a server has two configuration files in the `/etc/lvm/` folder:

```
lvm.conf
lvm_<host_tag>.conf
```

At startup, load the `/etc/lvm/lvm.conf` file, and process any tag settings in the file. If any host tags were defined, it loads the related `/etc/lvm/lvm_<host_tag>.conf` file. When it searches for a specific configuration file entry, it searches the host tag file first, then the `lvm.conf` file, and stops at the first match. Within the `lvm_<host_tag>.conf` file, use the reverse order that tags were set. This allows the

file for the last tag set to be searched first. New tags set in the host tag file will trigger additional configuration file loads.

4.6.5 Using Tags for a Simple Activation Control in a Cluster

You can set up a simple hostname activation control by enabling the `hostname_tags` option in a the `/etc/lvm/lvm.conf` file. Use the same file on every machine in a cluster so that it is a global setting.

- 1 In a text editor, add the following code to the `/etc/lvm/lvm.conf` file:

```
tags {
    hostname_tags = 1
}
```

- 2 Replicate the file to all hosts in the cluster.
- 3 From any machine in the cluster, add `db1` to the list of machines that activate `vg1/lvol2`:

```
lvchange --addtag @db1 vg1/lvol2
```

- 4 On the `db1` server, enter the following to activate it:

```
lvchange -ay vg1/vol2
```

4.6.6 Using Tags to Activate On Preferred Hosts in a Cluster

The examples in this section demonstrate two methods to accomplish the following:

- Activate volume group `vg1` only on the database hosts `db1` and `db2`.
- Activate volume group `vg2` only on the file server host `fs1`.

- Activate nothing initially on the file server backup host `fsb1`, but be prepared for it to take over from the file server host `fs1`.
- Section “Option 1: Centralized Admin and Static Configuration Replicated Between Hosts” (page 52)
- Section “Option 2: Localized Admin and Configuration” (page 53)

Option 1: Centralized Admin and Static Configuration Replicated Between Hosts

In the following solution, the single configuration file is replicated among multiple hosts.

- 1 Add the `@database` tag to the metadata of volume group `vg1`. In a terminal console, enter

```
vgchange --addtag @database vg1
```

- 2 Add the `@fileserver` tag to the metadata of volume group `vg2`. In a terminal console, enter

```
vgchange --addtag @fileserver vg2
```

- 3 In a text editor, modify the `/etc/lvm/lvm.conf` file with the following code to define the `@database`, `@fileserver`, `@fileserverbackup` tags.

```
tags {
    database {
        host_list = [ "db1", "db2" ]
    }
    fileserver {
        host_list = [ "fs1" ]
    }
    fileserverbackup {
        host_list = [ "fsb1" ]
    }
}

activation {
    # Activate only if host has a tag that matches a metadata tag
```

```
volume_list = [ "*" ]
}
```

- 4** Replicate the modified `/etc/lvm/lvm.conf` file to the four hosts: `db1`, `db2`, `fs1`, and `fsb1`.
- 5** If the file server host goes down, `vg2` can be brought up on `fsb1` by entering the following commands in a terminal console on any node:

```
vgchange --addtag @fileserverbackup vg2
vgchange -ay vg2
```

Option 2: Localized Admin and Configuration

In the following solution, each host holds locally the information about which classes of volume to activate.

- 1** Add the `@database` tag to the metadata of volume group `vg1`. In a terminal console, enter

```
vgchange --addtag @database vg1
```

- 2** Add the `@fileserver` tag to the metadata of volume group `vg2`. In a terminal console, enter

```
vgchange --addtag @fileserver vg2
```

- 3** Enable host tags in the `/etc/lvm/lvm.conf` file:

- 3a** In a text editor, modify the `/etc/lvm/lvm.conf` file with the following code to enable host tag configuration files.

```
tags {
    hosttags = 1
}
```

- 3b** Replicate the modified `/etc/lvm/lvm.conf` file to the four hosts: `db1`, `db2`, `fs1`, and `fsb1`.

- 4** On host `db1`, create an activation configuration file for the database host `db1`. In a text editor, create a `/etc/lvm/lvm_db1.conf` file and add the following code:

```
activation {
    volume_list = [ "@database" ]
}
```

- 5** On host `db2`, create an activation configuration file for the database host `db2`. In a text editor, create a `/etc/lvm/lvm_db2.conf` file and add the following code:

```
activation {
    volume_list = [ "@database" ]
}
```

- 6** On host `fs1`, create an activation configuration file for the file server host `fs1`. In a text editor, create a `/etc/lvm/lvm_fs1.conf` file and add the following code:

```
activation {
    volume_list = [ "@fileserver" ]
}
```

- 7** If the file server host `fs1` goes down, to bring up a spare file server host `fsb1` as a file server:

- 7a** On host `fsb1`, create an activation configuration file for the host `fsb1`. In a text editor, create a `/etc/lvm/lvm_fsb1.conf` file and add the following code:

```
activation {
    volume_list = [ "@fileserver" ]
}
```

- 7b** In a terminal console, enter one of the following commands:

```
vgchange -ay vg2
```

```
vgchange -ay @fileserver
```

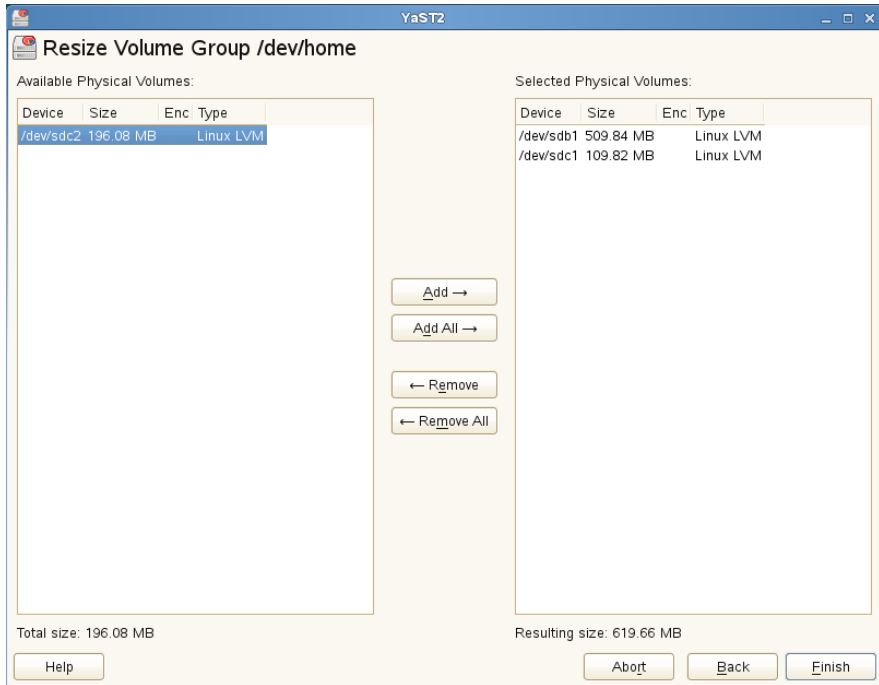

4.7 Resizing a Volume Group

You can add and remove Linux LVM partitions from a volume group to expand or reduce its size.

WARNING

Removing a partition can result in data loss if the partition is in use by a logical volume.

- 1 Log in as the `root` user, then open YaST.
- 2 In YaST, open the *Partitioner*.
- 3 In the left panel, select Volume Management and expand it to see the list of volume groups.
- 4 Under Volume Management, select the volume group, then click the *Overview* tab.
- 5 At the bottom of the page, click *Resize*.



6 Do one of the following:

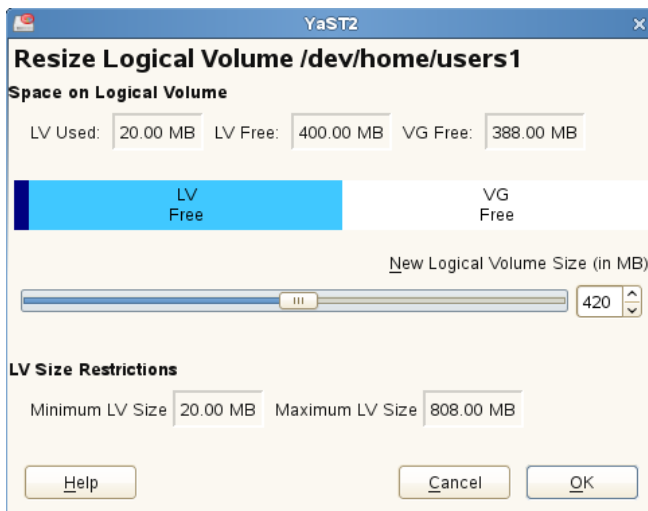
- **Add:** Expand the size of the volume group by moving one or more physical volumes (LVM partitions) from the *Available Physical Volumes* list to the *Selected Physical Volumes* list.
- **Remove:** Reduce the size of the volume group by moving one or more physical volumes (LVM partitions) from the *Selected Physical Volumes* list to the *Available Physical Volumes* list.

7 Click *Finish*.

8 Click *Next*, verify that the changes are listed, then click *Finish*.

4.8 Resizing a Logical Volume with YaST

- 1 Log in as the `root` user, then open YaST.
- 2 In YaST, open the *Partitioner*.
- 3 In the left panel, select Volume Management and expand it to see the list of volume groups.
- 4 Under Volume Management, select the volume group, then click the *Logical Volumes* tab.
- 5 At the bottom of the page, click *Resize* to open the *Resize Logical Volume* dialog.



- 6 Use the slider to expand or reduce the size of the logical volume.

WARNING

Reducing the size of a logical volume that contains data can cause data corruption.

7 Click *OK*.

8 Click *Next*, verify that the change is listed, then click *Finish*.

4.9 Resizing a Logical Volume with Commands

The `lvresize`, `lvextend`, and `lvreduce` commands are used to resize logical volumes. See the man pages for each of these commands for syntax and options information.

You can also increase the size of a logical volume by using the YaST Partitioner. YaST uses `parted(8)` to grow the partition.

To extend an LV there must be enough unallocated space available on the VG.

LVs can be extended or shrunk while they are being used, but this may not be true for a file system on them. Extending or shrinking the LV does not automatically modify the size of file systems in the volume. You must use a different command to grow the file system afterwards. For information about resizing file systems, see Chapter 5, *Resizing File Systems* (page 61).

Ensure that you use the right sequence:

- If you extend an LV, you must extend the LV before you attempt to grow the file system.
- If you shrink an LV, you must shrink the file system before you attempt to shrink the LV.

To extend the size of a logical volume:

- 1 Open a terminal console, log in as the `root` user.
- 2 If the logical volume contains file systems that are hosted for a virtual machine (such as a Xen VM), shut down the VM.
- 3 Dismount the file systems on the logical volume.

- 4 At the terminal console prompt, enter the following command to grow the size of the logical volume:

```
lvextend -L +size /dev/vgname/lvname
```

For *size*, specify the amount of space you want to add to the logical volume, such as 10GB. Replace */dev/vgname/lvname* with the Linux path to the logical volume, such as */dev/vg1/v1*. For example:

```
lvextend -L +10GB /dev/vg1/v1
```

For example, to extend an LV with a (mounted and active) ReiserFS on it by 10GB:

```
lvextend -L +10G /dev/vgname/lvname  
resize_reiserfs -s +10GB -f /dev/vg-name/lv-name
```

For example, to shrink an LV with a ReiserFS on it by 5GB:

```
umount /mountpoint-of-LV  
resize_reiserfs -s -5GB /dev/vgname/lvname  
lvreduce /dev/vgname/lvname  
mount /dev/vgname/lvname /mountpoint-of-LV
```

4.10 Deleting a Volume Group

WARNING

Deleting a volume group destroys all of the data in each of its member partitions.

- 1 Log in as the `root` user, then open YaST.
- 2 In YaST, open the *Partitioner*.
- 3 In the left panel, select Volume Management and expand the list of groups.
- 4 Under Volume Management, select the volume group, then click the *Overview* tab.
- 5 At the bottom of the page, click *Delete*, then click *Yes* to confirm the deletion.

- 6 Click *Next*, verify that the deleted volume group is listed (deletion is indicated by a red colored font), then click *Finish*.

4.11 Deleting an LVM Partition (Physical Volume)

WARNING

Deleting a partition destroys all of the data in the partition.

- 1 Log in as the `root` user, then open YaST.
- 2 In YaST, open the *Partitioner*.
- 3 If the Linux LVM partition is in use as a member of a volume group, remove the partition from the volume group, or delete the volume group.
- 4 In the YaST Partitioner under *Hard Disks*, select the device (such as `sdc`).
- 5 On the Partitions page, select a partition that you want to remove, click *Delete*, then click *Yes* to confirm the deletion.
- 6 Click *Next*, verify that the deleted partition is listed (deletion is indicated by a red colored font), then click *Finish*.

Resizing File Systems

When your data needs grow for a volume, you might need to increase the amount of space allocated to its file system.

- Section 5.1, “Guidelines for Resizing” (page 61)
- Section 5.2, “Increasing the Size of an Ext2, Ext3, or Ext4 File System” (page 63)
- Section 5.3, “Increasing the Size of a Reiser File System” (page 64)
- Section 5.4, “Decreasing the Size of an Ext2 or Ext3 File System” (page 66)
- Section 5.5, “Decreasing the Size of a Reiser File System” (page 67)

5.1 Guidelines for Resizing

Resizing any partition or file system involves some risks that can potentially result in losing data.

WARNING

To avoid data loss, ensure that you back up your data before you begin any resizing task.

Consider the following guidelines when planning to resize a file system.

- Section 5.1.1, “File Systems that Support Resizing” (page 62)

- Section 5.1.2, “Increasing the Size of a File System” (page 62)
- Section 5.1.3, “Decreasing the Size of a File System” (page 63)

5.1.1 File Systems that Support Resizing

The file system must support resizing in order to take advantage of increases in available space for the volume. In SUSE Linux Enterprise Server 11, file system resizing utilities are available for file systems Ext2, Ext3, Ext4, and ReiserFS. The utilities support increasing and decreasing the size as follows:

Table 5.1 *File System Support for Resizing*

File System	Utility	Increase Size (Grow)	Decrease Size (Shrink)
Ext2	resize2fs	Offline only	Offline only
Ext3	resize2fs	Online or offline	Offline only
Ext4	resize2fs	Offline only	Offline only
ReiserFS	resize_reiserfs	Online or offline	Offline only

5.1.2 Increasing the Size of a File System

You can grow a file system to the maximum space available on the device, or specify an exact size. Ensure that you grow the size of the device or logical volume before you attempt to increase the size of the file system.

When specifying an exact size for the file system, ensure that the new size satisfies the following conditions:

- The new size must be greater than the size of the existing data; otherwise, data loss occurs.
- The new size must be equal to or less than the current device size because the file system size cannot extend beyond the space available.

5.1.3 Decreasing the Size of a File System

When decreasing the size of the file system on a device, ensure that the new size satisfies the following conditions:

- The new size must be greater than the size of the existing data; otherwise, data loss occurs.
- The new size must be equal to or less than the current device size because the file system size cannot extend beyond the space available.

If you plan to also decrease the size of the logical volume that holds the file system, ensure that you decrease the size of the file system before you attempt to decrease the size of the device or logical volume.

5.2 Increasing the Size of an Ext2, Ext3, or Ext4 File System

The size of Ext2, Ext3, and Ext4 file systems can be increased by using the `resize2fs` command when the file system is mounted. The size of an Ext3 file system can also be increased by using the `resize2fs` command when the file system is unmounted.

- 1 Open a terminal console, then log in as the `root` user or equivalent.
- 2 If the file system is Ext2 or Ext4, you must unmount the file system. The Ext3 file system can be mounted or unmounted.
- 3 Increase the size of the file system using one of the following methods:
 - To extend the file system size to the maximum available size of the device called `/dev/sda1`, enter

```
resize2fs /dev/sda1
```

If a size parameter is not specified, the size defaults to the size of the partition.

- To extend the file system to a specific size, enter

```
resize2fs /dev/sda1 size
```

The *size* parameter specifies the requested new size of the file system. If no units are specified, the unit of the size parameter is the block size of the file system. Optionally, the size parameter can be suffixed by one of the following the unit designators: s for 512 byte sectors; K for kilobytes (1 kilobyte is 1024 bytes); M for megabytes; or G for gigabytes.

Wait until the resizing is completed before continuing.

- 4 If the file system is not mounted, mount it now.

For example, to mount an Ext2 file system for a device named `/dev/sda1` at mount point `/home`, enter

```
mount -t ext2 /dev/sda1 /home
```

- 5 Check the effect of the resize on the mounted file system by entering

```
df -h
```

The Disk Free (`df`) command shows the total size of the disk, the number of blocks used, and the number of blocks available on the file system. The `-h` option print sizes in human-readable format, such as 1K, 234M, or 2G.

5.3 Increasing the Size of a Reiser File System

A ReiserFS file system can be increased in size while mounted or unmounted.

- 1 Open a terminal console, then log in as the `root` user or equivalent.
- 2 Increase the size of the file system on the device called `/dev/sda2`, using one of the following methods:
 - To extend the file system size to the maximum available size of the device, enter

```
resize_reiserfs /dev/sda2
```

When no size is specified, this increases the volume to the full size of the partition.

- To extend the file system to a specific size, enter

```
resize_reiserfs -s size /dev/sda2
```

Replace *size* with the desired size in bytes. You can also specify units on the value, such as 50000K (kilobytes), 250M (megabytes), or 2G (gigabytes). Alternatively, you can specify an increase to the current size by prefixing the value with a plus (+) sign. For example, the following command increases the size of the file system on `/dev/sda2` by 500 MB:

```
resize_reiserfs -s +500M /dev/sda2
```

Wait until the resizing is completed before continuing.

- 3** If the file system is not mounted, mount it now.

For example, to mount an ReiserFS file system for device `/dev/sda2` at mount point `/home`, enter

```
mount -t reiserfs /dev/sda2 /home
```

- 4** Check the effect of the resize on the mounted file system by entering

```
df -h
```

The Disk Free (`df`) command shows the total size of the disk, the number of blocks used, and the number of blocks available on the file system. The `-h` option print sizes in human-readable format, such as 1K, 234M, or 2G.

5.4 Decreasing the Size of an Ext2 or Ext3 File System

You can shrink the size of the Ext2, Ext3, or Ext4 file systems when the volume is unmounted.

- 1 Open a terminal console, then log in as the `root` user or equivalent.
- 2 Unmount the file system.
- 3 Decrease the size of the file system on the device such as `/dev/sda1` by entering

```
resize2fs /dev/sda1 <size>
```

Replace *size* with an integer value in kilobytes for the desired size. (A kilobyte is 1024 bytes.)

Wait until the resizing is completed before continuing.

- 4 Mount the file system. For example, to mount an Ext2 file system for a device named `/dev/sda1` at mount point `/home`, enter

```
mount -t ext2 /dev/md0 /home
```

- 5 Check the effect of the resize on the mounted file system by entering

```
df -h
```

The Disk Free (`df`) command shows the total size of the disk, the number of blocks used, and the number of blocks available on the file system. The `-h` option print sizes in human-readable format, such as 1K, 234M, or 2G.

5.5 Decreasing the Size of a Reiser File System

Reiser file systems can be reduced in size only if the volume is unmounted.

- 1 Open a terminal console, then log in as the `root` user or equivalent.
- 2 Unmount the device by entering

```
umount /mnt/point
```

If the partition you are attempting to decrease in size contains system files (such as the root (/) volume), unmounting is possible only when booting from a bootable CD or floppy.

- 3 Decrease the size of the file system on a device called `/dev/sda1` by entering

```
resize_reiserfs -s size /dev/sda2
```

Replace *size* with the desired size in bytes. You can also specify units on the value, such as 50000K (kilobytes), 250M (megabytes), or 2G (gigabytes). Alternatively, you can specify a decrease to the current size by prefixing the value with a minus (-) sign. For example, the following command reduces the size of the file system on `/dev/md0` by 500 MB:

```
resize_reiserfs -s -500M /dev/sda2
```

Wait until the resizing is completed before continuing.

- 4 Mount the file system by entering

```
mount -t reiserfs /dev/sda2 /mnt/point
```

- 5 Check the effect of the resize on the mounted file system by entering

```
df -h
```

The Disk Free (`df`) command shows the total size of the disk, the number of blocks used, and the number of blocks available on the file system. The `-h` option print sizes in human-readable format, such as 1K, 234M, or 2G.

Using UUIDs to Mount Devices

This section describes the optional use of UUIDs instead of device names to identify file system devices in the boot loader file and the `/etc/fstab` file.

- Section 6.1, “Naming Devices with udev” (page 69)
- Section 6.2, “Understanding UUIDs” (page 70)
- Section 6.3, “Using UUIDs in the Boot Loader and `/etc/fstab` File (x86)” (page 71)
- Section 6.4, “Using UUIDs in the Boot Loader and `/etc/fstab` File (IA64)” (page 74)
- Section 6.5, “Additional Information” (page 75)

6.1 Naming Devices with udev

In the Linux 2.6 and later kernel, `udev` provides a userspace solution for the dynamic `/dev` directory, with persistent device naming. As part of the hotplug system, `udev` is executed if a device is added to or removed from the system.

A list of rules is used to match against specific device attributes. The `udev` rules infrastructure (defined in the `/etc/udev/rules.d` directory) provides stable names for all disk devices, regardless of their order of recognition or the connection used for the device. The `udev` tools examine every appropriate block device that the kernel creates to apply naming rules based on certain buses, drive types, or file systems. For information

about how to define your own rules for `udev`, see *Writing udev Rules* [http://reactivated.net/writing_udev_rules.html].

Along with the dynamic kernel-provided device node name, `udev` maintains classes of persistent symbolic links pointing to the device in the `/dev/disk` directory, which is further categorized by the `by-id`, `by-label`, `by-path`, and `by-uuid` subdirectories.

NOTE

Other programs besides `udev`, such as LVM or `md`, might also generate UUIDs, but they are not listed in `/dev/disk`.

6.2 Understanding UUIDs

A UUID (Universally Unique Identifier) is a 128-bit number for a file system that is unique on both the local system and across other systems. It is a randomly generated with system hardware information and time stamps as part of its seed. UUIDs are commonly used to uniquely tag devices.

- Section 6.2.1, “Using UUIDs to Assemble or Activate File System Devices” (page 70)
- Section 6.2.2, “Finding the UUID for a File System Device” (page 71)

6.2.1 Using UUIDs to Assemble or Activate File System Devices

The UUID is always unique to the partition and does not depend on the order in which it appears or where it is mounted. With certain SAN devices attached to the server, the system partitions are renamed and moved to be the last device. For example, if `root (/)` is assigned to `/dev/sda1` during the install, it might be assigned to `/dev/sdg1` after the SAN is connected. One way to avoid this problem is to use the UUID in the boot loader and `/etc/fstab` files for the boot device.

The device ID assigned by the manufacturer for a drive never changes, no matter where the device is mounted, so it can always be found at boot. The UUID is a property of the file system and can change if you reformat the drive. In a boot loader file, you typically specify the location of the device (such as `/dev/sda1`) to mount it at system boot. The boot loader can also mount devices by their UUIDs and administrator-specified volume labels. However, if you use a label and file location, you cannot change the label name when the partition is mounted.

You can use the UUID as criterion for assembling and activating software RAID devices. When a RAID is created, the `md` driver generates a UUID for the device, and stores the value in the `md` superblock.

6.2.2 Finding the UUID for a File System Device

You can find the UUID for any block device in the `/dev/disk/by-uuid` directory. For example, a UUID looks like this:

```
e014e482-1c2d-4d09-84ec-61b3aefde77a
```

6.3 Using UUIDs in the Boot Loader and `/etc/fstab` File (x86)

After the install, you can optionally use the following procedure to configure the UUID for the system device in the boot loader and `/etc/fstab` files for your x86 system.

Before you begin, make a copy of `/boot/grub/menu.lst` file and the `/etc/fstab` file.

- 1 Install the SUSE Linux Enterprise Server for x86 with no SAN devices connected.
- 2 After the install, boot the system.
- 3 Open a terminal console as the `root` user or equivalent.

- 4** Navigate to the `/dev/disk/by-uuid` directory to find the UUID for the device where you installed `/boot`, `/root`, and `swap`.

4a At the terminal console prompt, enter

```
cd /dev/disk/by-uuid
```

4b List all partitions by entering

```
ll
```

4c Find the UUID, such as

```
e014e482-1c2d-4d09-84ec-61b3aefde77a -> /dev/sda1
```

- 5** Edit `/boot/grub/menu.lst` file, using the Boot Loader option in YaST2 or using a text editor.

For example, change

```
kernel /boot/vmlinuz root=/dev/sda1
```

to

```
kernel /boot/vmlinuz  
root=/dev/disk/by-uuid/e014e482-1c2d-4d09-84ec-61b3aefde77a
```

IMPORTANT

If you make a mistake, you can boot the server without the SAN connected, and fix the error by using the backup copy of the `/boot/grub/menu.lst` file as a guide.

If you use the Boot Loader option in YaST, there is a defect where it adds some duplicate lines to the boot loader file when you change a value. Use an editor to remove the following duplicate lines:

```
color white/blue black/light-gray
```

```
default 0
```

```
timeout 8
```

```
gfxmenu (sd0,1)/boot/message
```

When you use YaST to change the way that the root (/) device is mounted (such as by UUID or by label), the boot loader configuration needs to be saved again to make the change effective for the boot loader.

6 As the `root` user or equivalent, do one of the following to place the UUID in the `/etc/fstab` file:

- Open YaST to *System > Partitioner*, select the device of interest, then modify *Fstab Options*.
- Edit the `/etc/fstab` file to modify the system device from the location to the UUID.

For example, if the root (/) volume has a device path of `/dev/sda1` and its UUID is `e014e482-1c2d-4d09-84ec-61b3aefde77a`, change line entry from

```
/dev/sda1    /                reiserfs    acl,user_xattr    1 1
```

to

```
UUID=e014e482-1c2d-4d09-84ec-61b3aefde77a    /    reiserfs    acl,user_xattr  
1 1
```

IMPORTANT

Do not leave stray characters or spaces in the file.

6.4 Using UUIDs in the Boot Loader and /etc/fstab File (IA64)

After the install, use the following procedure to configure the UUID for the system device in the boot loader and `/etc/fstab` files for your IA64 system. IA64 uses the EFI BIOS. Its file system configuration file is `/boot/efi/SuSE/elilo.conf` instead of `/etc/fstab`.

Before you begin, make a copy of the `/boot/efi/SuSE/elilo.conf` file.

- 1** Install the SUSE Linux Enterprise Server for IA64 with no SAN devices connected.
- 2** After the install, boot the system.
- 3** Open a terminal console as the `root` user or equivalent.
- 4** Navigate to the `/dev/disk/by-uuid` directory to find the UUID for the device where you installed `/boot`, `/root`, and `swap`.

4a At the terminal console prompt, enter

```
cd /dev/disk/by-uuid
```

4b List all partitions by entering

```
ll
```

4c Find the UUID, such as

```
e014e482-1c2d-4d09-84ec-61b3aefde77a -> /dev/sda1
```

- 5** Edit the boot loader file, using the Boot Loader option in YaST2.

For example, change

```
root=/dev/sda1
```

to

```
root=/dev/disk/by-uuid/e014e482-1c2d-4d09-84ec-61b3aefde77a
```

- 6 Edit the `/boot/efi/SuSE/elilo.conf` file to modify the system device from the location to the UUID.

For example, change

```
/dev/sda1 / reiserfs acl,user_xattr 1 1
```

to

```
UUID=e014e482-1c2d-4d09-84ec-61b3aefde77a / reiserfs acl,user_xattr  
1 1
```

IMPORTANT

Do not leave stray characters or spaces in the file.

6.5 Additional Information

For more information about using `udev(8)` for managing devices, see “Dynamic Kernel Device Management with `udev`” [http://www.suse.com/documentation/sles11/book_sle_admin/data/cha_udev.html] in the *SUSE Linux Enterprise Server 11 Administration Guide*.

For more information about `udev(8)` commands, see its man page. Enter the following at a terminal console prompt:

```
man 8 udev
```


Managing Multipath I/O for Devices

7

This section describes how to manage failover and path load balancing for multiple paths between the servers and block storage devices.

- Section 7.1, “Understanding Multipathing” (page 78)
- Section 7.2, “Planning for Multipathing” (page 78)
- Section 7.3, “Multipath Management Tools” (page 89)
- Section 7.4, “Configuring the System for Multipathing” (page 98)
- Section 7.5, “Enabling and Starting Multipath I/O Services” (page 111)
- Section 7.6, “Configuring Path Failover Policies and Priorities” (page 112)
- Section 7.7, “Configuring Multipath I/O for the Root Device” (page 126)
- Section 7.8, “Configuring Multipath I/O for an Existing Software RAID” (page 131)
- Section 7.9, “Scanning for New Devices without Rebooting” (page 134)
- Section 7.10, “Scanning for New Partitioned Devices without Rebooting” (page 137)
- Section 7.11, “Viewing Multipath I/O Status” (page 139)
- Section 7.12, “Managing I/O in Error Situations” (page 140)
- Section 7.13, “Resolving Stalled I/O” (page 142)

- Section 7.14, “Troubleshooting MPIO” (page 142)
- Section 7.15, “What’s Next” (page 143)

7.1 Understanding Multipathing

- Section 7.1.1, “What Is Multipathing?” (page 78)
- Section 7.1.2, “Benefits of Multipathing” (page 78)

7.1.1 What Is Multipathing?

Multipathing is the ability of a server to communicate with the same physical or logical block storage device across multiple physical paths between the host bus adapters in the server and the storage controllers for the device, typically in Fibre Channel (FC) or iSCSI SAN environments. You can also achieve multiple connections with direct attached storage when multiple channels are available.

7.1.2 Benefits of Multipathing

Linux multipathing provides connection fault tolerance and can provide load balancing across the active connections. When multipathing is configured and running, it automatically isolates and identifies device connection failures, and reroutes I/O to alternate connections.

Typical connection problems involve faulty adapters, cables, or controllers. When you configure multipath I/O for a device, the multipath driver monitors the active connection between devices. When the multipath driver detects I/O errors for an active path, it fails over the traffic to the device’s designated secondary path. When the preferred path becomes healthy again, control can be returned to the preferred path.

7.2 Planning for Multipathing

- Section 7.2.1, “Guidelines for Multipathing” (page 79)

- Section 7.2.2, “Using By-ID Names for Multipathed Devices” (page 81)
- Section 7.2.3, “Using LVM2 on Multipath Devices” (page 82)
- Section 7.2.4, “Using mdadm with Multipath Devices” (page 83)
- Section 7.2.5, “Using --noflush with Multipath Devices” (page 83)
- Section 7.2.6, “SAN Timeout Settings When the Root Device Is Multipathed” (page 84)
- Section 7.2.7, “Partitioning Multipath Devices” (page 85)
- Section 7.2.8, “Supported Architectures for Multipath I/O” (page 86)
- Section 7.2.9, “Supported Storage Arrays for Multipathing” (page 86)

7.2.1 Guidelines for Multipathing

Use the guidelines in this section when planning your multipath I/O solution.

- Section “Prerequisites” (page 79)
- Section “Vendor-Provided Multipath Solutions” (page 80)
- Section “Disk Management Tasks” (page 80)
- Section “Software RAIDs” (page 80)
- Section “High-Availability Solutions” (page 81)
- Section “Volume Managers” (page 81)
- Section “Virtualization Environments” (page 81)

Prerequisites

- Multipathing is managed at the device level.

- The storage array you use for the multipathed device must support multipathing. For more information, see Section 7.2.9, “Supported Storage Arrays for Multipathing” (page 86).
- You need to configure multipathing only if multiple physical paths exist between host bus adapters in the server and host bus controllers for the block storage device. You configure multipathing for the logical device as seen by the server.

Vendor-Provided Multipath Solutions

For some storage arrays, the vendor provides its own multipathing software to manage multipathing for the array’s physical and logical devices. In this case, you should follow the vendor’s instructions for configuring multipathing for those devices.

Disk Management Tasks

Perform the following disk management tasks before you attempt to configure multipathing for a physical or logical device that has multiple paths:

- Use third-party tools to carve physical disks into smaller logical disks.
- Use third-party tools to partition physical or logical disks. If you change the partitioning in the running system, the Device Mapper Multipath (DM-MP) module does not automatically detect and reflect these changes. DM-MPIO must be reinitialized, which usually requires a reboot.
- Use third-party SAN array management tools to create and configure hardware RAID devices.
- Use third-party SAN array management tools to create logical devices such as LUNs. Logical device types that are supported for a given array depend on the array vendor.

Software RAIDs

The Linux software RAID management software runs on top of multipathing. For each device that has multiple I/O paths and that you plan to use in a software RAID, you must configure the device for multipathing before you attempt to create the software RAID device. Automatic discovery of multipathed devices is not available. The software RAID is not aware of the multipathing management running underneath.

High-Availability Solutions

High-availability solutions for clustering typically run on top of the multipathing server. For example, the Distributed Replicated Block Device (DRBD) high-availability solution for mirroring devices across a LAN runs on top of multipathing. For each device that has multiple I/O paths and that you plan to use in a DRBD solution, you must configure the device for multipathing before you configure DRBD.

Volume Managers

Volume managers such as LVM2 and EVMS run on top of multipathing. You must configure multipathing for a device before you use LVM2 or EVMS to create segment managers and file systems on it.

Virtualization Environments

When using multipathing in a virtualization environment, the multipathing is controlled in the host server environment. Configure multipathing for the device before you assign it to a virtual guest machine.

7.2.2 Using By-ID Names for Multipathed Devices

If you want to use the entire LUN directly (for example, if you are using the SAN features to partition your storage), you can use the `/dev/disk/by-id/xxx` names for `mkfs`, `fstab`, your application, and so on.

If the user-friendly names option is enabled in the `/etc/multipath.conf` file, you can use the `/dev/disk/by-id/dm-uuid-.*-mpath-.*` device name because this name is aliased to the device ID. For information, see Section “Configuring User-Friendly Names or Alias Names in `/etc/multipath.conf`” (page 104).

7.2.3 Using LVM2 on Multipath Devices

By default, LVM2 does not recognize multipathed devices. To make LVM2 recognize the multipathed devices as possible physical volumes, you must modify `/etc/lvm/lvm.conf`. It is important to modify it so that it does not scan and use the physical paths, but only accesses the multipath I/O storage through the multipath I/O layer. If you are using user-friendly names, ensure that you specify the path so that it scans only the device mapper names for the device (`/dev/disk/by-id/dm-uuid-.*-mpath-.*`) after multipathing is configured.

To modify `/etc/lvm/lvm.conf` for multipath use:

- 1 Open the `/etc/lvm/lvm.conf` file in a text editor.

If `/etc/lvm/lvm.conf` does not exist, you can create one based on your current LVM configuration by entering the following at a terminal console prompt:

```
lvm dumpconfig > /etc/lvm/lvm.conf
```

- 2 Change the `filter` and `types` entries in `/etc/lvm/lvm.conf` as follows:

```
filter = [ "a|/dev/disk/by-id/.*/", "r|.*/" ]
types = [ "device-mapper", 1 ]
```

This allows LVM2 to scan only the by-id paths and reject everything else.

If you are using user-friendly names, specify the path as follows so that only the device mapper names are scanned after multipathing is configured:

```
filter = [ "a|/dev/disk/by-id/dm-uuid-.*-mpath-.*|", "r|.*/" ]
```

- 3 If you are also using LVM2 on non-multipathed devices, make the necessary adjustments in the `filter` and `types` entries to suit your setup. Otherwise, the other LVM devices are not visible with a `pvs` scan after you modify the `lvm.conf` file for multipathing.

You want only those devices that are configured with LVM to be included in the LVM cache, so ensure that you are specific about which other non-multipathed devices are included by the filter.

For example, if your local disk is `/dev/sda` and all SAN devices are `/dev/sdb` and above, specify the local and multipathing paths in the filter as follows:

```
filter = [ "a|/dev/sda.*|", "a|/dev/disk/by-id/*|", "r|.*|" ]
types = [ "device-mapper", 253 ]
```

- 4 Save the file.
- 5 Add `dm-multipath` to `/etc/sysconfig/kernel:INITRD_MODULES`.
- 6 Make a new `initrd` to ensure that the Device Mapper Multipath services are loaded with the changed settings. Running `mkinitrd` is needed only if the root (`/`) device or any parts of it (such as `/var`, `/etc`, `/log`) are on the SAN and multipath is needed to boot.

Enter the following at a terminal console prompt:

```
mkinitrd -f multipath
```

- 7 Reboot the server to apply the changes.

7.2.4 Using mdadm with Multipath Devices

The `mdadm` tool requires that the devices be accessed by the ID rather than by the device node path. Therefore, the `DEVICE` entry in `/etc/mdadm.conf` should be set as follows:

```
DEVICE /dev/disk/by-id/*
```

If you are using user-friendly names, specify the path as follows so that only the device mapper names are scanned after multipathing is configured:

```
DEVICE /dev/disk/by-id/dm-uuid-.*-mpath-.*
```

7.2.5 Using --noflush with Multipath Devices

The `--noflush` option should always be used when running on multipath devices.

For example, in scripts where you perform a table reload, you use the `--noflush` option on `resume` to ensure that any outstanding I/O is not flushed, because you need the multipath topology information.

```
load
resume --noflush
```

7.2.6 SAN Timeout Settings When the Root Device Is Multipathed

A system with root (/) on a multipath device might stall when all paths have failed and are removed from the system because a `dev_loss_tmo` time-out is received from the storage subsystem (such as Fibre Channel storage arrays).

If the system device is configured with multiple paths and the multipath `no_path_retry` setting is active, you should modify the storage subsystem's `dev_loss_tmo` setting accordingly to ensure that no devices are removed during an all-paths-down scenario. We strongly recommend that you set the `dev_loss_tmo` value to be equal to or higher than the `no_path_retry` setting from multipath.

The recommended setting for the storage subsystem's `dev_loss_tmo` is:

```
<dev_loss_tmo> = <no_path_retry> * <polling_interval>
```

where the following definitions apply for the multipath values:

- `no_path_retry` is the number of retries for multipath I/O until the path is considered to be lost, and queuing of IO is stopped.
- `polling_interval` is the time in seconds between path checks.

Each of these multipath values should be set from the `/etc/multipath.conf` configuration file. For information, see Section 7.4.5, “Creating and Configuring the `/etc/multipath.conf` File” (page 101).

7.2.7 Partitioning Multipath Devices

Behavior changes for how multipathed devices are handled might affect your configuration if you are upgrading.

- Section “SUSE Linux Enterprise Server 11” (page 85)
- Section “SUSE Linux Enterprise Server 10” (page 85)
- Section “SUSE Linux Enterprise Server 9” (page 85)

SUSE Linux Enterprise Server 11

In SUSE Linux Enterprise Server 11, the default multipath setup relies on `udev` to overwrite the existing symbolic links in the `/dev/disk/by-id` directory when multipathing is started. Before you start multipathing, the link points to the SCSI device by using its `scsi-xxx` name. When multipathing is running, the symbolic link points to the device by using its `dm-uuid-xxx` name. This ensures that the symbolic links in the `/dev/disk/by-id` path persistently point to the same device regardless of whether multipath is started or not. The configuration files (such as `lvm.conf` and `md.conf`) do not need to be modified because they automatically point to the correct device.

SUSE Linux Enterprise Server 10

In SUSE Linux Enterprise Server 10, the `kpartx` software is used in the `/etc/init.d/boot.multipath` to add symlinks to the `/dev/dm-*` line in the `multipath.conf` configuration file for any newly created partitions without requiring a reboot. This triggers `udev` to fill in the `/dev/disk/by-*` symlinks. The main benefit is that you can call `kpartx` with the new parameters without rebooting the server.

SUSE Linux Enterprise Server 9

In SUSE Linux Enterprise Server 9, it is not possible to partition multipath I/O devices themselves. If the underlying physical device is already partitioned, the multipath I/O device reflects those partitions and the layer provides `/dev/disk/by-id/<name>p1 . . . pN` devices so you can access the partitions through the multipath I/O layer. As

a consequence, the devices need to be partitioned prior to enabling multipath I/O. If you change the partitioning in the running system, DM-MPIO does not automatically detect and reflect these changes. The device must be reinitialized, which usually requires a reboot.

7.2.8 Supported Architectures for Multipath I/O

The multipathing drivers and tools support all seven of the supported processor architectures: IA32, AMD64/EM64T, IPF/IA64, p-Series (32-bit and 64-bit), and z-Series (31-bit and 64-bit).

7.2.9 Supported Storage Arrays for Multipathing

The multipathing drivers and tools support most storage arrays. The storage array that houses the multipathed device must support multipathing in order to use the multipathing drivers and tools. Some storage array vendors provide their own multipathing management tools. Consult the vendor's hardware documentation to determine what settings are required.

- Section “Storage Arrays That Are Automatically Detected for Multipathing” (page 86)
- Section “Tested Storage Arrays for Multipathing Support” (page 88)
- Section “Storage Arrays that Require Specific Hardware Handlers” (page 88)

Storage Arrays That Are Automatically Detected for Multipathing

The `multipath-tools` package automatically detects the following storage arrays:

3PARdata VV
Compaq HSV110
Compaq MSA1000

DDN SAN MultiDirector
DEC HSG80
EMC CLARiiON CX
EMC Symmetrix
FSC CentricStor
Hewlett Packard (HP) A6189A
HP HSV110
HP HSV210
HP Open
Hitachi DF400
Hitachi DF500
Hitachi DF600
IBM 3542
IBM ProFibre 4000R
NetApp
SGI TP9100
SGI TP9300
SGI TP9400
SGI TP9500
STK OPENstorage DS280
Sun StorEdge 3510
Sun T4

In general, most other storage arrays should work. When storage arrays are automatically detected, the default settings for multipathing apply. If you want non-default settings, you must manually create and configure the `/etc/multipath.conf` file. For information, see Section 7.4.5, “Creating and Configuring the `/etc/multipath.conf` File” (page 101).

Testing of the IBM zSeries device with multipathing has shown that the `dev_loss_tmo` parameter should be set to 90 seconds, and the `fast_io_fail_tmo` parameter should be set to 5 seconds. If you are using zSeries devices, you must manually create and configure the `/etc/multipath.conf` file to specify the values. For information, see Section “Configuring Default Settings for zSeries in `/etc/multipath.conf`” (page 110).

Hardware that is not automatically detected requires an appropriate entry for configuration in the `DEVICES` section of the `/etc/multipath.conf` file. In this case, you must manually create and configure the configuration file. For information, see Section 7.4.5, “Creating and Configuring the `/etc/multipath.conf` File” (page 101).

Consider the following caveats:

- Not all of the storage arrays that are automatically detected have been tested on SUSE Linux Enterprise Server. For information, see Section “Tested Storage Arrays for Multipathing Support” (page 88).
- Some storage arrays might require specific hardware handlers. A hardware handler is a kernel module that performs hardware-specific actions when switching path groups and dealing with I/O errors. For information, see Section “Storage Arrays that Require Specific Hardware Handlers” (page 88).
- After you modify the `/etc/multipath.conf` file, you must run `mkinitrd` to re-create the INITRD on your system, then reboot in order for the changes to take effect.

Tested Storage Arrays for Multipathing Support

The following storage arrays have been tested with SUSE Linux Enterprise Server:

EMC
Hitachi
Hewlett-Packard/Compaq
IBM
NetApp
SGI

Most other vendor storage arrays should also work. Consult your vendor’s documentation for guidance. For a list of the default storage arrays recognized by the `multipath-tools` package, see Section “Storage Arrays That Are Automatically Detected for Multipathing” (page 86).

Storage Arrays that Require Specific Hardware Handlers

Storage arrays that require special commands on failover from one path to the other or that require special nonstandard error handling might require more extensive support. Therefore, the Device Mapper Multipath service has hooks for hardware handlers. For example, one such handler for the EMC CLARiiON CX family of arrays is already provided.

IMPORTANT

Consult the hardware vendor's documentation to determine if its hardware handler must be installed for Device Mapper Multipath.

The `multipath -t` command shows an internal table of storage arrays that require special handling with specific hardware handlers. The displayed list is not an exhaustive list of supported storage arrays. It lists only those arrays that require special handling and that the `multipath-tools` developers had access to during the tool development.

IMPORTANT

Arrays with true active/active multipath support do not require special handling, so they are not listed for the `multipath -t` command.

A listing in the `multipath -t` table does not necessarily mean that SUSE Linux Enterprise Server was tested on that specific hardware. For a list of tested storage arrays, see Section “Tested Storage Arrays for Multipathing Support” (page 88).

7.3 Multipath Management Tools

The multipathing support in SUSE Linux Enterprise Server 10 and later is based on the Device Mapper Multipath module of the Linux 2.6 kernel and the `multipath-tools` userspace package. You can use the Multiple Devices Administration utility (MDADM, `mdadm`) to view the status of multipathed devices.

- Section 7.3.1, “Device Mapper Multipath Module” (page 90)
- Section 7.3.2, “Multipath I/O Management Tools” (page 92)
- Section 7.3.3, “Using MDADM for Multipathed Devices” (page 94)
- Section 7.3.4, “The Linux `multipath(8)` Command” (page 95)

7.3.1 Device Mapper Multipath Module

The Device Mapper Multipath (DM-MP) module provides the multipathing capability for Linux. DM-MPIO is the preferred solution for multipathing on SUSE Linux Enterprise Server 11. It is the only multipathing option shipped with the product that is completely supported by Novell and SUSE.

DM-MPIO features automatic configuration of the multipathing subsystem for a large variety of setups. Configurations of up to 8 paths to each device are supported. Configurations are supported for active/passive (one path active, others passive) or active/active (all paths active with round-robin load balancing).

The DM-MPIO framework is extensible in two ways:

- Using specific hardware handlers. For information, see Section “Storage Arrays that Require Specific Hardware Handlers” (page 88).
- Using load-balancing algorithms that are more sophisticated than the round-robin algorithm

The user-space component of DM-MPIO takes care of automatic path discovery and grouping, as well as automated path retesting, so that a previously failed path is automatically reinstated when it becomes healthy again. This minimizes the need for administrator attention in a production environment.

DM-MPIO protects against failures in the paths to the device, and not failures in the device itself. If one of the active paths is lost (for example, a network adapter breaks or a fiber-optic cable is removed), I/O is redirected to the remaining paths. If the configuration is active/passive, then the path fails over to one of the passive paths. If you are using the round-robin load-balancing configuration, the traffic is balanced across the remaining healthy paths. If all active paths fail, inactive secondary paths must be waked up, so failover occurs with a delay of approximately 30 seconds.

If a disk array has more than one storage processor, ensure that the SAN switch has a connection to the storage processor that owns the LUNs you want to access. On most disk arrays, all LUNs belong to both storage processors, so both connections are active.

NOTE

On some disk arrays, the storage array manages the traffic through storage processors so that it presents only one storage processor at a time. One processor is active and the other one is passive until there is a failure. If you are connected to the wrong storage processor (the one with the passive path) you might not see the expected LUNs, or you might see the LUNs but get errors when you try to access them.

Table 7.1 *Multipath I/O Features of Storage Arrays*

Features of Storage Arrays	Description
Active/passive controllers	<p>One controller is active and serves all LUNs. The second controller acts as a standby. The second controller also presents the LUNs to the multipath component so that the operating system knows about redundant paths. If the primary controller fails, the second controller takes over, and it serves all LUNs.</p> <p>In some arrays, the LUNs can be assigned to different controllers. A given LUN is assigned to one controller to be its active controller. One controller does the disk I/O for any given LUN at a time, and the second controller is the standby for that LUN. The second controller also presents the paths, but disk I/O is not possible. Servers that use that LUN are connected to the LUN's assigned controller. If the primary controller for a set of LUNs fails, the second controller takes over, and it serves all LUNs.</p>
Active/active controllers	Both controllers share the load for all LUNs, and can process disk I/O for any given LUN. If one controller fails, the second controller automatically handles all traffic.
Load balancing	The Device Mapper Multipath driver automatically load balances traffic across all active paths.

Features of Storage Arrays	Description
Controller failover	When the active controller fails over to the passive, or standby, controller, the Device Mapper Multipath driver automatically activates the paths between the host and the standby, making them the primary paths.
Boot/Root device support	<p>Multipathing is supported for the root (/) device in SUSE Linux Enterprise Server 10 and later. The host server must be connected to the currently active controller and storage processor for the boot device.</p> <p>Multipathing is supported for the <code>/boot</code> device in SUSE Linux Enterprise Server 11 and later.</p>

Device Mapper Multipath detects every path for a multipathed device as a separate SCSI device. The SCSI device names take the form `/dev/sdN`, where *N* is an autogenerated letter for the device, beginning with a and issued sequentially as the devices are created, such as `/dev/sda`, `/dev/sdb`, and so on. If the number of devices exceeds 26, the letters are duplicated so that the next device after `/dev/sdz` will be named `/dev/sdaa`, `/dev/sdab`, and so on.

If multiple paths are not automatically detected, you can configure them manually in the `/etc/multipath.conf` file. The `multipath.conf` file does not exist until you create and configure it. For information, see Section 7.4.5, “Creating and Configuring the `/etc/multipath.conf` File” (page 101).

7.3.2 Multipath I/O Management Tools

The `multipath-tools` user-space package takes care of automatic path discovery and grouping. It automatically tests the path periodically, so that a previously failed path is automatically reinstated when it becomes healthy again. This minimizes the need for administrator attention in a production environment.

Table 7.2 *Tools in the multipath-tools Package*

Tool	Description
<code>multipath</code>	Scans the system for multipathed devices and assembles them.
<code>multipathd</code>	Waits for maps events, then executes <code>multipath</code> .
<code>devmap-name</code>	Provides a meaningful device name to <code>udev</code> for device maps (devmaps).
<code>kpartx</code>	Maps linear devmaps to partitions on the multipathed device, which makes it possible to create multipath monitoring for partitions on the device.

The file list for a package can vary for different server architectures. For a list of files included in the `multipath-tools` package, go to the *SUSE Linux Enterprise Server Technical Specifications > Package Descriptions* Web page [<http://www.novell.com/products/server/techspecs.html?tab=1>], find your architecture and select *Packages Sorted by Name*, then search on “multipath-tools” to find the package list for that architecture.

You can also determine the file list for an RPM file by querying the package itself: using the `rpm -ql` or `rpm -qpl` command options.

- To query an installed package, enter

```
rpm -ql <package_name>
```

- To query a package not installed, enter

```
rpm -qpl <URL_or_path_to_package>
```

To check that the `multipath-tools` package is installed, do the following:

- Enter the following at a terminal console prompt:

```
rpm -q multipath-tools
```

If it is installed, the response repeats the package name and provides the version information, such as:

```
multipath-tools-04.7-34.23
```

If it is not installed, the response reads:

```
package multipath-tools is not installed
```

7.3.3 Using MDADM for Multipathed Devices

Udev is the default device handler, and devices are automatically known to the system by the Worldwide ID instead of by the device node name. This resolves problems in previous releases of MDADM and LVM where the configuration files (`mdadm.conf` and `lvm.conf`) did not properly recognize multipathed devices.

Just as for LVM2, MDADM requires that the devices be accessed by the ID rather than by the device node path. Therefore, the `DEVICE` entry in `/etc/mdadm.conf` should be set as follows:

```
DEVICE /dev/disk/by-id/*
```

If you are using user-friendly names, specify the path as follows so that only the device mapper names are scanned after multipathing is configured:

```
DEVICE /dev/disk/by-id/dm-uuid-.*-mpath-.*
```

To verify that MDADM is installed:

- Ensure that the `mdadm` package is installed by entering the following at a terminal console prompt:

```
rpm -q mdadm
```

If it is installed, the response repeats the package name and provides the version information. For example:


```
mdadm-2.6-0.11
```

If it is not installed, the response reads:

```
package mdadm is not installed
```

For information about modifying the `/etc/lvm/lvm.conf` file, see Section 7.2.3, “Using LVM2 on Multipath Devices” (page 82).

7.3.4 The Linux `multipath(8)` Command

Use the Linux `multipath(8)` command to configure and manage multipathed devices.

General syntax for the `multipath(8)` command:

```
multipath [-v verbosity] [-d] [-h|-l|-ll|-f|-F] [-p failover | multibus |  
group_by_serial | group_by_prio | group_by_node_name ]
```

General Examples

`multipath`

Configure all multipath devices.

`multipath devicename`

Configures a specific multipath device.

Replace *devicename* with the device node name such as `/dev/sdb` (as shown by `udev` in the `$DEVNAME` variable), or in the `major:minor` format.

`multipath -f`

Selectively suppresses a multipath map, and its device-mapped partitions.

`multipath -d`

Dry run. Displays potential multipath devices, but does not create any devices and does not update device maps.

`multipath -v2 -d`

Displays multipath map information for potential multipath devices in a dry run. The `-v2` option shows only local disks. This verbosity level prints the created or updated multipath names only for use to feed other tools like `kpartx`.

There is no output if the devices already exists and there are no changes. Use `multipath -ll` to see the status of configured multipath devices.

`multipath -v2 devicename`

Configures a specific potential multipath device and displays multipath map information for it. This verbosity level prints only the created or updated multipath names for use to feed other tools like `kpartx`.

There is no output if the device already exists and there are no changes. Use `multipath -ll` to see the status of configured multipath devices.

Replace *devicename* with the device node name such as `/dev/sdb` (as shown by `udev` in the `$DEVNAME` variable), or in the `major:minor` format.

`multipath -v3`

Configures potential multipath devices and displays multipath map information for them. This verbosity level prints all detected paths, multipaths, and device maps. Both `wwid` and `devnode` blacklisted devices are displayed.

`multipath -v3 devicename`

Configures a specific potential multipath device and displays information for it. The `-v3` option shows the full path list. This verbosity level prints all detected paths, multipaths, and device maps. Both `wwid` and `devnode` blacklisted devices are displayed.

Replace *devicename* with the device node name such as `/dev/sdb` (as shown by `udev` in the `$DEVNAME` variable), or in the `major:minor` format.

`multipath -ll`

Display the status of all multipath devices.

`multipath -ll devicename`

Displays the status of a specified multipath device.

Replace *devicename* with the device node name such as `/dev/sdb` (as shown by `udev` in the `$DEVNAME` variable), or in the `major:minor` format.

multipath -F

Flushes all unused multipath device maps. This unresolves the multiple paths; it does not delete the devices.

multipath -F *devicename*

Flushes unused multipath device maps for a specified multipath device. This unresolves the multiple paths; it does not delete the device.

Replace *devicename* with the device node name such as `/dev/sdb` (as shown by `udev` in the `$DEVNAME` variable), or in the `major:minor` format.

multipath -p [failover | multibus | group_by_serial | group_by_prio | group_by_node_name]

Sets the group policy by specifying one of the group policy options that are described in Table 7.3, “Group Policy Options for the multipath -p Command” (page 97):

Table 7.3 *Group Policy Options for the multipath -p Command*

Policy Option	Description
failover	One path per priority group. You can use only one path at a time.
multibus	All paths in one priority group.
group_by_serial	One priority group per detected SCSI serial number (the controller node worldwide number).
group_by_prio	One priority group per path priority value. Paths with the same priority are in the same priority group. Priorities are determined by callout programs specified as a global, per-controller, or per-multipath option in the <code>/etc/multipath.conf</code> configuration file.

Policy Option	Description
<code>group_by_node_name</code>	One priority group per target node name. Target node names are fetched in the <code>/sys/class/fc*_transport/target*/node_name</code> location.

7.4 Configuring the System for Multipathing

- Section 7.4.1, “Preparing SAN Devices for Multipathing” (page 98)
- Section 7.4.2, “Partitioning Multipathed Devices” (page 99)
- Section 7.4.3, “Configuring the Server for Multipathing” (page 99)
- Section 7.4.4, “Adding multipathd to the Boot Sequence” (page 100)
- Section 7.4.5, “Creating and Configuring the `/etc/multipath.conf` File” (page 101)

7.4.1 Preparing SAN Devices for Multipathing

Before configuring multipath I/O for your SAN devices, prepare the SAN devices, as necessary, by doing the following:

- Configure and zone the SAN with the vendor’s tools.
- Configure permissions for host LUNs on the storage arrays with the vendor’s tools.
- Install the Linux HBA driver module. Upon module installation, the driver automatically scans the HBA to discover any SAN devices that have permissions for the host. It presents them to the host for further configuration.

NOTE

Ensure that the HBA driver you are using does not have native multipathing enabled.

See the vendor's specific instructions for more details.

- After the driver module is loaded, discover the device nodes assigned to specific array LUNs or partitions.
- If the SAN device will be used as the root device on the server, modify the timeout settings for the device as described in Section 7.2.6, “SAN Timeout Settings When the Root Device Is Multipathed” (page 84).

If the LUNs are not seen by the HBA driver, `ls SCSI` can be used to check whether the SCSI devices are seen correctly by the operating system. When the LUNs are not seen by the HBA driver, check the zoning setup of the SAN. In particular, check whether LUN masking is active and whether the LUNs are correctly assigned to the server.

If the LUNs are seen by the HBA driver, but there are no corresponding block devices, additional kernel parameters are needed to change the SCSI device scanning behavior, such as to indicate that LUNs are not numbered consecutively. For information, see *TID 3955167: Troubleshooting SCSI (LUN) Scanning Issues* in the Novell Support Knowledgebase [<http://support.novell.com/>].

7.4.2 Partitioning Multipathed Devices

Partitioning devices that have multiple paths is not recommended, but it is supported. You can use the `kpartx` tool to create partitions on multipathed devices without rebooting. You can also partition the device before you attempt to configure multipathing by using the Partitioner function in YaST2, or by using a third-party partitioning tool.

7.4.3 Configuring the Server for Multipathing

The system must be manually configured to automatically load the device drivers for the controllers to which the multipath I/O devices are connected within the `initrd`.

You need to add the necessary driver module to the variable `INITRD_MODULES` in the file `/etc/sysconfig/kernel`.

For example, if your system contains a RAID controller accessed by the `cciss` driver and multipathed devices connected to a QLogic controller accessed by the driver `qla2xxx`, this entry would look like:

```
INITRD_MODULES="cciss"
```

Because the QLogic driver is not automatically loaded on startup, add it here:

```
INITRD_MODULES="cciss qla23xx"
```

After changing `/etc/sysconfig/kernel`, you must re-create the `initrd` on your system with the `mkinitrd` command, then reboot in order for the changes to take effect.

When you are using LILO as a boot manager, reinstall it with the `/sbin/lilo` command. No further action is required if you are using GRUB.

7.4.4 Adding multipathd to the Boot Sequence

Use either of the methods in this section to add multipath I/O services (`multipathd`) to the boot sequence.

- Section “Using YaST to Add multipathd” (page 100)
- Section “Using the Command Line to Add multipathd” (page 101)

Using YaST to Add multipathd

- 1 In YaST, click *System > System Services (Runlevel) > Simple Mode*.
- 2 Select *multipathd*, then click *Enable*.
- 3 Click *OK* to acknowledge the service startup message.

- 4 Click *Finish*, then click *Yes*.

The changes do not take affect until the server is restarted.

Using the Command Line to Add multipathd

- 1 Open a terminal console, then log in as the `root` user or equivalent.
- 2 At the terminal console prompt, enter

```
insserv multipathd
```

7.4.5 Creating and Configuring the `/etc/multipath.conf` File

The `/etc/multipath.conf` file does not exist unless you create it. The `/usr/share/doc/packages/multipath-tools/multipath.conf.synthetic` file contains a sample `/etc/multipath.conf` file that you can use as a guide for multipath settings. See `/usr/share/doc/packages/multipath-tools/multipath.conf.annotated` for a template with extensive comments for each of the attributes and their options.

- Section “Creating the `multipath.conf` File” (page 102)
- Section “Verifying the Setup in the `etc/multipath.conf` File” (page 102)
- Section “Configuring User-Friendly Names or Alias Names in `/etc/multipath.conf`” (page 104)
- Section “Blacklisting Non-Multipathed Devices in `/etc/multipath.conf`” (page 109)
- Section “Configuring Default Multipath Behavior in `/etc/multipath.conf`” (page 110)
- Section “Configuring Default Settings for zSeries in `/etc/multipath.conf`” (page 110)
- Section “Applying the `/etc/multipath.conf` File Changes” (page 111)

Creating the multipath.conf File

If the `/etc/multipath.conf` file does not exist, copy the example to create the file:

- 1 In a terminal console, log in as the `root` user.
- 2 Enter the following command (all on one line, of course) to copy the template:

```
cp /usr/share/doc/packages/multipath-tools/multipath.conf.synthetic  
/etc/multipath.conf
```

- 3 Use the `/usr/share/doc/packages/multipath-tools/multipath.conf.annotated` file as a reference to determine how to configure multipathing for your system.
- 4 Ensure that there is an appropriate `device` entry for your SAN. Most vendors provide documentation on the proper setup of the `device` section.

The `/etc/multipath.conf` file requires a different `device` section for different SANs. If you are using a storage subsystem that is automatically detected (see Section “Tested Storage Arrays for Multipathing Support” (page 88)), the default entry for that device can be used; no further configuration of the `/etc/multipath.conf` file is required.

- 5 Save the file.

Verifying the Setup in the etc/multipath.conf File

After setting up the configuration, you can perform a “dry run” by entering

```
multipath -v2 -d
```

This command scans the devices, then displays what the setup would look like. The output is similar to the following:

```
26353900f02796769  
[size=127 GB]  
[features="0"]  
[hwhandler="1      emc"]
```



```

\_ round-robin 0 [first]
  \_ 1:0:1:2 sdav 66:240 [ready ]
  \_ 0:0:1:2 sdr  65:16  [ready ]

\_ round-robin 0
  \_ 1:0:0:2 sdag 66:0   [ready ]
  \_ 0:0:0:2 sdc  8:32   [ready ]

```

Paths are grouped into priority groups. Only one priority group is in active use at a time. To model an active/active configuration, all paths end in the same group. To model active/passive configuration, the paths that should not be active in parallel are placed in several distinct priority groups. This normally happens automatically on device discovery.

The output shows the order, the scheduling policy used to balance I/O within the group, and the paths for each priority group. For each path, its physical address (host:bus:target:lun), device node name, major:minor number, and state is shown.

By using a verbosity level of -v3 in the dry run, you can see all detected paths, multipaths, and device maps. Both wwid and device node blacklisted devices are displayed.

```

multipath -v3 d

```

The following is an example of -v3 output on a 64-bit SLES server with two Qlogic HBA connected to a Xiotech Magnitude 3000 SAN. Some multiple entries have been omitted to shorten the example.

```

dm-22: device node name blacklisted
< content omitted >
loop7: device node name blacklisted
< content omitted >
md0: device node name blacklisted
< content omitted >
dm-0: device node name blacklisted
sdf: not found in pathvec
sdf: mask = 0x1f
sdf: dev_t = 8:80
sdf: size = 105005056
sdf: subsystem = scsi
sdf: vendor = XIotech
sdf: product = Magnitude 3D
sdf: rev = 3.00
sdf: h:b:t:l = 1:0:0:2
sdf: tgt_node_name = 0x202100d0b2028da
sdf: serial = 000028DA0014

```

```

sdf: getuid = /lib/udev/scsi_id -g -u -s /block/%n (config file default)
sdf: uid = 200d0b2da28001400 (callout)
sdf: prio = const (config file default)
sdf: const prio = 1
< content omitted >
raml5: device node name blacklisted
< content omitted >
===== paths list =====
uuid          hcil      dev dev_t pri dm_st  chk_st  vend/prod/rev
200d0b2da28001400 1:0:0:2 sdf 8:80 1 [undef][undef] XIOTech,Magnitude 3D
200d0b2da28005400 1:0:0:1 sde 8:64 1 [undef][undef] XIOTech,Magnitude 3D
200d0b2da28004d00 1:0:0:0 sdd 8:48 1 [undef][undef] XIOTech,Magnitude 3D
200d0b2da28001400 0:0:0:2 sdc 8:32 1 [undef][undef] XIOTech,Magnitude 3D
200d0b2da28005400 0:0:0:1 sdb 8:16 1 [undef][undef] XIOTech,Magnitude 3D
200d0b2da28004d00 0:0:0:0 sda 8:0 1 [undef][undef] XIOTech,Magnitude 3D
params = 0 0 2 1 round-robin 0 1 1 8:80 1000 round-robin 0 1 1 8:32 1000
status = 2 0 0 0 2 1 A 0 1 0 8:80 A 0 E 0 1 0 8:32 A 0
sdf: mask = 0x4
sdf: path checker = directio (config file default)
directio: starting new request
directio: async io getevents returns 1 (errno=Success)
directio: io finished 4096/0
sdf: state = 2
< content omitted >

```

Configuring User-Friendly Names or Alias Names in /etc/multipath.conf

A multipath device can be identified by its WWID, by a user-friendly name, or by an alias that you assign for it. Table 7.4, “Comparison of Multipath Device Name Types” (page 104) describes the types of device names that can be used for a device in the /etc/multipath.conf file.

Table 7.4 *Comparison of Multipath Device Name Types*

Name Types	Description
WWID (default)	The WWID (Worldwide Identifier) is an identifier for the multipath device that is guaranteed to be globally unique and unchanging. The default name used in multipathing is the ID of the logical unit as found in the /dev/disk/by-id directory. Because device node names in the form of /dev/sdn and /dev/dm-n can change on reboot, referring to multipath devices by their ID is preferred.

Name Types	Description
User-friendly	The Device Mapper Multipath device names in the <code>/dev/mapper</code> directory also reference the ID of the logical unit. These multipath device names are user-friendly names in the form of <code>/dev/mapper/mpath<n></code> , such as <code>/dev/mapper/mpath0</code> . The names are unique and persistent because they use the <code>/var/lib/multipath/bindings</code> file to track the association between the UUID and user-friendly names.
Alias	An alias name is a globally unique name that the administrator provides for a multipath device. Alias names override the WWID and the user-friendly <code>/dev/mapper/mpathN</code> names.

The global multipath `user_friendly_names` option in the `/etc/multipath.conf` file is used to enable or disable the use of user-friendly names for multipath devices. If it is set to “no” (the default), multipath uses the WWID as the name of the device. If it is set to “yes”, multipath uses the `/var/lib/multipath/bindings` file to assign a persistent and unique name to the device in the form of `mpath<n>`. The `bindings_file` option in the `/etc/multipath.conf` file can be used to specify an alternate location for the `bindings` file.

The global multipath `alias` option in the `/etc/multipath.conf` file is used to explicitly assign a name to the device. If an alias name is set up for a multipath device, the alias is used instead of the WWID or the user-friendly name.

Using the `user_friendly_names` option can be problematic in the following situations:

- **Root Device Is Using Multipath:** If the system root device is using multipath and you use the `user_friendly_names` option, the user-friendly settings in the `/var/lib/multipath/bindings` file are included in the `initrd`. If you later change the storage setup, such as by adding or removing devices, there is a mismatch between the bindings setting inside the `initrd` and the bindings settings in `/var/lib/multipath/bindings`.

WARNING

A bindings mismatch between `initrd` and `/var/lib/multipath/bindings` can lead to a wrong assignment of mount points to devices, which can result in file system corruption and data loss.

To avoid this problem, we recommend that you use the default WWID settings for the system root device. You can also use the `alias` option to override the `user_friendly_names` option for the system root device in the `/etc/multipath.conf` file.

For example:

```
multipaths {
    multipath {
        wwid          36006048000028350131253594d303030
        alias          mpatha
    }
    multipath {
        wwid          36006048000028350131253594d303041
        alias          mpathb
    }
    multipath {
        wwid          36006048000028350131253594d303145
        alias          mpathc
    }
    multipath {
        wwid          36006048000028350131253594d303334
        alias          mpathd
    }
}
```

IMPORTANT

We recommend that you do not use aliases for the system root device, because the ability to seamlessly switch off multipathing via the kernel command line is lost because the device name differs.

- **Mounting /var from Another Partition:** The default location of the `user_friendly_names` configuration file is `/var/lib/multipath/bindings`. If the `/var` data is not located on the system root device but mounted from another partition, the `bindings` file is not available when setting up multipathing.

Ensure that the `/var/lib/multipath/bindings` file is available on the system root device and multipath can find it. For example, this can be done as follows:

1. Move the `/var/lib/multipath/bindings` file to `/etc/multipath/bindings`.
2. Set the `bindings_file` option in the `defaults` section of `/etc/multipath.conf` to this new location. For example:

```
defaults {  
    user_friendly_names yes  
    bindings_file "/etc/multipath/bindings"  
}
```

- **Multipath Is in the initrd:** Even if the system root device is not on multipath, it is possible for multipath to be included in the `initrd`. For example, this can happen if the system root device is on LVM. If you use the `user_friendly_names` option and multipath is in the `initrd`, you should boot with the parameter `multipath=off` to avoid problems.

This disables multipath only in the `initrd` during system boots. After the system boots, the `boot.multipath` and `multipathd` boot scripts are able to activate multipathing.

For an example of `multipath.conf` settings, see the `/usr/share/doc/packages/multipath-tools/multipath.conf.synthetic` file.

To enable user-friendly names or to specify aliases:

- 1 In a terminal console, log in as the `root` user.
- 2 Open the `/etc/multipath.conf` file in a text editor.
- 3 (Optional) Modify the location of the `/var/lib/multipath/bindings` file.

The alternate path must be available on the system root device where multipath can find it.

- 3a Move the `/var/lib/multipath/bindings` file to `/etc/multipath/bindings`.

- 3b** Set the `bindings_file` option in the `defaults` section of `/etc/multipath.conf` to this new location. For example:

```
defaults {  
    user_friendly_names yes  
    bindings_file "/etc/multipath/bindings"  
}
```

4 (Optional) Enable user-friendly names:

- 4a** Uncomment the `defaults` section and its ending bracket.

- 4b** Uncomment the `user_friendly_names` option, then change its value from No to Yes.

For example:

```
## Use user friendly names, instead of using WWIDs as names.  
defaults {  
    user_friendly_names yes  
}
```

5 (Optional) Specify your own names for devices by using the `alias` option in the `multipath` section.

For example:

```
## Use alias names, instead of using WWIDs as names.  
multipaths {  
    multipath {  
        wwid          36006048000028350131253594d303030  
        alias         mpatha  
    }  
    multipath {  
        wwid          36006048000028350131253594d303041  
        alias         mpathb  
    }  
    multipath {  
        wwid          36006048000028350131253594d303145  
        alias         mpathc  
    }  
    multipath {  
        wwid          36006048000028350131253594d303334
```

```

        alias                mpathd
    }
}

```

6 Save your changes, then close the file.

Blacklisting Non-Multipathed Devices in `/etc/multipath.conf`

The `/etc/multipath.conf` file should contain a `blacklist` section where all non-multipathed devices are listed. For example, local IDE hard drives and floppy drives are not normally multipathed. If you have single-path devices that `multipath` is trying to manage and you want `multipath` to ignore them, put them in the `blacklist` section to resolve the problem.

NOTE

The keyword `devnode_blacklist` has been deprecated and replaced with the keyword `blacklist`.

For example, to blacklist local devices and all arrays from the `cciss` driver from being managed by `multipath`, the `blacklist` section looks like this:

```

blacklist {
    wwid 26353900f02796769
    devnode "(ram|raw|loop|fd|md|dm-|sr|scd|st|sda) [0-9]*"
    devnode "^hd[a-z] [0-9]*"
    devnode "^cciss!c[0-9]d[0-9].*"
}

```

You can also blacklist only the partitions from a driver instead of the entire array. For example, using the following regular expression blacklists only partitions from the `cciss` driver and not the entire array:

```
^cciss!c[0-9]d[0-9]*[p[0-9]]*
```

After you modify the `/etc/multipath.conf` file, you must run `mkinitrd` to re-create the `initrd` on your system, then reboot in order for the changes to take effect.

After you do this, the local devices should no longer be listed in the multipath maps when you issue the `multipath -ll` command.

Configuring Default Multipath Behavior in `/etc/multipath.conf`

The `/etc/multipath.conf` file should contain a `defaults` section where you can specify default behaviors. If the field is not otherwise specified in a `device` section, the default setting is applied for that SAN configuration.

The following `defaults` section specifies a simple failover policy:

```
defaults {
    multipath_tool    "/sbin/multipath -v0"
    udev_dir          /dev
    polling_interval  10
    default_selector   "round-robin 0"
    default_path_grouping_policy    failover
    default_getuid     "/lib/udev/scsi_id -g -u -s /block/%n"
    default_prio_callout    "/bin/true"
    default_features    "0"
    rr_min_io          100
    failback            immediate
}
```

Configuring Default Settings for zSeries in `/etc/multipath.conf`

Testing of the IBM zSeries device with multipathing has shown that the `dev_loss_tmo` parameter should be set to 90 seconds, and the `fast_io_fail_tmo` parameter should be set to 5 seconds. If you are using zSeries devices, modify the `/etc/multipath.conf` file to specify the values as follows:

```
defaults {
    dev_loss_tmo 90
    fast_io_fail_tmo 5
}
```

The `dev_loss_tmo` parameter sets the number of seconds to wait before marking a multipath link as bad. When the path fails, any current I/O on that failed path fails. The default value varies according to the device driver being used. The valid range of values

is 0 to 600 seconds. To use the driver's internal timeouts, set the value to zero (0) or to any value greater than 600.

The `fast_io_fail_tmo` parameter sets the length of time to wait before failing I/O when a link problem is detected. I/O that reaches the driver fails. If I/O is in a blocked queue, the I/O does not fail until the `dev_loss_tmo` time elapses and the queue is unblocked.

Applying the `/etc/multipath.conf` File Changes

Changes to the `/etc/multipath.conf` file cannot take effect when `multipathd` is running. After you make changes, save and close the file, then do the following to apply the changes:

- 1 Stop the `multipathd` service.

- 2 Clear old multipath bindings by entering

```
/sbin/multipath -F
```

- 3 Create new multipath bindings by entering

```
/sbin/multipath -v2 -l
```

- 4 Start the `multipathd` service.

- 5 Run `mkinitrd` to re-create the `initrd` on your system, then reboot in order for the changes to take effect.

7.5 Enabling and Starting Multipath I/O Services

To start multipath services and enable them to start at reboot:

- 1 Open a terminal console, then log in as the `root` user or equivalent.

- 2 At the terminal console prompt, enter

```
chkconfig multipathd on
```

```
chkconfig boot.multipath on
```

If the `boot.multipath` service does not start automatically on system boot, do the following to start them manually:

1 Open a terminal console, then log in as the `root` user or equivalent.

2 Enter

```
/etc/init.d/boot.multipath start
```

```
/etc/init.d/multipathd start
```

7.6 Configuring Path Failover Policies and Priorities

In a Linux host, when there are multiple paths to a storage controller, each path appears as a separate block device, and results in multiple block devices for single LUN. The Device Mapper Multipath service detects multiple paths with the same LUN ID, and creates a new multipath device with that ID. For example, a host with two HBAs attached to a storage controller with two ports via a single unzoned Fibre Channel switch sees four block devices: `/dev/sda`, `/dev/sdb`, `/dev/sdc`, and `/dev/sdd`. The Device Mapper Multipath service creates a single block device, `/dev/mpath/mpath1` that reroutes I/O through those four underlying block devices.

This section describes how to specify policies for failover and configure priorities for the paths.

- Section 7.6.1, “Configuring the Path Failover Policies” (page 113)
- Section 7.6.2, “Configuring Failover Priorities” (page 113)
- Section 7.6.3, “Using a Script to Set Path Priorities” (page 122)
- Section 7.6.4, “Configuring ALUA (`mpath_prio_alua`)” (page 124)

- Section 7.6.5, “Reporting Target Path Groups” (page 126)

7.6.1 Configuring the Path Failover Policies

Use the `multipath` command with the `-p` option to set the path failover policy:

```
multipath devicename -p policy
```

Replace `policy` with one of the following policy options:

Table 7.5 *Group Policy Options for the `multipath -p` Command*

Policy Option	Description
<code>failover</code>	One path per priority group.
<code>multibus</code>	All paths in one priority group.
<code>group_by_serial</code>	One priority group per detected serial number.
<code>group_by_prio</code>	One priority group per path priority value. Priorities are determined by callout programs specified as a global, per-controller, or per-multipath option in the <code>/etc/multipath.conf</code> configuration file.
<code>group_by_node_name</code>	One priority group per target node name. Target node names are fetched in the <code>/sys/class/fc_transport/target*/node_name</code> location.

7.6.2 Configuring Failover Priorities

You must manually enter the failover priorities for the device in the `/etc/multipath.conf` file. Examples for all settings and options can be found in the `/usr/share/doc/packages/multipath-tools/multipath.conf.annotated` file.

- Section “Understanding Priority Groups and Attributes” (page 114)

- Section “Configuring for Round-Robin Load Balancing” (page 122)
- Section “Configuring for Single Path Failover” (page 122)
- Section “Grouping I/O Paths for Round-Robin Load Balancing” (page 122)

Understanding Priority Groups and Attributes

A *priority group* is a collection of paths that go to the same physical LUN. By default, I/O is distributed in a round-robin fashion across all paths in the group. The `multipath` command automatically creates priority groups for each LUN in the SAN based on the `path_grouping_policy` setting for that SAN. The `multipath` command multiplies the number of paths in a group by the group’s priority to determine which group is the primary. The group with the highest calculated value is the primary. When all paths in the primary group are failed, the priority group with the next highest value becomes active.

A *path priority* is an integer value assigned to a path. The higher the value, the higher the priority is. An external program is used to assign priorities for each path. For a given device, the paths with the same priorities belong to the same priority group.

Table 7.6 *Multipath Attributes*

Multipath Attribute	Description	Values
<code>user_friendly_names</code>	Specifies whether to use IDs or to use the <code>/var/lib/multipath/bindings</code> file to assign a persistent and unique alias to the multipath devices in the form of <code>/dev/mapper/mpathN</code> .	<p>yes: Autogenerate user-friendly names as aliases for the multipath devices instead of the actual ID.</p> <p>no: Default. Use the WWIDs shown in the <code>/dev/disk/by-id/</code> location.</p>

Multipath Attribute	Description	Values
blacklist	Specifies the list of device names to ignore as non-multipathed devices, such as cciss, fd, hd, md, dm, sr, scd, st, ram, raw, loop.	For an example, see Section “Blacklisting Non-Multipathed Devices in <code>/etc/multipath.conf</code> ” (page 109).
blacklist_exceptions	Specifies the list of device names to treat as multipath devices even if they are included in the blacklist.	For an example, see the <code>/usr/share/doc/packages/multipath-tools/multipath.conf.annotated</code> file.
failback	<p>Specifies whether to monitor the failed path recovery, and indicates the timing for group failback after failed paths return to service.</p> <p>When the failed path recovers, the path is added back into the multipath enabled path list based on this setting. Multipath evaluates the priority groups, and changes the active priority group when the priority of the primary path exceeds the secondary group.</p>	<p>immediate: When a path recovers, enable the path immediately.</p> <p>n (> 0): When the path recovers, wait <i>n</i> seconds before enabling the path. Specify an integer value greater than 0.</p> <p>manual: (Default) The failed path is not monitored for recovery. The administrator runs the <code>multipath</code> command to update enabled paths and priority groups.</p> <p>We recommend failback setting of “manual” for multipath in cluster environments in order to prevent multipath failover ping-pong.</p> <pre>failback "manual"</pre>

Multipath Attribute	Description	Values
		<p>IMPORTANT</p> <p>Ensure that you verify the failback setting with your storage system vendor. Different storage systems can require different settings.</p>
getuid	The default program and arguments to call to obtain a unique path identifier. Should be specified with an absolute path.	<p><code>/lib/udev/scsi_id -g -u -s</code></p> <p>This is the default location and arguments.</p> <p>Example:</p> <pre>getuid "/lib/udev/scsi_id -g -u -d /dev/%n"</pre>
no_path_retry	Specifies the behaviors to use on path failure.	<p>n (> 0): Specifies the number of retries until <code>multipath</code> stops the queuing and fails the path. Specify an integer value greater than 0.</p> <p>fail: Specified immediate failure (no queuing).</p> <p>queue : Never stop queuing (queue forever until the path comes alive).</p> <p>We recommend a retry setting of “fail” or “0” in the <code>/etc/multipath.conf</code> file when working in a cluster. This causes the resources to fail over when the connection is lost to storage. Otherwise, the messages queue and the resource failover cannot occur.</p>

Multipath Attribute	Description	Values
<hr/> <div> <div>IMPORTANT</div> <div> <p>Ensure that you verify the retry settings with your storage system vendor. Different storage systems can require different settings.</p> </div> </div> <hr/>		
path_grouping_policy	Specifies the path grouping policy for a multipath device hosted by a given controller.	<p>failover: (Default) One path is assigned per priority group so that only one path at a time is used.</p> <p>multibus: All valid paths are in one priority group. Traffic is load-balanced across all active paths in the group.</p> <p>group_by_prio: One priority group exists for each path priority value. Paths with the same priority are in the same priority group. Priorities are assigned by an external program.</p> <p>group_by_serial: Paths are grouped by the SCSI target serial number (controller node WWN).</p> <p>group_by_node_name: One priority group is assigned per target node name. Target node names are fetched in <code>/sys/class/fc_transport/target*/node_name</code>.</p>
path_checker	Determines the state of the path.	<p>directio: (Default in <code>multipath -tools</code> version 0.4.8 and later) Reads the first sector that has direct I/O. This is useful for DASD devices.</p>

Multipath Attribute	Description	Values
		<p>Logs failure messages in <code>/var/log/messages</code>.</p> <p>readsector0: (Default in <code>multipath-tools</code> version 0.4.7 and earlier) Reads the first sector of the device. Logs failure messages in <code>/var/log/messages</code>.</p> <p>tur: Issues a SCSI test unit ready command to the device. This is the preferred setting if the LUN supports it. On failure, the command does not fill up <code>/var/log/messages</code> with messages.</p> <p>Some SAN vendors provide custom <code>path_checker</code> options:</p> <ul style="list-style-type: none"> • emc_clariion: Queries the EMC Clariion EVPD page 0xC0 to determine the path state. • hp_sw: Checks the path state (Up, Down, or Ghost) for HP storage arrays with Active/Standby firmware. • rdac: Checks the path state for the LSI/Engenio RDAC storage controller.
<code>path_selector</code>	Specifies the path-selector algorithm to use for load balancing.	round-robin 0: (Default) The load-balancing algorithm used to balance traffic across all active paths in a priority group.

Multipath Attribute	Description	Values
		<p>Beginning in SUSE Linux Enterprise Server 11, the following additional I/O balancing options are available:</p> <p>least-pending 0: Provides a least-pending-I/O dynamic load balancing policy for bio based device mapper multipath. This load balancing policy considers the number of unserved requests pending on a path and selects the path with least count of pending service requests.</p> <p>This policy is especially useful when the SAN environment has heterogeneous components. For example, when there is one 8 GB HBA and one 2 GB HBA connected to the same server, the 8 GB HBA could be utilized better with this algorithm.</p> <p>queue-length 0: A dynamic load balancer that balances the number of in-flight I/O on paths similar to the least-pending option.</p> <p>service-time 0: A service-time oriented load balancer that balances I/O on paths according to the latency.</p>
pg_timeout	Specifies path group timeout handling.	NONE (internal default)

Multipath Attribute	Description	Values
<p><code>prio_callout</code></p> <p>Multipath <code>prio_callouts</code> are located in shared libraries in <code>/lib/libmultipath/lib*</code>. By using shared libraries, the callouts are loaded into memory on daemon startup.</p>	<p>Specifies the program and arguments to use to determine the layout of the multipath map.</p> <p>When queried by the <code>multipath</code> command, the specified <code>mpath_prio_*</code> callout program returns the priority for a given path in relation to the entire multipath layout.</p> <p>When it is used with the <code>path_grouping_policy</code> of <code>group_by_prio</code>, all paths with the same priority are grouped into one multipath group. The group with the highest aggregate priority becomes the active group.</p> <p>When all paths in a group fail, the group with the next highest aggregate priority becomes active. Additionally, a failover command (as determined by the hardware handler) might be sent to the target.</p>	<p>If no <code>prio_callout</code> attribute is used, all paths are equal. This is the default.</p> <p>/bin/true: Use this value when the <code>group_by_priority</code> is not being used.</p> <p>The <code>prioritizer</code> programs generate path priorities when queried by the <code>multipath</code> command. The program names must begin with <code>mpath_prio_</code> and are named by the device type or balancing method used. Current prioritizer programs include the following:</p> <p><code>mpath_prio_alua %n</code>: Generates path priorities based on the SCSI-3 ALUA settings.</p> <p><code>mpath_prio_balance_units</code>: Generates the same priority for all paths.</p> <p><code>mpath_prio_emc %n</code>: Generates the path priority for EMC arrays.</p> <p><code>mpath_prio_hds_modular %b</code>: Generates the path priority for Hitachi HDS Modular storage arrays.</p> <p><code>mpath_prio_hp_sw %n</code>: Generates the path priority for Compaq/HP controller in active/standby mode.</p> <p><code>mpath_prio_netapp %n</code>: Generates the path priority for NetApp arrays.</p>

Multipath Attribute	Description	Values
	<p>The <code>mpath_prio_*</code> program can also be a custom script created by a vendor or administrator for a specified setup.</p> <p>A <code>%n</code> in the command line expands to the device name in the <code>/dev</code> directory.</p> <p>A <code>%b</code> expands to the device number in <i>major:minor</i> format in the <code>/dev</code> directory.</p> <p>A <code>%d</code> expands to the device ID in the <code>/dev/disk/by-id</code> directory.</p> <p>If devices are hot-pluggable, use the <code>%d</code> flag instead of <code>%n</code>. This addresses the short time that elapses between the time when devices are available and when <code>udev</code> creates the device nodes.</p>	<p><code>mpath_prio_random %n</code>: Generates a random priority for each path.</p> <p><code>mpath_prio_rdac %n</code>: Generates the path priority for LSI/Engenio RDAC controller.</p> <p><code>mpath_prio_tpc %n</code>: You can optionally use a script created by a vendor or administrator that gets the priorities from a file where you specify priorities to use for each path.</p> <p><code>mpath_prio_spec.sh %n</code>: Provides the path of a user-created script that generates the priorities for multipathing based on information contained in a second data file. (This path and filename are provided as an example. Specify the location of your script instead.) The script can be created by a vendor or administrator. The script's target file identifies each path for all multipathed devices and specifies a priority for each path. For an example, see Section 7.6.3, "Using a Script to Set Path Priorities" (page 122).</p>
<code>rr_min_io</code>	Specifies the number of I/O transactions to route to a path before switching to the next path in the same path group, as	<p><code>n (>0)</code>: Specify an integer value greater than 0.</p> <p><code>1000</code>: Default.</p>

Multipath Attribute	Description	Values
	determined by the specified algorithm in the <code>path_selector</code> setting.	
<code>rr_weight</code>	Specifies the weighting method to use for paths.	uniform: Default. All paths have the same round-robin weights. priorities: Each path's weight is determined by the path's priority times the <code>rr_min_io</code> setting.

Configuring for Round-Robin Load Balancing

All paths are active. I/O is configured for some number of seconds or some number of I/O transactions before moving to the next open path in the sequence.

Configuring for Single Path Failover

A single path with the highest priority (lowest value setting) is active for traffic. Other paths are available for failover, but are not used unless failover occurs.

Grouping I/O Paths for Round-Robin Load Balancing

Multiple paths with the same priority fall into the active group. When all paths in that group fail, the device fails over to the next highest priority group. All paths in the group share the traffic load in a round-robin load balancing fashion.

7.6.3 Using a Script to Set Path Priorities

You can create a script that interacts with Device Mapper Multipath (DM-MPIO) to provide priorities for paths to the LUN when set as a resource for the `prio_callout` setting.

First, set up a text file that lists information about each device and the priority values you want to assign to each path. For example, name the file `/usr/local/etc/primary-paths`. Enter one line for each path in the following format:

```
host_wwpn target_wwpn scsi_id priority_value
```

Return a priority value for each path on the device. Ensure that the variable `FILE_PRIMARY_PATHS` resolves to a real file with appropriate data (host wwpn, target wwpn, scsi_id and priority value) for each device.

The contents of the `primary-paths` file for a single LUN with eight paths each might look like this:

```
0x10000000c95eb4 0x200200a0b8122c6e 2:0:0:0 sdb
3600a0b8000122c6d00000000453174fc 50
```

```
0x10000000c95eb4 0x200200a0b8122c6e 2:0:0:1 sdc
3600a0b80000fd6320000000045317563 2
```

```
0x10000000c95eb4 0x200200a0b8122c6e 2:0:0:2 sdd
3600a0b8000122c6d0000000345317524 50
```

```
0x10000000c95eb4 0x200200a0b8122c6e 2:0:0:3 sde
3600a0b80000fd6320000000245317593 2
```

```
0x10000000c95eb4 0x200300a0b8122c6e 2:0:1:0 sdi
3600a0b8000122c6d00000000453174fc 5
```

```
0x10000000c95eb4 0x200300a0b8122c6e 2:0:1:1 sdj
3600a0b80000fd6320000000045317563 51
```

```
0x10000000c95eb4 0x200300a0b8122c6e 2:0:1:2 sdk
3600a0b8000122c6d0000000345317524 5
```

```
0x10000000c95eb4 0x200300a0b8122c6e 2:0:1:3 sdl
3600a0b80000fd6320000000245317593 51
```

To continue the example mentioned in Table 7.6, “Multipath Attributes” (page 114), create a script named `/usr/local/sbin/path_prio.sh`. You can use any path and filename. The script does the following:

- On query from multipath, grep the device and its path from the `/usr/local/etc/primary-paths` file.
- Return to multipath the priority value in the last column for that entry in the file.

7.6.4 Configuring ALUA (mpath_prio_alua)

The `mpath_prio_alua(8)` command is used as a priority callout for the Linux `multipath(8)` command. It returns a number that is used by DM-MPIO to group SCSI devices with the same priority together. This path priority tool is based on ALUA (Asynchronous Logical Unit Access).

- Section “Syntax” (page 124)
- Section “Prerequisite” (page 124)
- Section “Options” (page 124)
- Section “Return Values” (page 125)

Syntax

```
mpath_prio_alua [-d directory] [-h] [-v] [-V] device [device...]
```

Prerequisite

SCSI devices.

Options

`-d directory`

Specifies the Linux directory path where the listed device node names can be found. The default directory is `/dev`. When you use this option, specify the device node name only (such as `sda`) for the device or devices you want to manage.

`-h`

Displays help for this command, then exits.

-v

Turns on verbose output to display status in human-readable format. Output includes information about which port group the specified device is in and its current state.

-V

Displays the version number of this tool, then exits.

device [*device*...]

Specifies the SCSI device (or multiple devices) that you want to manage. The device must be a SCSI device that supports the Report Target Port Groups (`sg_rtpg(8)`) command. Use one of the following formats for the device node name:

- The full Linux directory path, such as `/dev/sda`. Do not use with the `-d` option.
- The device node name only, such as `sda`. Specify the directory path by using the `-d` option.
- The major and minor number of the device separated by a colon (`:`) with no spaces, such as `8:0`. This creates a temporary device node in the `/dev` directory with a name in the format of `tmpdev-<major>:<minor>-<pid>`. For example, `/dev/tmpdev-8:0-<pid>`.

Return Values

On success, returns a value of 0 and the priority value for the group. Table 7.7, “ALUA Priorities for Device Mapper Multipath” (page 125) shows the priority values returned by the `mpath_prio_alua` command.

Table 7.7 *ALUA Priorities for Device Mapper Multipath*

Priority Value	Description
50	The device is in the active, optimized group.
10	The device is in an active but non-optimized group.
1	The device is in the standby group.
0	All other groups.

Values are widely spaced because of the way the `multipath` command handles them. It multiplies the number of paths in a group with the priority value for the group, then selects the group with the highest result. For example, if a non-optimized path group has six paths ($6 \times 10 = 60$) and the optimized path group has a single path ($1 \times 50 = 50$), the non-optimized group has the highest score, so `multipath` chooses the non-optimized group. Traffic to the device uses all six paths in the group in a round-robin fashion.

On failure, returns a value of 1 to 5 indicating the cause for the command's failure. For information, see the man page for `mpath_prio_alua`.

7.6.5 Reporting Target Path Groups

Use the SCSI Report Target Port Groups (`sg_rtpg(8)`) command. For information, see the man page for `sg_rtpg(8)`.

7.7 Configuring Multipath I/O for the Root Device

Device Mapper Multipath I/O (DM-MPIO) is available and supported for `/boot` and `/root` in SUSE Linux Enterprise Server 11. In addition, the YaST partitioner in the YaST2 installer supports enabling multipath during the install.

- Section 7.7.1, “Enabling Multipath I/O at Install Time” (page 126)
- Section 7.7.2, “Enabling Multipath I/O for an Existing Root Device” (page 130)
- Section 7.7.3, “Disabling Multipath I/O on the Root Device” (page 131)

7.7.1 Enabling Multipath I/O at Install Time

The multipath software must be running at install time if you want to install the operating system on a multipath device. The `multipathd` daemon is not automatically active during the system installation. You can start it by using the *Configure Multipath* option in the YaST partitioner.

- Section “Enabling Multipath I/O at Install Time on an Active/Active Multipath Storage LUN” (page 127)
- Section “Enabling Multipath I/O at Install Time on an Active/Passive Multipath Storage LUN” (page 127)

Enabling Multipath I/O at Install Time on an Active/Active Multipath Storage LUN

- 1 During the install on the YaST2 Installation Settings page, click on *Partitioning* to open the YaST partitioner.
- 2 Select *Custom Partitioning (for experts)*.
- 3 Select the *Hard Disks* main icon, click the *Configure* button, then select *Configure Multipath*.
- 4 Start multipath.

YaST2 starts to rescan the disks and shows available multipath devices (such as `/dev/mapper/3600a0b80000f4593000012ae4ab0ae65`). This is the device that should be used for all further processing.

- 5 Click *Next* to continue with the installation.

Enabling Multipath I/O at Install Time on an Active/Passive Multipath Storage LUN

The `multipathd` daemon is not automatically active during the system installation. You can start it by using the *Configure Multipath* option in the YaST partitioner.

To enable multipath I/O at install time for an active/passive multipath storage LUN:

- 1 During the install on the YaST2 Installation Settings page, click on *Partitioning* to open the YaST partitioner.
- 2 Select *Custom Partitioning (for experts)*.

3 Select the *Hard Disks* main icon, click the *Configure* button, then select *Configure Multipath*.

4 Start multipath.

YaST2 starts to rescan the disks and shows available multipath devices (such as `/dev/mapper/3600a0b80000f4593000012ae4ab0ae65`). This is the device that should be used for all further processing. Write down the device path and UUID; you need it later.

5 Click *Next* to continue with the installation.

6 After all settings are done and the installation finished, YaST2 starts to write the boot loader information, and displays a countdown to restart the system. Stop the counter by clicking the *Stop* button and press CTRL+ALT+F5 to access a console.

7 Use the console to determine if a passive path was entered in the `/boot/grub/device.map` file for the `hd0` entry.

This is necessary because the installation does not distinguish between active and passive paths.

7a Mount the root device to `/mnt` by entering

```
mount /dev/mapper/UUID_part2 /mnt
```

For example, enter

```
mount /dev/mapper/3600a0b80000f4593000012ae4ab0ae65_part2 /mnt
```

7b Mount the boot device to `/mnt/boot` by entering

```
mount /dev/mapper/UUID_part1 /mnt/boot
```

For example, enter

```
mount /dev/mapper/3600a0b80000f4593000012ae4ab0ae65_part1 /mnt/boot
```

7c Open `/mnt/boot/grub/device.map` file by entering

```
less /mnt/boot/grub/device.map
```

7d In the `/mnt/boot/grub/device.map` file, determine if the `hd0` entry points to a passive path, then do one of the following:

- **Active path:** No action is needed; skip Step 8 (page 129) and continue with Step 9 (page 130).
- **Passive path:** The configuration must be changed and the boot loader must be reinstalled. Continue with Step 8 (page 129).

8 If the `hd0` entry points to a passive path, change the configuration and reinstall the boot loader:

8a At the console, enter the following commands at the console prompt:

```
mount -o bind /dev /mnt/dev
mount -o bind /sys /mnt/sys
mount -o bind /proc /mnt/proc
chroot
```

8b At the console, run `multipath -ll`, then check the output to find the active path.

Passive paths are flagged as `ghost`.

8c In the `/mnt/boot/grub/device.map` file, change the `hd0` entry to an active path, save the changes, and close the file.

8d In case the selection was to boot from MBR, `/etc/grub.conf` should look like the following:

```
setup --stage2=/boot/grub/stage2 (hd0) (hd0,0)
quit
```

8e Reinstall the boot loader by entering

```
grub < /etc/grub.conf
```

8f Enter the following commands:

```
exit  
  
umount /mnt/*  
  
umount /mnt
```

9 Return to the YaST graphical environment by pressing CTRL+ALT+F7.

10 Click *OK* to continue with the installation reboot.

7.7.2 Enabling Multipath I/O for an Existing Root Device

- 1** Install Linux with only a single path active, preferably one where the `by-id` symlinks are listed in the partitioner.
- 2** Mount the devices by using the `/dev/disk/by-id` path used during the install.
- 3** After installation, add `dm-multipath` to `/etc/sysconfig/kernel:INITRD_MODULES`.
- 4** For System Z, before running `mkinitrd`, edit the `/etc/zipl.conf` file to change the `by-path` information in `zipl.conf` with the same `by-id` information that was used in the `/etc/fstab`.
- 5** Re-run `/sbin/mkinitrd` to update the `initrd` image.
- 6** For System Z, after running `mkinitrd`, run `zipl`.
- 7** Reboot the server.

7.7.3 Disabling Multipath I/O on the Root Device

- Add `multipath=off` to the kernel command line.

This affects only the root device. All other devices are not affected.

7.8 Configuring Multipath I/O for an Existing Software RAID

Ideally, you should configure multipathing for devices before you use them as components of a software RAID device. If you add multipathing after creating any software RAID devices, the DM-MPIO service might be starting after the `multipath` service on reboot, which makes multipathing appear not to be available for RAIDs. You can use the procedure in this section to get multipathing running for a previously existing software RAID.

For example, you might need to configure multipathing for devices in a software RAID under the following circumstances:

- If you create a new software RAID as part of the Partitioning settings during a new install or upgrade.
- If you did not configure the devices for multipathing before using them in the software RAID as a member device or spare.
- If you grow your system by adding new HBA adapters to the server or expanding the storage subsystem in your SAN.

NOTE

The following instructions assume the software RAID device is `/dev/mapper/mpath0`, which is its device name as recognized by the kernel. Ensure that you modify the instructions for the device name of your software RAID.

- 1 Open a terminal console, then log in as the `root` user or equivalent.

Except where otherwise directed, use this console to enter the commands in the following steps.

- 2 If any software RAID devices are currently mounted or running, enter the following commands for each device to dismount the device and stop it.

```
umount /dev/mapper/mpath0
```

```
mdadm --misc --stop /dev/mapper/mpath0
```

- 3 Stop the `boot.md` service by entering

```
/etc/init.d/boot.md stop
```

- 4 Start the `boot.multipath` and `multipathd` services by entering the following commands:

```
/etc/init.d/boot.multipath start
```

```
/etc/init.s/multipathd start
```

- 5 After the multipathing services are started, verify that the software RAID's component devices are listed in the `/dev/disk/by-id` directory. Do one of the following:

- **Devices Are Listed:** The device names should now have symbolic links to their Device Mapper Multipath device names, such as `/dev/dm-1`.
- **Devices Are Not Listed:** Force the multipath service to recognize them by flushing and rediscovering the devices.

To do this, enter the following commands:

```
multipath -F
```

```
multipath -v0
```

The devices should now be listed in `/dev/disk/by-id`, and have symbolic links to their Device Mapper Multipath device names. For example:

```
lrwxrwxrwx 1 root root 10 Jun 15 09:36 scsi-mpath1 -> ../../dm-1
```

6 Restart the `boot.md` service and the RAID device by entering

```
/etc/init.d/boot.md start
```

7 Check the status of the software RAID by entering

```
mdadm --detail /dev/mapper/mpath0
```

The RAID's component devices should match their Device Mapper Multipath device names that are listed as the symbolic links of devices in the `/dev/disk/by-id` directory.

8 Make a new `initrd` to ensure that the Device Mapper Multipath services are loaded before the RAID services on reboot. Running `mkinitrd` is needed only if the root (`/`) device or any parts of it (such as `/var`, `/etc`, `/log`) are on the SAN and `multipath` is needed to boot.

Enter

```
mkinitrd -f multipath
```

9 Reboot the server to apply these post-install configuration settings.

10 Verify that the software RAID array comes up properly on top of the multipathed devices by checking the RAID status. Enter

```
mdadm --detail /dev/mapper/mpath0
```

For example:

```
Number Major Minor RaidDevice State
0 253 0 0 active sync /dev/dm-0
1 253 1 1 active sync /dev/dm-1
2 253 2 2 active sync /dev/dm-2
```

7.9 Scanning for New Devices without Rebooting

If your system has already been configured for multipathing and you later need to add more storage to the SAN, you can use the `rescan-scsi-bus.sh` script to scan for the new devices. By default, this script scans all HBAs with typical LUN ranges.

Syntax

```
rescan-scsi-bus.sh [options] [host [host ...]]
```

You can specify hosts on the command line (deprecated), or use the `--hosts=LIST` option (recommended).

Options

For most storage subsystems, the script can be run successfully without options. However, some special cases might need to use one or more of the following parameters for the `rescan-scsi-bus.sh` script:

Option	Description
<code>-l</code>	Activates scanning for LUNs 0-7. [Default: 0]
<code>-L NUM</code>	Activates scanning for LUNs 0 to NUM. [Default: 0]
<code>-w</code>	Scans for target device IDs 0 to 15. [Default: 0 to 7]
<code>-c</code>	Enables scanning of channels 0 or 1. [Default: 0]

Option	Description
<code>-r</code> <code>--remove</code>	Enables removing of devices. [Default: Disabled]
<code>-i</code> <code>--issueLip</code>	Issues a Fibre Channel LIP reset. [Default: Disabled]
<code>--forcerescan</code>	Rescans existing devices.
<code>--forceremove</code>	Removes and re-adds every device.
<hr/> WARNING <hr/> Use with caution, this option is dangerous. <hr/>	
<code>--nooptscan</code>	Don't stop looking for LUNs if 0 is not found.
<code>--color</code>	Use colored prefixes OLD/NEW/DEL.
<code>--hosts=LIST</code>	Scans only hosts in LIST, where LIST is a comma-separated list of single values and ranges. No spaces are allowed.
	<code>--hosts=A[-B] [, C[-D]]</code>
<code>--channels=LIST</code>	Scans only channels in LIST, where LIST is a comma-separated list of single values and ranges. No spaces are allowed.
	<code>--channels=A[-B] [, C[-D]]</code>
<code>--ids=LIST</code>	Scans only target IDs in LIST, where LIST is a comma-separated list of single values and ranges. No spaces are allowed.

Option	Description
	<code>--ids=A[-B] [,C[-D]]</code>
<code>--luns=LIST</code>	Scans only LUNs in LIST, where LIST is a comma-separated list of single values and ranges. No spaces are allowed.
	<code>--luns=A[-B] [,C[-D]]</code>

Procedure

Use the following procedure to scan the devices and make them available to multipathing without rebooting the system.

- 1 On the storage subsystem, use the vendor's tools to allocate the device and update its access control settings to allow the Linux system access to the new storage. Refer to the vendor's documentation for details.
- 2 Scan all targets for a host to make its new device known to the middle layer of the Linux kernel's SCSI subsystem. At a terminal console prompt, enter

```
rescan-scsi-bus.sh [options]
```

- 3 Check for scanning progress in the system log (the `/var/log/messages` file). At a terminal console prompt, enter

```
tail -30 /var/log/messages
```

This command displays the last 30 lines of the log. For example:

```
# tail -30 /var/log/messages
. . .
Feb 14 01:03 kernel: SCSI device sde: 81920000
Feb 14 01:03 kernel: SCSI device sdf: 81920000
Feb 14 01:03 multipathd: sde: path checker registered
Feb 14 01:03 multipathd: sdf: path checker registered
Feb 14 01:03 multipathd: mpath4: event checker started
Feb 14 01:03 multipathd: mpath5: event checker started
```

```
Feb 14 01:03:multipathd: mpath4: remaining active paths: 1
Feb 14 01:03 multipathd: mpath5: remaining active paths: 1
```

- 4 Repeat Step 2 (page 136) through Step 3 (page 136) to add paths through other HBA adapters on the Linux system that are connected to the new device.
- 5 Run the `multipath` command to recognize the devices for DM-MPIO configuration. At a terminal console prompt, enter

```
multipath
```

You can now configure the new device for multipathing.

7.10 Scanning for New Partitioned Devices without Rebooting

Use the example in this section to detect a newly added multipathed LUN without rebooting.

- 1 Open a terminal console, then log in as the `root` user.
- 2 Scan all targets for a host to make its new device known to the middle layer of the Linux kernel's SCSI subsystem. At a terminal console prompt, enter

```
rescan-scsi-bus.sh [options]
```

For syntax and options information for the `rescan-scsi-bus.sh` script, see Section 7.9, “Scanning for New Devices without Rebooting” (page 134).

- 3 Verify that the device is seen (the link has a new time stamp) by entering

```
ls -lrt /dev/dm-*
```

- 4 Verify the new WWN of the device appears in the log by entering

```
tail -33 /var/log/messages
```

- 5 Use a text editor to add a new alias definition for the device in the `/etc/multipath.conf` file, such as `oradata3`.

- 6 Create a partition table for the device by entering

```
fdisk /dev/dm-8
```

- 7 Trigger udev by entering

```
echo 'add' > /sys/block/dm-8/uevent
```

This generates the device-mapper devices for the partitions on `dm-8`.

- 8 Create a file system and label for the new partition by entering

```
mke2fs -j /dev/dm-9
```

```
tune2fs -L oradata3 /dev/dm-9
```

- 9 Restart DM-MPIO to let it read the aliases by entering

```
/etc/init.d/multipathd restart
```

- 10 Verify that the device is recognized by `multipathd` by entering

```
multipath -ll
```

- 11 Use a text editor to add a mount entry in the `/etc/fstab` file.

At this point, the alias you created in Step 5 (page 138) is not yet in the `/dev/disk/by-label` directory. Add the mount entry the `/dev/dm-9` path, then change the entry before the next time you reboot to

```
LABEL=oradata3
```

- 12 Create a directory to use as the mount point, then mount the device by entering

```
md /oradata3
```

```
mount /oradata3
```

7.11 Viewing Multipath I/O Status

Querying the multipath I/O status outputs the current status of the multipath maps.

The `multipath -l` option displays the current path status as of the last time that the path checker was run. It does not run the path checker.

The `multipath -ll` option runs the path checker, updates the path information, then displays the current status information. This option always displays the latest information about the path status.

- At a terminal console prompt, enter

```
multipath -ll
```

This displays information for each multipathed device. For example:

```
3600601607cf30e00184589a37a31d911  
[size=127 GB][features="0"][hwhandler="1 emc"]
```

```
\_ round-robin 0 [active][first]  
  \_ 1:0:1:2 sdav 66:240 [ready ][active]  
  \_ 0:0:1:2 sdr  65:16  [ready ][active]
```

```
\_ round-robin 0 [enabled]  
  \_ 1:0:0:2 sdag 66:0   [ready ][active]  
  \_ 0:0:0:2 sdc  8:32   [ready ][active]
```

For each device, it shows the device's ID, size, features, and hardware handlers.

Paths to the device are automatically grouped into priority groups on device discovery. Only one priority group is active at a time. For an active/active configuration, all paths are in the same group. For an active/passive configuration, the passive paths are placed in separate priority groups.

The following information is displayed for each group:

- Scheduling policy used to balance I/O within the group, such as round-robin
- Whether the group is active, disabled, or enabled
- Whether the group is the first (highest priority) group
- Paths contained within the group

The following information is displayed for each path:

- The physical address as *host:bus:target:lun*, such as 1:0:1:2
- Device node name, such as *sda*
- Major:minor numbers
- Status of the device

7.12 Managing I/O in Error Situations

You might need to configure multipathing to queue I/O if all paths fail concurrently by enabling `queue_if_no_path`. Otherwise, I/O fails immediately if all paths are gone. In certain scenarios, where the driver, the HBA, or the fabric experience spurious errors, DM-MPIO should be configured to queue all I/O where those errors lead to a loss of all paths, and never propagate errors upward.

When you use multipathed devices in a cluster, you might choose to disable `queue_if_no_path`. This automatically fails the path instead of queuing the I/O, and escalates the I/O error to cause a failover of the cluster resources.

Because enabling `queue_if_no_path` leads to I/O being queued indefinitely unless a path is reinstated, ensure that `multipathd` is running and works for your scenario. Otherwise, I/O might be stalled indefinitely on the affected multipathed device until reboot or until you manually return to failover instead of queuing.

To test the scenario:

- 1 In a terminal console, log in as the `root` user.
- 2 Activate queuing instead of failover for the device I/O by entering:

```
dmsetup message device_ID 0 queue_if_no_path
```

Replace the *device_ID* with the ID for your device. The 0 value represents the sector and is used when sector information is not needed.

For example, enter:

```
dmsetup message 3600601607cf30e00184589a37a31d911 0 queue_if_no_path
```

3 Return to failover for the device I/O by entering:

```
dmsetup message device_ID 0 fail_if_no_path
```

This command immediately causes all queued I/O to fail.

Replace the *device_ID* with the ID for your device. For example, enter:

```
dmsetup message 3600601607cf30e00184589a37a31d911 0 fail_if_no_path
```

To set up queuing I/O for scenarios where all paths fail:

- 1** In a terminal console, log in as the `root` user.
- 2** Open the `/etc/multipath.conf` file in a text editor.
- 3** Uncomment the defaults section and its ending bracket, then add the `default_features` setting, as follows:

```
defaults {  
    default_features "1 queue_if_no_path"  
}
```

- 4** After you modify the `/etc/multipath.conf` file, you must run `mkinitrd` to re-create the `initrd` on your system, then reboot in order for the changes to take effect.
- 5** When you are ready to return over to failover for the device I/O, enter:

```
dmsetup message mapname 0 fail_if_no_path
```

Replace the *mapname* with the mapped alias name or the device ID for the device. The 0 value represents the sector and is used when sector information is not needed.

This command immediately causes all queued I/O to fail and propagates the error to the calling application.

7.13 Resolving Stalled I/O

If all paths fail concurrently and I/O is queued and stalled, do the following:

- 1 Enter the following command at a terminal console prompt:

```
dmsetup message mapname 0 fail_if_no_path
```

Replace *mapname* with the correct device ID or mapped alias name for the device. The 0 value represents the sector and is used when sector information is not needed.

This command immediately causes all queued I/O to fail and propagates the error to the calling application.

- 2 Reactivate queueing by entering the following command at a terminal console prompt:

```
dmsetup message mapname 0 queue_if_no_path
```

7.14 Troubleshooting MPIO

For information about troubleshooting multipath I/O issues on SUSE Linux Enterprise Server, see the following Technical Information Documents (TIDs) in the Novell Support Knowledgebase:

- *Troubleshooting SLES Multipathing (MPIO) Problems (TID 3231766)* [http://www.novell.com/support/search.do?cmd=displayKC&docType=kc&externalId=3231766&sliceId=SAL_Public]
- *DM MPIO Device Blacklisting Not Honored in multipath.conf (TID3029706)* [<http://www.novell.com/support/search.do?cmd=displayKC>

`&docType=kc&externalId=3029706&sliceId=SAL_Public
&dialogID=57872426&stateId=0%200%2057878058]`

- *Troubleshooting SCSI (LUN) Scanning Issues (TID 3955167)* [http://www.novell.com/support/search.do?cmd=displayKC&docType=kc&externalId=3955167&sliceId=SAL_Public&dialogID=57868704&stateId=0%200%2057878206]

7.15 What's Next

If you want to use software RAIDs, create and configure them before you create file systems on the devices. For information, see the following:

- Chapter 8, *Software RAID Configuration* (page 145)
- Chapter 10, *Managing Software RAIDs 6 and 10 with mdadm* (page 161)

Software RAID Configuration

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

IMPORTANT

Software RAID is not supported underneath clustered file systems such as OCFS2, because RAID does not support concurrent activation. If you want RAID for OCFS2, you need the RAID to be handled by the storage subsystem.

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

- Section 8.1, “Understanding RAID Levels” (page 146)
- Section 8.2, “Soft RAID Configuration with YaST” (page 148)
- Section 8.3, “Troubleshooting Software RAIDs” (page 150)

- Section 8.4, “For More Information” (page 151)

8.1 Understanding RAID Levels

This section describes common RAID levels 0, 1, 2, 3, 4, 5, and nested RAID levels.

- Section 8.1.1, “RAID 0” (page 146)
- Section 8.1.2, “RAID 1” (page 146)
- Section 8.1.3, “RAID 2 and RAID 3” (page 147)
- Section 8.1.4, “RAID 4” (page 147)
- Section 8.1.5, “RAID 5” (page 147)
- Section 8.1.6, “Nested RAID Levels” (page 147)

8.1.1 RAID 0

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

8.1.2 RAID 1

This level provides adequate security for your data, because the data is copied to another hard disk 1:1. This is known as *hard disk mirroring*. If a disk is destroyed, a copy of its contents is available on another mirrored disk. All disks except one could be damaged without endangering your data. However, if damage is not detected, damaged data might be mirrored to the correct disk and the data is corrupted that way. The writing performance suffers a little in the copying process compared to when using single disk access (10 to 20 percent slower), but read access is significantly faster in comparison to any one of the normal physical hard disks, because the data is duplicated so can be

scanned in parallel. RAID 1 generally provides nearly twice the read transaction rate of single disks and almost the same write transaction rate as single disks.

8.1.3 RAID 2 and RAID 3

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

8.1.4 RAID 4

Level 4 provides block-level striping just like Level 0 combined with a dedicated parity disk. If a data disk fails, the parity data is used to create a replacement disk. However, the parity disk might create a bottleneck for write access. Nevertheless, Level 4 is sometimes used.

8.1.5 RAID 5

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

8.1.6 Nested RAID Levels

Several other RAID levels have been developed, such as RAIDn, RAID 10, RAID 0+1, RAID 30, and RAID 50. Some of them being proprietary implementations created by hardware vendors. These levels are not very widespread, and are not explained here.

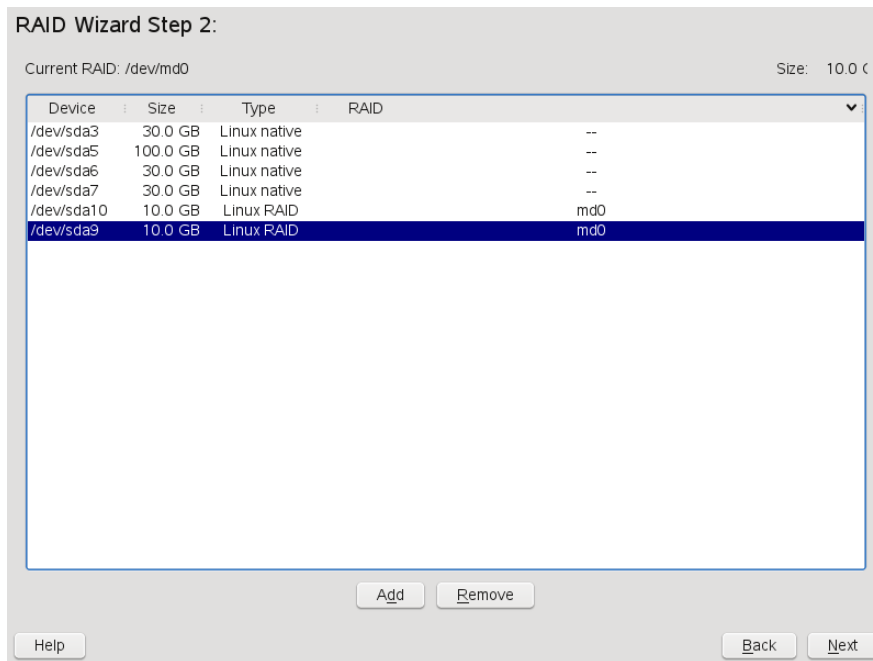
8.2 Soft RAID Configuration with YaST

The YaST soft RAID configuration can be reached from the YaST Expert Partitioner. This partitioning tool enables you to edit and delete existing partitions and create new ones that should be used with soft RAID.

You can create RAID partitions by first clicking *Create > Do not format* then selecting *0xFD Linux RAID* as the partition identifier. For RAID 0 and RAID 1, at least two partitions are needed—for RAID 1, usually exactly two and no more. If RAID 5 is used, at least three partitions are required. It is recommended to use only partitions of the same size because each segment can contribute only the same amount of space as the smallest sized partition. The RAID partitions should be stored on different hard disks to decrease the risk of losing data if one is defective (RAID 1 and 5) and to optimize the performance of RAID 0. After creating all the partitions to use with RAID, click *RAID > Create RAID* to start the RAID configuration.

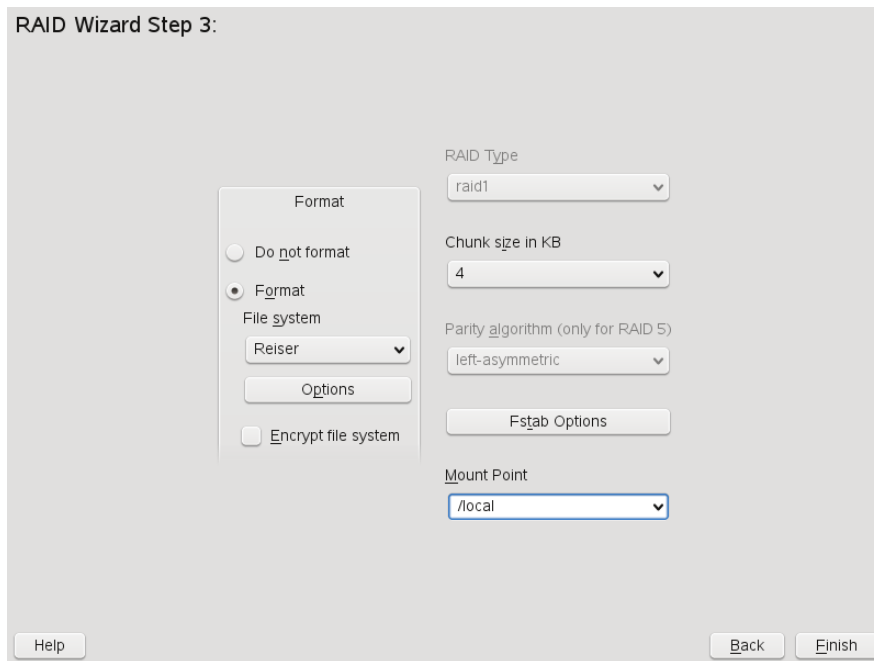
In the next dialog, choose among RAID levels 0, 1, and 5, then click *Next*. The following dialog (see Figure 8.1, “RAID Partitions” (page 149)) lists all partitions with either the *Linux RAID* or *Linux native* type. No swap or DOS partitions are shown. If a partition is already assigned to a RAID volume, the name of the RAID device (for example, `/dev/md0`) is shown in the list. Unassigned partitions are indicated with “--”.

Figure 8.1 RAID Partitions



To add a previously unassigned partition to the selected RAID volume, first select the partition then click *Add*. At this point, the name of the RAID device is displayed next to the selected partition. Assign all partitions reserved for RAID. Otherwise, the space on the partition remains unused. After assigning all partitions, click *Next* to proceed to the settings dialog where you can fine-tune the performance (see Figure 8.2, “File System Settings” (page 150)).

Figure 8.2 File System Settings



RAID Wizard Step 3:

Format

☐ Do not format

☒ Format

File system

Reiser

Options

☐ Encrypt file system

RAID Type

raid1

Chunk size in KB

4

Parity algorithm (only for RAID 5)

left-asymmetric

Estab Options

Mount Point

/local

Help Back Finish

As with conventional partitioning, set the file system to use as well as encryption and the mount point for the RAID volume. After completing the configuration with *Finish*, see the `/dev/md0` device and others indicated with *RAID* in the Expert Partitioner.

8.3 Troubleshooting Software RAIDs

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

Although you can access all data during the rebuild, you might encounter some performance issues until the RAID has been fully rebuilt.

8.4 For More Information

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

- *Linux RAID wiki* [https://raid.wiki.kernel.org/index.php/Linux_Raid]
- *The Software RAID HOWTO* in the `/usr/share/doc/packages/mdadm/Software-RAID.HOWTO.html` file

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

Configuring Software RAID for the Root Partition

In SUSE Linux Enterprise Server 11, the Device Mapper RAID tool has been integrated into the YaST Partitioner. You can use the partitioner at install time to create a software RAID for the system device that contains your root (/) partition.

- Section 9.1, “Prerequisites for the Software RAID” (page 153)
- Section 9.2, “Enabling iSCSI Initiator Support at Install Time” (page 154)
- Section 9.3, “Enabling Multipath I/O Support at Install Time” (page 155)
- Section 9.4, “Creating a Software RAID Device for the Root (/) Partition” (page 155)

9.1 Prerequisites for the Software RAID

Ensure that your configuration meets the following requirements:

- You need two or more hard drives, depending on the type of software RAID you plan to create.
 - **RAID 0 (Striping):** RAID 0 requires two or more devices. RAID 0 offers no fault tolerance benefits, and it is not recommended for the system device.
 - **RAID 1 (Mirroring):** RAID 1 requires two devices.

- **RAID 5 (Redundant Striping):** RAID 5 requires three or more devices.
- The hard drives should be similarly sized. The RAID assumes the size of the smallest drive.
- The block storage devices can be any combination of local (in or directly attached to the machine), Fibre Channel storage subsystems, or iSCSI storage subsystems.
- If you are using hardware RAID devices, do not attempt to run software RAIDs on top of it.
- If you are using iSCSI target devices, enable the iSCSI initiator support before you create the RAID device.
- If your storage subsystem provides multiple I/O paths between the server and its directly attached local devices, Fibre Channel devices, or iSCSI devices that you want to use in the software RAID, you must enable the multipath support before you create the RAID device.

9.2 Enabling iSCSI Initiator Support at Install Time

If there are iSCSI target devices that you want to use for the root (/) partition, you must enable the iSCSI Initiator software to make those devices available to you before you create the software RAID device.

- 1 Proceed with the YaST install of SUSE Linux Enterprise 11 until you reach the Installation Settings page.
- 2 Click *Partitioning* to open the Preparing Hard Disk page, click *Custom Partitioning (for experts)*, then click *Next*.
- 3 On the Expert Partitioner page, expand *Hard Disks* in the *System View* panel to view the default proposal.
- 4 On the *Hard Disks* page, select *ConfigureConfigure iSCSI*, then click *Continue* when prompted to continue with initializing the iSCSI initiator configuration.

9.3 Enabling Multipath I/O Support at Install Time

If there are multiple I/O paths to the devices you want to use to create a software RAID device for the root (/) partition, you must enable multipath support before you create the software RAID device.

- 1 Proceed with the YaST install of SUSE Linux Enterprise 11 until you reach the Installation Settings page.
- 2 Click *Partitioning* to open the Preparing Hard Disk page, click *Custom Partitioning (for experts)*, then click *Next*.
- 3 On the Expert Partitioner page, expand *Hard Disks* in the *System View* panel to view the default proposal.
- 4 On the *Hard Disks* page, select *ConfigureConfigure Multipath*, then click *Yes* when prompted to activate multipath.

This re-scans the devices and resolves the multiple paths so that each device is listed only once in the list of hard disks.

9.4 Creating a Software RAID Device for the Root (/) Partition

- 1 Proceed with the YaST install of SUSE Linux Enterprise 11 until you reach the Installation Settings page.
- 2 Click *Partitioning* to open the Preparing Hard Disk page, click *Custom Partitioning (for experts)*, then click *Next*.
- 3 On the Expert Partitioner page, expand *Hard Disks* in the *System View* panel to view the default proposal, select the proposed partitions, then click *Delete*.
- 4 Create a swap partition.

- 4a** On the Expert Partitioner page under *Hard Disks*, select the device you want to use for the swap partition, then click *Add* on the *Hard Disk Partitions* tab.
 - 4b** Under *New Partition Type*, select *Primary Partition*, then click *Next*.
 - 4c** Under *New Partition Size*, specify the size to use, then click *Next*.
 - 4d** Under *Format Options*, select *Format partition*, then select *Swap* from the drop-down list.
 - 4e** Under *Mount Options*, select *Mount partition*, then select *swap* from the drop-down list.
 - 4f** Click *Finish*.
- 5** Set up the *0xFD Linux RAID* format for each of the devices you want to use for the software RAID.
 - 5a** On the Expert Partitioner page under *Hard Disks*, select the device you want to use in the RAID, then click *Add* on the *Hard Disk Partitions* tab.
 - 5b** Under *New Partition Type*, select *Primary Partition*, then click *Next*.
 - 5c** Under *New Partition Size*, specify to use the maximum size, then click *Next*.
 - 5d** Under *Format Options*, select *Do not format partition*, then select *0xFD Linux RAID* from the drop-down list.
 - 5e** Under *Mount Options*, select *Do not mount partition*.
 - 5f** Click *Finish*.
 - 5g** Repeat Step 5a (page 156) to Step 5f (page 156) for each device that you plan to use in the software RAID
- 6** Create the RAID device.
 - 6a** In the *System View* panel, select *RAID*, then click *Add RAID* on the RAID page.

The devices that you prepared in Step 5 (page 156) are listed in *Available Devices*.

Add RAID /dev/md0

RAID Type

☒ RAID 0 (Striping)
☐ RAID 1 (Mirroring)
☐ RAID 5 (Redundant Striping)

Available Devices:

Device	Size
/dev/sda1	11.99 GB
/dev/sdb1	11.99 GB

Selected Devices:

Device	Size
--------	------

Total size: 23.98 GB

Resulting size: 0.00 B

Buttons: Add ->, Add All ->, <- Remove, <- Remove All, Help, Abort, Back, Next


- 6b** Under *RAID Type*, select *RAID 0 (Striping)*, *RAID 1 (Mirroring)*, or *RAID 5 (Redundant Striping)*.

For example, select RAID 1 (Mirroring).

- 6c** In the *Available Devices* panel, select the devices you want to use for the RAID, then click *Add* to move the devices to the *Selected Devices* panel.

Specify two or more devices for a RAID 1, two devices for a RAID 0, or at least three devices for a RAID 5.

To continue the example, two devices are selected for RAID 1.

 Add RAID /dev/md0

RAID Type

☐ RAID 0 (Striping)
☒ RAID 1 (Mirroring)
☐ RAID 5 (Redundant Striping)

Available Devices:

Device	Size

Selected Devices:

Device	Size
/dev/sda1	11.99 GB
/dev/sdb1	11.99 GB

Add →
 Add All →
 ← Remove
 ← Remove All

Total size: 0.00 B Resulting size: 11.99 GB

Help Abort Back **Next**

6d Click *Next*.

6e Under *RAID Options*, select the chunk size from the drop-down list.

The default chunk size for a RAID 1 (Mirroring) is 4 KB.

The default chunk size for a RAID 0 (Striping) is 32 KB.

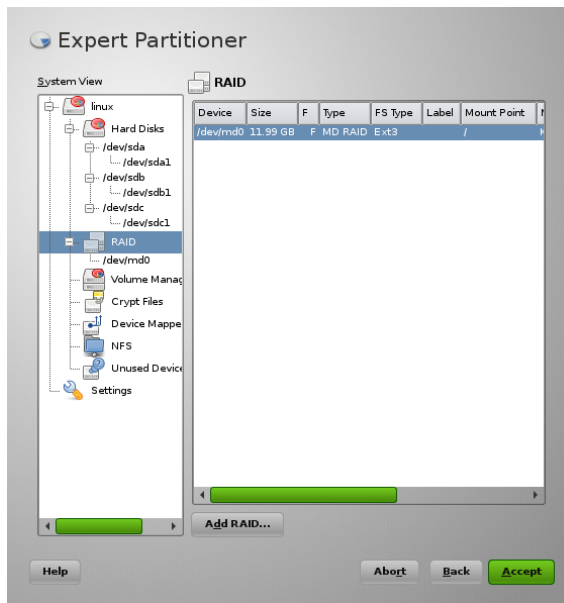
Available chunk sizes are 4 KB, 8 KB, 16 KB, 32 KB, 64 KB, 128 KB, 256 KB, 512 KB, 1 MB, 2 MB, or 4 MB.

6f Under *Formatting Options*, select *Format partition*, then select the file system type (such as Ext3) from the *File system* drop-down list.

6g Under *Mounting Options*, select *Mount partition*, then select */* from the *Mount Point* drop-down list.

6h Click *Finish*.

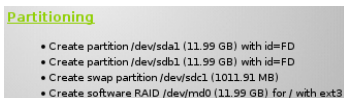
The software RAID device is managed by Device Mapper, and creates a device under the `/dev/md0` path.



- 7 On the Expert Partitioner page, click *Accept*.

The new proposal appears under *Partitioning* on the Installation Settings page.

For example, the setup for the



- 8 Continue with the install.

Whenever you reboot your server, Device Mapper is started at boot time so that the software RAID is automatically recognized, and the operating system on the root (`/`) partition can be started.

Managing Software RAID 6 and 10 with mdadm

10

This section describes how to create software RAID 6 and 10 devices, using the Multiple Devices Administration (`mdadm(8)`) tool. You can also use `mdadm` to create RAID 0, 1, 4, and 5. The `mdadm` tool provides the functionality of legacy programs `mdtools` and `raidtools`.

- Section 10.1, “Creating a RAID 6” (page 161)
- Section 10.2, “Creating Nested RAID 10 Devices with `mdadm`” (page 163)
- Section 10.3, “Creating a Complex RAID 10 with `mdadm`” (page 168)
- Section 10.4, “Creating a Degraded RAID Array” (page 173)

10.1 Creating a RAID 6

- Section 10.1.1, “Understanding RAID 6” (page 161)
- Section 10.1.2, “Creating a RAID 6” (page 162)

10.1.1 Understanding RAID 6

RAID 6 is essentially an extension of RAID 5 that allows for additional fault tolerance by using a second independent distributed parity scheme (dual parity). Even if two of

the hard disk drives fail during the data recovery process, the system continues to be operational, with no data loss.

RAID 6 provides for extremely high data fault tolerance by sustaining multiple simultaneous drive failures. It handles the loss of any two devices without data loss. Accordingly, it requires $N+2$ drives to store N drives worth of data. It requires a minimum of 4 devices.

The performance for RAID 6 is slightly lower but comparable to RAID 5 in normal mode and single disk failure mode. It is very slow in dual disk failure mode.

Table 10.1 *Comparison of RAID 5 and RAID 6*

Feature	RAID 5	RAID 6
Number of devices	$N+1$, minimum of 3	$N+2$, minimum of 4
Parity	Distributed, single	Distributed, dual
Performance	Medium impact on write and rebuild	More impact on sequential write than RAID 5
Fault-tolerance	Failure of one component device	Failure of two component devices

10.1.2 Creating a RAID 6

The procedure in this section creates a RAID 6 device `/dev/md0` with four devices: `/dev/sda1`, `/dev/sdb1`, `/dev/sdc1`, and `/dev/sdd1`. Ensure that you modify the procedure to use your actual device nodes.

- 1 Open a terminal console, then log in as the `root` user or equivalent.
- 2 Create a RAID 6 device. At the command prompt, enter

```
mdadm --create /dev/md0 --run --level=raid6 --chunk=128 --raid-devices=4
/dev/sdb1 /dev/sdc1 /dev/sdc1 /dev/sdd1
```

The default chunk size is 64 KB.

- 3 Create a file system on the RAID 6 device `/dev/md0`, such as a Reiser file system (reiserfs). For example, at the command prompt, enter

```
mkfs.reiserfs /dev/md0
```

Modify the command if you want to use a different file system.

- 4 Edit the `/etc/mdadm.conf` file to add entries for the component devices and the RAID device `/dev/md0`.
- 5 Edit the `/etc/fstab` file to add an entry for the RAID 6 device `/dev/md0`.
- 6 Reboot the server.

The RAID 6 device is mounted to `/local`.

- 7 (Optional) Add a hot spare to service the RAID array. For example, at the command prompt enter:

```
mdadm /dev/md0 -a /dev/sde1
```

10.2 Creating Nested RAID 10 Devices with mdadm

- Section 10.2.1, “Understanding Nested RAID Devices” (page 163)
- Section 10.2.2, “Creating Nested RAID 10 (1+0) with mdadm” (page 165)
- Section 10.2.3, “Creating Nested RAID 10 (0+1) with mdadm” (page 167)

10.2.1 Understanding Nested RAID Devices

A nested RAID device consists of a RAID array that uses another RAID array as its basic element, instead of using physical disks. The goal of this configuration is to improve the performance and fault tolerance of the RAID.

Linux supports nesting of RAID 1 (mirroring) and RAID 0 (striping) arrays. Generally, this combination is referred to as RAID 10. To distinguish the order of the nesting, this document uses the following terminology:

- **RAID 1+0:** RAID 1 (mirror) arrays are built first, then combined to form a RAID 0 (stripe) array.
- **RAID 0+1:** RAID 0 (stripe) arrays are built first, then combined to form a RAID 1 (mirror) array.

The following table describes the advantages and disadvantages of RAID 10 nesting as 1+0 versus 0+1. It assumes that the storage objects you use reside on different disks, each with a dedicated I/O capability.

Table 10.2 *Nested RAID Levels*

RAID Level	Description	Performance and Fault Tolerance
10 (1+0)	RAID 0 (stripe) built with RAID 1 (mirror) arrays	<p>RAID 1+0 provides high levels of I/O performance, data redundancy, and disk fault tolerance. Because each member device in the RAID 0 is mirrored individually, multiple disk failures can be tolerated and data remains available as long as the disks that fail are in different mirrors.</p> <p>You can optionally configure a spare for each underlying mirrored array, or configure a spare to serve a spare group that serves all mirrors.</p>
10 (0+1)	RAID 1 (mirror) built with RAID 0 (stripe) arrays	<p>RAID 0+1 provides high levels of I/O performance and data redundancy, but slightly less fault tolerance than a 1+0. If multiple disks fail on one side of the mirror, then the other mirror is available. However, if disks are lost concurrently on both sides of the mirror, all data is lost.</p> <p>This solution offers less disk fault tolerance than a 1+0 solution, but if you need to perform maintenance or maintain the mirror on a different site, you can take an entire side of the mirror offline and still have a fully functional storage device. Also, if you lose the connection</p>

RAID Level	Description	Performance and Fault Tolerance
		between the two sites, either site operates independently of the other. That is not true if you stripe the mirrored segments, because the mirrors are managed at a lower level.
		If a device fails, the mirror on that side fails because RAID 1 is not fault-tolerant. Create a new RAID 0 to replace the failed side, then resynchronize the mirrors.

10.2.2 Creating Nested RAID 10 (1+0) with mdadm

A nested RAID 1+0 is built by creating two or more RAID 1 (mirror) devices, then using them as component devices in a RAID 0.

IMPORTANT

If you need to manage multiple connections to the devices, you must configure multipath I/O before configuring the RAID devices. For information, see Chapter 7, *Managing Multipath I/O for Devices* (page 77).

The procedure in this section uses the device names shown in the following table. Ensure that you modify the device names with the names of your own devices.

Table 10.3 Scenario for Creating a RAID 10 (1+0) by Nesting

Raw Devices	RAID 1 (mirror)	RAID 1+0 (striped mirrors)
/dev/sdb1 /dev/sdc1	/dev/md0	/dev/md2
/dev/sdd1	/dev/md1	

Raw Devices	RAID 1 (mirror)	RAID 1+0 (striped mirrors)
<hr/>		
<code>/dev/sde1</code>		
<hr/>		

- 1 Open a terminal console, then log in as the `root` user or equivalent.
- 2 Create 2 software RAID 1 devices, using two different devices for each RAID 1 device. At the command prompt, enter these two commands:

```
mdadm --create /dev/md0 --run --level=1 --raid-devices=2 /dev/sdb1 /dev/sdc1
```

```
mdadm --create /dev/md1 --run --level=1 --raid-devices=2 /dev/sdd1 /dev/sde1
```

- 3 Create the nested RAID 1+0 device. At the command prompt, enter the following command using the software RAID 1 devices you created in Step 2 (page 166):

```
mdadm --create /dev/md2 --run --level=0 --chunk=64 --raid-devices=2 /dev/md0 /dev/md1
```

The default chunk size is 64 KB.

- 4 Create a file system on the RAID 1+0 device `/dev/md2`, such as a Reiser file system (reiserfs). For example, at the command prompt, enter

```
mkfs.reiserfs /dev/md2
```

Modify the command if you want to use a different file system.

- 5 Edit the `/etc/mdadm.conf` file to add entries for the component devices and the RAID device `/dev/md2`.
- 6 Edit the `/etc/fstab` file to add an entry for the RAID 1+0 device `/dev/md2`.
- 7 Reboot the server.

The RAID 1+0 device is mounted to `/local`.

10.2.3 Creating Nested RAID 10 (0+1) with mdadm

A nested RAID 0+1 is built by creating two to four RAID 0 (striping) devices, then mirroring them as component devices in a RAID 1.

IMPORTANT

If you need to manage multiple connections to the devices, you must configure multipath I/O before configuring the RAID devices. For information, see Chapter 7, *Managing Multipath I/O for Devices* (page 77).

In this configuration, spare devices cannot be specified for the underlying RAID 0 devices because RAID 0 cannot tolerate a device loss. If a device fails on one side of the mirror, you must create a replacement RAID 0 device, then add it into the mirror.

The procedure in this section uses the device names shown in the following table. Ensure that you modify the device names with the names of your own devices.

Table 10.4 Scenario for Creating a RAID 10 (0+1) by Nesting

Raw Devices	RAID 0 (stripe)	RAID 0+1 (mirrored stripes)
/dev/sdb1 /dev/sdc1	/dev/md0	/dev/md2
/dev/sdd1 /dev/sde1	/dev/md1	

- 1 Open a terminal console, then log in as the root user or equivalent.
- 2 Create two software RAID 0 devices, using two different devices for each RAID 0 device. At the command prompt, enter these two commands:

```
mdadm --create /dev/md0 --run --level=0 --chunk=64 --raid-devices=2 /dev/sdb1 /dev/sdc1
```

```
mdadm --create /dev/md1 --run --level=0 --chunk=64 --raid-devices=2 /dev/sdd1  
/dev/sde1
```

The default chunk size is 64 KB.

- 3 Create the nested RAID 0+1 device. At the command prompt, enter the following command using the software RAID 0 devices you created in Step 2 (page 167):

```
mdadm --create /dev/md2 --run --level=1 --raid-devices=2 /dev/md0 /dev/md1
```

- 4 Create a file system on the RAID 0+1 device `/dev/md2`, such as a Reiser file system (reiserfs). For example, at the command prompt, enter

```
mkfs.reiserfs /dev/md2
```

Modify the command if you want to use a different file system.

- 5 Edit the `/etc/mdadm.conf` file to add entries for the component devices and the RAID device `/dev/md2`.
- 6 Edit the `/etc/fstab` file to add an entry for the RAID 0+1 device `/dev/md2`.
- 7 Reboot the server.

The RAID 0+1 device is mounted to `/local`.

10.3 Creating a Complex RAID 10 with mdadm

- Section 10.3.1, “Understanding the mdadm RAID10” (page 169)
- Section 10.3.2, “Creating a RAID 10 with mdadm” (page 172)

10.3.1 Understanding the mdadm RAID10

In mdadm, the RAID10 level creates a single complex software RAID that combines features of both RAID 0 (striping) and RAID 1 (mirroring). Multiple copies of all data blocks are arranged on multiple drives following a striping discipline. Component devices should be the same size.

- Section “Comparing the Complex RAID10 and Nested RAID 10 (1+0)” (page 169)
- Section “Number of Replicas in the mdadm RAID10” (page 170)
- Section “Number of Devices in the mdadm RAID10” (page 170)
- Section “Near Layout” (page 171)
- Section “Far Layout” (page 171)

Comparing the Complex RAID10 and Nested RAID 10 (1+0)

The complex RAID 10 is similar in purpose to a nested RAID 10 (1+0), but differs in the following ways:

Table 10.5 *Complex vs. Nested RAID 10*

Feature	mdadm RAID10 Option	Nested RAID 10 (1+0)
Number of devices	Allows an even or odd number of component devices	Requires an even number of component devices
Component devices	Managed as a single RAID device	Manage as a nested RAID device
Striping	Striping occurs in the near or far layout on component devices.	Striping occurs consecutively across component devices

Feature	mdadm RAID10 Option	Nested RAID 10 (1+0)
	The far layout provides sequential read throughput that scales by number of drives, rather than number of RAID 1 pairs.	
Multiple copies of data	Two or more copies, up to the number of devices in the array	Copies on each mirrored segment
Hot spare devices	A single spare can service all component devices	Configure a spare for each underlying mirrored array, or configure a spare to serve a spare group that serves all mirrors.

Number of Replicas in the mdadm RAID10

When configuring an mdadm RAID10 array, you must specify the number of replicas of each data block that are required. The default number of replicas is 2, but the value can be 2 to the number of devices in the array.

Number of Devices in the mdadm RAID10

You must use at least as many component devices as the number of replicas you specify. However, number of component devices in a RAID10 array does not need to be a multiple of the number of replicas of each data block. The effective storage size is the number of devices divided by the number of replicas.

For example, if you specify 2 replicas for an array created with 5 component devices, a copy of each block is stored on two different devices. The effective storage size for one copy of all data is $5/2$ or 2.5 times the size of a component device.

Near Layout

With the near layout, copies of a block of data are striped near each other on different component devices. That is, multiple copies of one data block are at similar offsets in different devices. Near is the default layout for RAID10. For example, if you use an odd number of component devices and two copies of data, some copies are perhaps one chunk further into the device.

The near layout for the `mdadm` RAID10 yields read and write performance similar to RAID 0 over half the number of drives.

Near layout with an even number of disks and two replicas:

sda1	sdb1	sdcl	sde1
0	0	1	1
2	2	3	3
4	4	5	5
6	6	7	7
8	8	9	9

Near layout with an odd number of disks and two replicas:

sda1	sdb1	sdcl	sde1	sdf1
0	0	1	1	2
2	3	3	4	4
5	5	6	6	7
7	8	8	9	9
10	10	11	11	12

Far Layout

The far layout stripes data over the early part of all drives, then stripes a second copy of the data over the later part of all drives, making sure that all copies of a block are on different drives. The second set of values starts halfway through the component drives.

With a far layout, the read performance of the `mdadm` RAID10 is similar to a RAID 0 over the full number of drives, but write performance is substantially slower than a RAID 0 because there is more seeking of the drive heads. It is best used for read-intensive operations such as for read-only file servers.

The speed of the `raid10` for writing is similar to other mirrored RAID types, like `raid1` and `raid10` using near layout, as the elevator of the file system schedules the writes in

a more optimal way than raw writing. Using raid10 in the far layout well-suited for mirrored writing applications.

Far layout with an even number of disks and two replicas:

```
sda1 sdb1 sdc1 sde1
0    1    2    3
4    5    6    7
.    .    .
3    0    1    2
7    4    5    6
```

Far layout with an odd number of disks and two replicas:

```
sda1 sdb1 sdc1 sde1 sdf1
0    1    2    3    4
5    6    7    8    9
.    .    .
4    0    1    2    3
9    5    6    7    8
```

10.3.2 Creating a RAID 10 with mdadm

The RAID10 option for mdadm creates a RAID 10 device without nesting. For information about RAID10-, see Section 10.3, “Creating a Complex RAID 10 with mdadm” (page 168).

The procedure in this section uses the device names shown in the following table. Ensure that you modify the device names with the names of your own devices.

Table 10.6 Scenario for Creating a RAID 10 Using the mdadm RAID10 Option

Raw Devices	RAID10 (near or far striping scheme)
/dev/sdf1	/dev/md3
/dev/sdg1	
/dev/sdh1	
/dev/sdi1	

- 1 In YaST, create a 0xFD Linux RAID partition on the devices you want to use in the RAID, such as `/dev/sdf1`, `/dev/sdg1`, `/dev/sdh1`, and `/dev/sdi1`.
- 2 Open a terminal console, then log in as the root user or equivalent.
- 3 Create a RAID 10 command. At the command prompt, enter (all on the same line):

```
mdadm --create /dev/md3 --run --level=10 --chunk=4 --raid-devices=4 /dev/sdf1  
/dev/sdg1 /dev/sdh1 /dev/sdi1
```

- 4 Create a Reiser file system on the RAID 10 device `/dev/md3`. At the command prompt, enter

```
mkfs.reiserfs /dev/md3
```

- 5 Edit the `/etc/mdadm.conf` file to add entries for the component devices and the RAID device `/dev/md3`. For example:

```
DEVICE /dev/md3
```

- 6 Edit the `/etc/fstab` file to add an entry for the RAID 10 device `/dev/md3`.
- 7 Reboot the server.

The RAID10 device is mounted to `/raid10`.

10.4 Creating a Degraded RAID Array

A degraded array is one in which some devices are missing. Degraded arrays are supported only for RAID 1, RAID 4, RAID 5, and RAID 6. These RAID types are designed to withstand some missing devices as part of their fault-tolerance features. Typically, degraded arrays occur when a device fails. It is possible to create a degraded array on purpose.

RAID Type	Allowable Number of Slots Missing
RAID 1	All but one device
RAID 4	One slot
RAID 5	One slot
RAID 6	One or two slots

To create a degraded array in which some devices are missing, simply give the word `missing` in place of a device name. This causes `mdadm` to leave the corresponding slot in the array empty.

When creating a RAID 5 array, `mdadm` automatically creates a degraded array with an extra spare drive. This is because building the spare into a degraded array is generally faster than resynchronizing the parity on a non-degraded, but not clean, array. You can override this feature with the `--force` option.

Creating a degraded array might be useful if you want create a RAID, but one of the devices you want to use already has data on it. In that case, you create a degraded array with other devices, copy data from the in-use device to the RAID that is running in degraded mode, add the device into the RAID, then wait while the RAID is rebuilt so that the data is now across all devices. An example of this process is given in the following procedure:

- 1 Create a degraded RAID 1 device `/dev/md0`, using one single drive `/dev/sd1`, enter the following at the command prompt:

```
mdadm --create /dev/md0 -l 1 -n 2 /dev/sd1 missing
```

The device should be the same size or larger than the device you plan to add to it.

- 2 If the device you want to add to the mirror contains data that you want to move to the RAID array, copy it now to the RAID array while it is running in degraded mode.
- 3 Add a device to the mirror. For example, to add `/dev/sdb1` to the RAID, enter the following at the command prompt:


```
mdadm /dev/md0 -a /dev/sdb1
```

You can add only one device at a time. You must wait for the kernel to build the mirror and bring it fully online before you add another mirror.

- 4** Monitor the build progress by entering the following at the command prompt:

```
cat /proc/mdstat
```

To see the rebuild progress while being refreshed every second, enter

```
watch -n 1 cat /proc/mdstat
```


Resizing Software RAID Arrays with mdadm

11

This section describes how to increase or reduce the size of a software RAID 1, 4, 5, or 6 device with the Multiple Device Administration (`mdadm(8)`) tool.

WARNING

Before starting any of the tasks described in this section, ensure that you have a valid backup of all of the data.

- Section 11.1, “Understanding the Resizing Process” (page 177)
- Section 11.2, “Increasing the Size of a Software RAID” (page 179)
- Section 11.3, “Decreasing the Size of a Software RAID” (page 186)

11.1 Understanding the Resizing Process

Resizing an existing software RAID device involves increasing or decreasing the space contributed by each component partition.

- Section 11.1.1, “Guidelines for Resizing a Software RAID ” (page 178)
- Section 11.1.2, “Overview of Tasks” (page 178)

11.1.1 Guidelines for Resizing a Software RAID

The `mdadm` (8) tool supports resizing only for software RAID levels 1, 4, 5, and 6. These RAID levels provide disk fault tolerance so that one component partition can be removed at a time for resizing. In principle, it is possible to perform a hot resize for RAID partitions, but you must take extra care for your data when doing so.

The file system that resides on the RAID must also be able to be resized in order to take advantage of the changes in available space on the device. In SUSE Linux Enterprise Server 11, file system resizing utilities are available for file systems Ext2, Ext3, and ReiserFS. The utilities support increasing and decreasing the size as follows:

Table 11.1 *File System Support for Resizing*

File System	Utility	Increase Size	Decrease Size
Ext2 or Ext3	resize2fs	Yes, offline only	Yes, offline only
ReiserFS	resize_reiserfs	Yes, online or offline	Yes, offline only

Resizing any partition or file system involves some risks that can potentially result in losing data.

WARNING

To avoid data loss, ensure that you back up your data before you begin any resizing task.

11.1.2 Overview of Tasks

Resizing the RAID involves the following tasks. The order in which these tasks is performed depends on whether you are increasing or decreasing its size.

Table 11.2 *Tasks Involved in Resizing a RAID*

Tasks	Description	Order If In- creas- ing Size	Order If De- creas- ing Size
Resize each of the component partitions.	Increase or decrease the active size of each component partition. You remove only one component partition at a time, modify its size, then return it to the RAID.	1	2
Resize the software RAID itself.	The RAID does not automatically know about the increases or decreases you make to the underlying component partitions. You must inform it about the new size.	2	3
Resize the file system.	You must resize the file system that resides on the RAID. This is possible only for file systems that provide tools for resizing, such as Ext2, Ext3, and ReiserFS.	3	1

11.2 Increasing the Size of a Software RAID

Before you begin, review the guidelines in Section 11.1, “Understanding the Resizing Process” (page 177).

- Section 11.2.1, “Increasing the Size of Component Partitions” (page 180)
- Section 11.2.2, “Increasing the Size of the RAID Array” (page 182)
- Section 11.2.3, “Increasing the Size of the File System” (page 183)

11.2.1 Increasing the Size of Component Partitions

Apply the procedure in this section to increase the size of a RAID 1, 4, 5, or 6. For each component partition in the RAID, remove the partition from the RAID, modify its size, return it to the RAID, then wait until the RAID stabilizes to continue. While a partition is removed, the RAID operates in degraded mode and has no or reduced disk fault tolerance. Even for RAIDs that can tolerate multiple concurrent disk failures, do not remove more than one component partition at a time.

WARNING

If a RAID does not have disk fault tolerance, or it is simply not consistent, data loss results if you remove any of its partitions. Be very careful when removing partitions, and ensure that you have a backup of your data available.

The procedure in this section uses the device names shown in the following table. Ensure that you modify the names to use the names of your own devices.

Table 11.3 *Scenario for Increasing the Size of Component Partitions*

RAID Device	Component Partitions
/dev/md0	/dev/sda1
	/dev/sdb1
	/dev/sdc1

To increase the size of the component partitions for the RAID:

- 1 Open a terminal console, then log in as the `root` user or equivalent.
- 2 Ensure that the RAID array is consistent and synchronized by entering

```
cat /proc/mdstat
```

If your RAID array is still synchronizing according to the output of this command, you must wait until synchronization is complete before continuing.

- 3** Remove one of the component partitions from the RAID array. For example, to remove `/dev/sda1`, enter

```
mdadm /dev/md0 --fail /dev/sda1 --remove /dev/sda1
```

In order to succeed, both the fail and remove actions must be done.

- 4** Increase the size of the partition that you removed in Step 3 (page 181) by doing one of the following:

- Increase the size of the partition, using a disk partitioner such as `fdisk(8)`, `cfdisk(8)`, or `parted(8)`. This option is the usual choice.
- Replace the disk on which the partition resides with a higher-capacity device.

This option is possible only if no other file systems on the original disk are accessed by the system. When the replacement device is added back into the RAID, it takes much longer to synchronize the data because all of the data that was on the original device must be rebuilt.

- 5** Re-add the partition to the RAID array. For example, to add `/dev/sda1`, enter

```
mdadm -a /dev/md0 /dev/sda1
```

Wait until the RAID is synchronized and consistent before continuing with the next partition.

- 6** Repeat Step 2 (page 180) through Step 5 (page 181) for each of the remaining component devices in the array. Ensure that you modify the commands for the correct component partition.
- 7** If you get a message that tells you that the kernel could not re-read the partition table for the RAID, you must reboot the computer after all partitions have been resized to force an update of the partition table.
- 8** Continue with Section 11.2.2, “Increasing the Size of the RAID Array” (page 182).

11.2.2 Increasing the Size of the RAID Array

After you have resized each of the component partitions in the RAID (see Section 11.2.1, “Increasing the Size of Component Partitions” (page 180)), the RAID array configuration continues to use the original array size until you force it to be aware of the newly available space. You can specify a size for the RAID or use the maximum available space.

The procedure in this section uses the device name `/dev/md0` for the RAID device. Ensure that you modify the name to use the name of your own device.

- 1 Open a terminal console, then log in as the `root` user or equivalent.
- 2 Check the size of the array and the device size known to the array by entering

```
mdadm -D /dev/md0 | grep -e "Array Size" -e "Device Size"
```

- 3 Do one of the following:

- Increase the size of the array to the maximum available size by entering

```
mdadm --grow /dev/md0 -z max
```

- Increase the size of the array to the maximum available size by entering

```
mdadm --grow /dev/md0 -z max --assume-clean
```

The array makes use of any space that has been added to the devices, but this space will not be synchronized. This is recommended for RAID1 because the sync is not needed. It can be useful for other RAID levels if the space that was added to the member devices was pre-zeroed.

- Increase the size of the array to a specified value by entering

```
mdadm --grow /dev/md0 -z size
```

Replace *size* with an integer value in kilobytes (a kilobyte is 1024 bytes) for the desired size.

4 Recheck the size of your array and the device size known to the array by entering

```
mdadm -D /dev/md0 | grep -e "Array Size" -e "Device Size"
```

5 Do one of the following:

- If your array was successfully resized, continue with Section 11.2.3, “Increasing the Size of the File System” (page 183).
- If your array was not resized as you expected, you must reboot, then try this procedure again.

11.2.3 Increasing the Size of the File System

After you increase the size of the array (see Section 11.2.2, “Increasing the Size of the RAID Array” (page 182)), you are ready to resize the file system.

You can increase the size of the file system to the maximum space available or specify an exact size. When specifying an exact size for the file system, ensure that the new size satisfies the following conditions:

- The new size must be greater than the size of the existing data; otherwise, data loss occurs.
- The new size must be equal to or less than the current RAID size because the file system size cannot extend beyond the space available.

Ext2 or Ext3

Ext2 and Ext3 file systems can be resized when mounted or unmounted with the `resize2fs` command.

- 1** Open a terminal console, then log in as the `root` user or equivalent.
- 2** Increase the size of the file system using one of the following methods:
 - To extend the file system size to the maximum available size of the software RAID device called `/dev/md0`, enter

```
resize2fs /dev/md0
```

If a size parameter is not specified, the size defaults to the size of the partition.

- To extend the file system to a specific size, enter

```
resize2fs /dev/md0 size
```

The *size* parameter specifies the requested new size of the file system. If no units are specified, the unit of the size parameter is the block size of the file system. Optionally, the size parameter can be suffixed by one of the following the unit designators: s for 512 byte sectors; K for kilobytes (1 kilobyte is 1024 bytes); M for megabytes; or G for gigabytes.

Wait until the resizing is completed before continuing.

- 3** If the file system is not mounted, mount it now.

For example, to mount an Ext2 file system for a RAID named `/dev/md0` at mount point `/raid`, enter

```
mount -t ext2 /dev/md0 /raid
```

- 4** Check the effect of the resize on the mounted file system by entering

```
df -h
```

The Disk Free (`df`) command shows the total size of the disk, the number of blocks used, and the number of blocks available on the file system. The `-h` option print sizes in human-readable format, such as 1K, 234M, or 2G.

ReiserFS

As with Ext2 and Ext3, a ReiserFS file system can be increased in size while mounted or unmounted. The resize is done on the block device of your RAID array.

- 1** Open a terminal console, then log in as the `root` user or equivalent.

- 2 Increase the size of the file system on the software RAID device called `/dev/md0`, using one of the following methods:

- To extend the file system size to the maximum available size of the device, enter

```
resize_reiserfs /dev/md0
```

When no size is specified, this increases the volume to the full size of the partition.

- To extend the file system to a specific size, enter

```
resize_reiserfs -s size /dev/md0
```

Replace *size* with the desired size in bytes. You can also specify units on the value, such as 50000K (kilobytes), 250M (megabytes), or 2G (gigabytes). Alternatively, you can specify an increase to the current size by prefixing the value with a plus (+) sign. For example, the following command increases the size of the file system on `/dev/md0` by 500 MB:

```
resize_reiserfs -s +500M /dev/md0
```

Wait until the resizing is completed before continuing.

- 3 If the file system is not mounted, mount it now.

For example, to mount an ReiserFS file system for a RAID named `/dev/md0` at mount point `/raid`, enter

```
mount -t reiserfs /dev/md0 /raid
```

- 4 Check the effect of the resize on the mounted file system by entering

```
df -h
```

The Disk Free (`df`) command shows the total size of the disk, the number of blocks used, and the number of blocks available on the file system. The `-h` option print sizes in human-readable format, such as 1K, 234M, or 2G.

11.3 Decreasing the Size of a Software RAID

Before you begin, review the guidelines in Section 11.1, “Understanding the Resizing Process” (page 177).

- Section 11.3.1, “Decreasing the Size of the File System” (page 186)
- Section 11.3.2, “Decreasing the Size of Component Partitions” (page 188)
- Section 11.3.3, “Decreasing the Size of the RAID Array” (page 190)

11.3.1 Decreasing the Size of the File System

When decreasing the size of the file system on a RAID device, ensure that the new size satisfies the following conditions:

- The new size must be greater than the size of the existing data; otherwise, data loss occurs.
- The new size must be equal to or less than the current RAID size because the file system size cannot extend beyond the space available.

In SUSE Linux Enterprise Server, Ext2, Ext3, and ReiserFS provide utilities for decreasing the size of the file system. Use the appropriate procedure below for decreasing the size of your file system.

The procedures in this section use the device name `/dev/md0` for the RAID device. Ensure that you modify commands to use the name of your own device.

Ext2 or Ext3

The Ext2 and Ext3 file systems can be resized when mounted or unmounted.

- 1 Open a terminal console, then log in as the `root` user or equivalent.
- 2 Decrease the size of the file system on the RAID by entering

```
resize2fs /dev/md0 <size>
```

Replace *size* with an integer value in kilobytes for the desired size. (A kilobyte is 1024 bytes.)

Wait until the resizing is completed before continuing.

- 3 If the file system is not mounted, mount it now. For example, to mount an Ext2 file system for a RAID named `/dev/md0` at mount point `/raid`, enter

```
mount -t ext2 /dev/md0 /raid
```

- 4 Check the effect of the resize on the mounted file system by entering

```
df -h
```

The Disk Free (`df`) command shows the total size of the disk, the number of blocks used, and the number of blocks available on the file system. The `-h` option print sizes in human-readable format, such as 1K, 234M, or 2G.

ReiserFS

ReiserFS file systems can be decreased in size only if the volume is unmounted.

- 1 Open a terminal console, then log in as the `root` user or equivalent.
- 2 Unmount the device by entering

```
umount /mnt/point
```

If the partition you are attempting to decrease in size contains system files (such as the root (`/`) volume), unmounting is possible only when booting from a bootable CD or floppy.

- 3 Decrease the size of the file system on the software RAID device called `/dev/md0` by entering

```
resize_reiserfs -s size /dev/md0
```

Replace *size* with the desired size in bytes. You can also specify units on the value, such as 50000K (kilobytes), 250M (megabytes), or 2G (gigabytes). Alternatively, you can specify a decrease to the current size by prefixing the value with a minus (-) sign. For example, the following command reduces the size of the file system on /dev/md0 by 500 MB:

```
resize_reiserfs -s -500M /dev/md0
```

Wait until the resizing is completed before continuing.

4 Mount the file system by entering

```
mount -t reiserfs /dev/md0 /mnt/point
```

5 Check the effect of the resize on the mounted file system by entering

```
df -h
```

The Disk Free (df) command shows the total size of the disk, the number of blocks used, and the number of blocks available on the file system. The -h option prints sizes in human-readable format, such as 1K, 234M, or 2G.

11.3.2 Decreasing the Size of Component Partitions

Resize the RAID's component partitions one at a time. For each component partition, you remove it from the RAID, modify its partition size, return the partition to the RAID, then wait until the RAID stabilizes. While a partition is removed, the RAID operates in degraded mode and has no or reduced disk fault tolerance. Even for RAID configurations that can tolerate multiple concurrent disk failures, you should never remove more than one component partition at a time.

WARNING

If a RAID does not have disk fault tolerance, or it is simply not consistent, data loss results if you remove any of its partitions. Be very careful when removing partitions, and ensure that you have a backup of your data available.

The procedure in this section uses the device names shown in the following table. Ensure that you modify the commands to use the names of your own devices.

Table 11.4 *Scenario for Increasing the Size of Component Partitions*

RAID Device	Component Partitions
/dev/md0	/dev/sda1
	/dev/sdb1
	/dev/sdc1

To resize the component partitions for the RAID:

- 1 Open a terminal console, then log in as the `root` user or equivalent.
- 2 Ensure that the RAID array is consistent and synchronized by entering

```
cat /proc/mdstat
```

If your RAID array is still synchronizing according to the output of this command, you must wait until synchronization is complete before continuing.

- 3 Remove one of the component partitions from the RAID array. For example, to remove `/dev/sda1`, enter

```
mdadm /dev/md0 --fail /dev/sda1 --remove /dev/sda1
```

In order to succeed, both the fail and remove actions must be done.

- 4 Increase the size of the partition that you removed in Step 3 (page 181) by doing one of the following:
 - Use a disk partitioner such as `fdisk`, `cfdisk`, or `parted` to increase the size of the partition.
 - Replace the disk on which the partition resides with a different device.

This option is possible only if no other file systems on the original disk are accessed by the system. When the replacement device is added back into the RAID, it takes much longer to synchronize the data.

- 5 Re-add the partition to the RAID array. For example, to add `/dev/sda1`, enter

```
mdadm -a /dev/md0 /dev/sda1
```

Wait until the RAID is synchronized and consistent before continuing with the next partition.

- 6 Repeat Step 2 (page 180) through Step 5 (page 181) for each of the remaining component devices in the array. Ensure that you modify the commands for the correct component partition.
- 7 If you get a message that tells you that the kernel could not re-read the partition table for the RAID, you must reboot the computer after resizing all of its component partitions.
- 8 Continue with Section 11.3.3, “Decreasing the Size of the RAID Array” (page 190).

11.3.3 Decreasing the Size of the RAID Array

After you have resized each of the component partitions in the RAID, the RAID array configuration continues to use the original array size until you force it to be aware of the newly available space. Use the `--grow` option to force it to read the change in available disk size. You can specify a size for the RAID or use the maximum available space.

The procedure in this section uses the device name `/dev/md0` for the RAID device. Ensure that you modify commands to use the name of your own device.

- 1 Open a terminal console, then log in as the `root` user or equivalent.
- 2 Check the size of the array and the device size known to the array by entering

```
mdadm -D /dev/md0 | grep -e "Array Size" -e "Device Size"
```


3 Do one of the following:

- Decrease the size of the array to the maximum available size by entering

```
mdadm --grow /dev/md0 -z max
```

- Decrease the size of the array to a specified value by entering

```
mdadm --grow /dev/md0 -z size
```

Replace *size* with an integer value in kilobytes for the desired size. (A kilobyte is 1024 bytes.)

4 Recheck the size of your array and the device size known to the array by entering

```
mdadm -D /dev/md0 | grep -e "Array Size" -e "Device Size"
```

5 Do one of the following:

- If your array was successfully resized, you are done.
- If your array was not resized as you expected, you must reboot, then try this procedure again.

iSNS for Linux

Storage area networks (SANs) can contain many disk drives that are dispersed across complex networks. This can make device discovery and device ownership difficult. iSCSI initiators must be able to identify storage resources in the SAN and determine whether they have access to them.

Internet Storage Name Service (iSNS) is a standards-based service that is available beginning with SUSE Linux Enterprise Server (SLES) 10 Support Pack 2. iSNS facilitates the automated discovery, management, and configuration of iSCSI devices on a TCP/IP network. iSNS provides intelligent storage discovery and management services comparable to those found in Fibre Channel networks.

IMPORTANT

iSNS should be used only in secure internal networks.

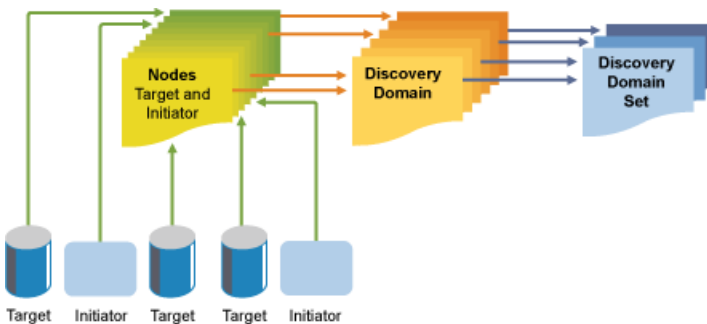
- Section 12.1, “How iSNS Works” (page 194)
- Section 12.2, “Installing iSNS Server for Linux” (page 195)
- Section 12.3, “Configuring iSNS Discovery Domains” (page 197)
- Section 12.4, “Starting iSNS” (page 204)
- Section 12.5, “Stopping iSNS” (page 204)
- Section 12.6, “For More Information” (page 205)

12.1 How iSNS Works

For an iSCSI initiator to discover iSCSI targets, it needs to identify which devices in the network are storage resources and what IP addresses it needs to access them. A query to an iSNS server returns a list of iSCSI targets and the IP addresses that the initiator has permission to access.

Using iSNS, you create iSNS discovery domains and discovery domain sets. You then group or organize iSCSI targets and initiators into discovery domains and group the discovery domains into discovery domain sets. By dividing storage nodes into domains, you can limit the discovery process of each host to the most appropriate subset of targets registered with iSNS, which allows the storage network to scale by reducing the number of unnecessary discoveries and by limiting the amount of time each host spends establishing discovery relationships. This lets you control and simplify the number of targets and initiators that must be discovered.

Figure 12.1 *iSNS Discovery Domains and Discovery Domain Sets*



Both iSCSI targets and iSCSI initiators use iSNS clients to initiate transactions with iSNS servers by using the iSNS protocol. They then register device attribute information in a common discovery domain, download information about other registered clients, and receive asynchronous notification of events that occur in their discovery domain.

iSNS servers respond to iSNS protocol queries and requests made by iSNS clients using the iSNS protocol. iSNS servers initiate iSNS protocol state change notifications and store properly authenticated information submitted by a registration request in an iSNS database.

Some of the benefits provided by iSNS for Linux include:

- Provides an information facility for registration, discovery, and management of networked storage assets.
- Integrates with the DNS infrastructure.
- Consolidates registration, discovery, and management of iSCSI storage.
- Simplifies storage management implementations.
- Improves scalability compared to other discovery methods.

An example of the benefits iSNS provides can be better understood through the following scenario:

Suppose you have a company that has 100 iSCSI initiators and 100 iSCSI targets. Depending on your configuration, all iSCSI initiators could potentially try to discover and connect to any of the 100 iSCSI targets. This could create discovery and connection difficulties. By grouping initiators and targets into discovery domains, you can prevent iSCSI initiators in one department from discovering the iSCSI targets in another department. The result is that the iSCSI initiators in a specific department only discover those iSCSI targets that are part of the department's discovery domain.

12.2 Installing iSNS Server for Linux

iSNS Server for Linux is included with SLES 10 SP2 and later, but is not installed or configured by default. You must install the iSNS package modules (`isns` and `yast2-isns` modules) and configure the iSNS service.

NOTE

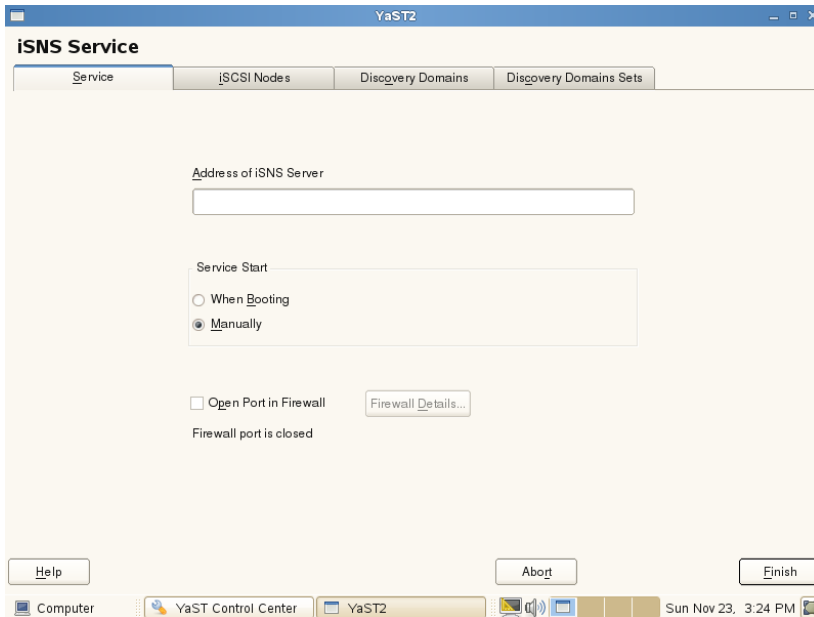
iSNS can be installed on the same server where iSCSI target or iSCSI initiator software is installed. Installing both the iSCSI target software and iSCSI initiator software on the same server is not supported.

To install iSNS for Linux:

- 1 Start YaST and select *Network Services* *iSNS Server*.
- 2 When prompted to install the `isns` package, click *Install*.

- 3 Follow the install dialog instructions to provide the SUSE Linux Enterprise Server 11 installation disks.

When the installation is complete, the iSNS Service configuration dialog opens automatically to the *Service* tab.



- 4 In *Address of iSNS Server*, specify the DNS name or IP address of the iSNS Server.
- 5 In *Service Start*, select one of the following:
 - **When Booting:** The iSNS service starts automatically on server startup.
 - **Manually (Default):** The iSNS service must be started manually by entering `rcisns start` or `/etc/init.d/isns start` at the server console of the server where you install it.
- 6 Specify the following firewall settings:

- **Open Port in Firewall:** Select the check box to open the firewall and allow access to the service from remote computers. The firewall port is closed by default.
- **Firewall Details:** If you open the firewall port, the port is open on all network interfaces by default. Click *Firewall Details* to select interfaces on which to open the port, select the network interfaces to use, then click *OK*.

7 Click *Finish* to apply the configuration settings and complete the installation.

8 Continue with Section 12.3, “Configuring iSNS Discovery Domains” (page 197).

12.3 Configuring iSNS Discovery Domains

In order for iSCSI initiators and targets to use the iSNS service, they must belong to a discovery domain.

IMPORTANT

The SNS service must be installed and running to configure iSNS discovery domains. For information, see Section 12.4, “Starting iSNS” (page 204).

- Section 12.3.1, “Creating iSNS Discovery Domains” (page 197)
- Section 12.3.2, “Creating iSNS Discovery Domain Sets” (page 199)
- Section 12.3.3, “Adding iSCSI Nodes to a Discovery Domain” (page 202)
- Section 12.3.4, “Adding Discovery Domains to a Discovery Domain Set” (page 203)

12.3.1 Creating iSNS Discovery Domains

A default discovery domain named *default DD* is automatically created when you install the iSNS service. The existing iSCSI targets and initiators that have been configured to use iSNS are automatically added to the default discovery domain.

To create a new discovery domain:

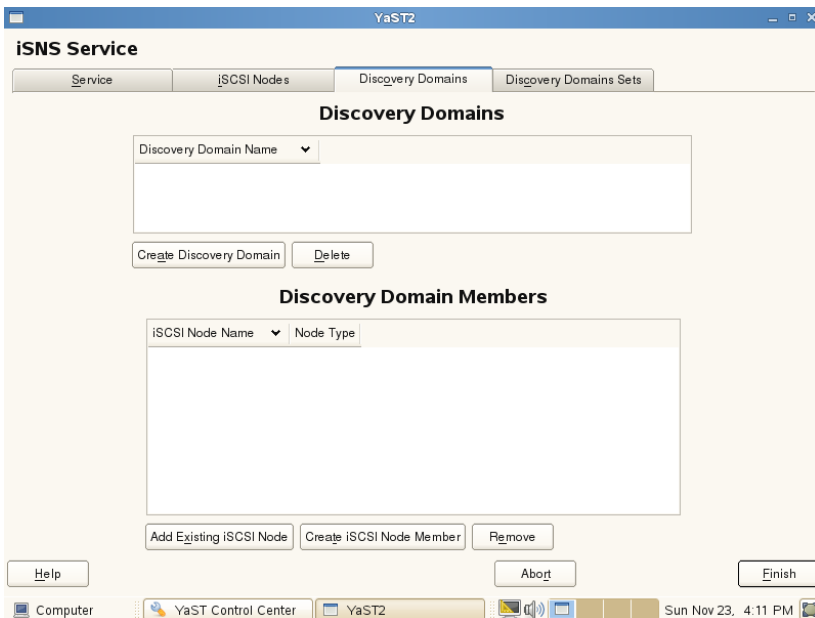
- 1 Start YaST and under *Network Services*, select *iSNS Server*.
- 2 Click the *Discovery Domains* tab.

The *Discovery Domains* area lists all discovery domains. You can create new discovery domains, or delete existing ones. Deleting a domain removes the members from the domain, but it does not delete the iSCSI node members.

The *Discovery Domain Members* area lists all iSCSI nodes assigned to a selected discovery domain. Selecting a different discovery domain refreshes the list with members from that discovery domain. You can add and delete iSCSI nodes from a selected discovery domain. Deleting an iSCSI node removes it from the domain, but it does not delete the iSCSI node.

Creating an iSCSI node allows a node that is not yet registered to be added as a member of the discovery domain. When the iSCSI initiator or target registers this node, then it becomes part of this domain.

When an iSCSI initiator performs a discovery request, the iSNS service returns all iSCSI node targets that are members of the same discovery domain.



- 3 Click the *Create Discovery Domain* button.

You can also select an existing discovery domain and click the *Delete* button to remove that discovery domain.

- 4 Specify the name of the discovery domain you are creating, then click *OK*.
- 5 Continue with Section 12.3.2, “Creating iSNS Discovery Domain Sets” (page 199).

12.3.2 Creating iSNS Discovery Domain Sets

Discovery domains must belong to a discovery domain set. You can create a discovery domain and add nodes to that discovery domain, but it is not active and the iSNS service does not function unless you add the discovery domain to a discovery domain set. A default discovery domain set named *default DDS* is automatically created when you install iSNS and the default discovery domain is automatically added to that domain set.

To create a discovery domain set:

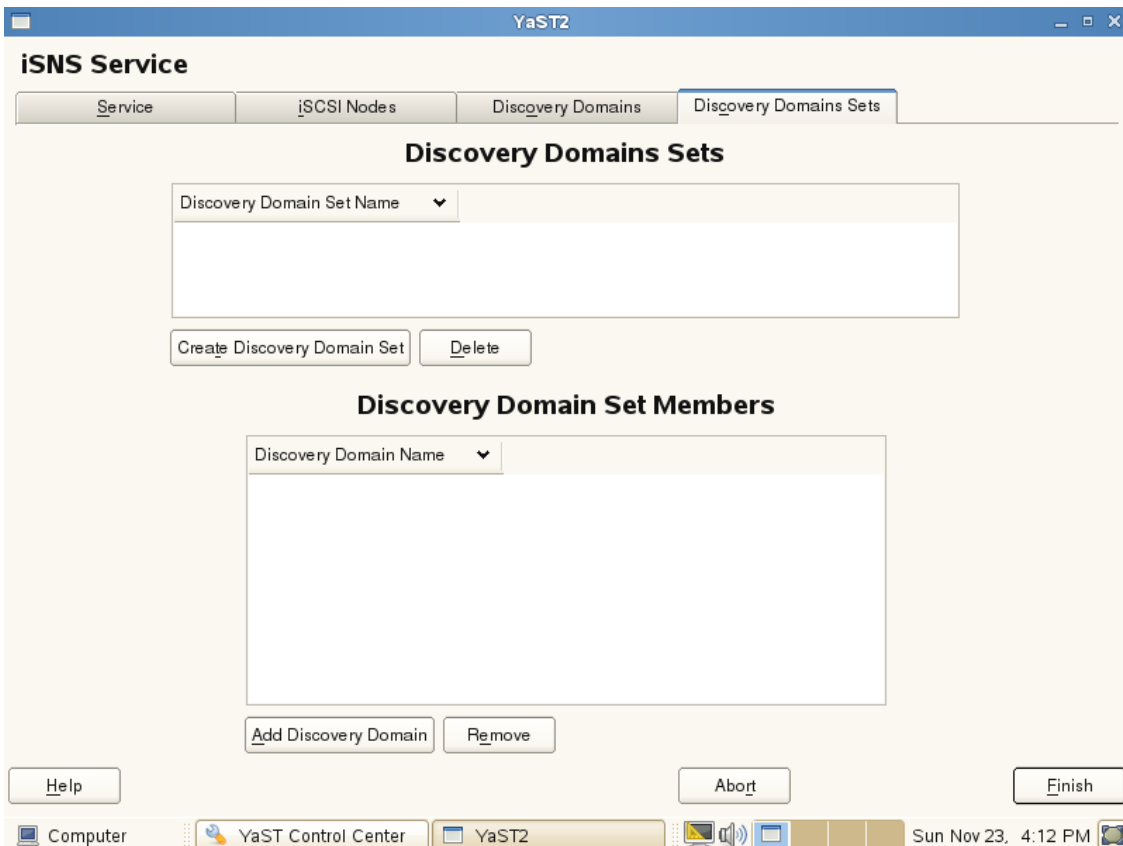
- 1 Start YaST and under *Network Services*, select *iSNS Server*.
- 2 Click the *Discovery Domains Sets* tab.

The *Discovery Domain Sets* area lists all of the discover domain sets. A discovery domain must be a member of a discovery domain set in order to be active.

In an iSNS database, a discovery domain set contains discovery domains, which in turn contains iSCSI node members.

The *Discovery Domain Set Members* area lists all discovery domains that are assigned to a selected discovery domain set. Selecting a different discovery domain set refreshes the list with members from that discovery domain set. You can add and delete discovery domains from a selected discovery domain set. Removing a discovery domain removes it from the domain set, but it does not delete the discovery domain.

Adding an discovery domain to a set allows a not yet registered iSNS discovery domain to be added as a member of the discovery domain set.



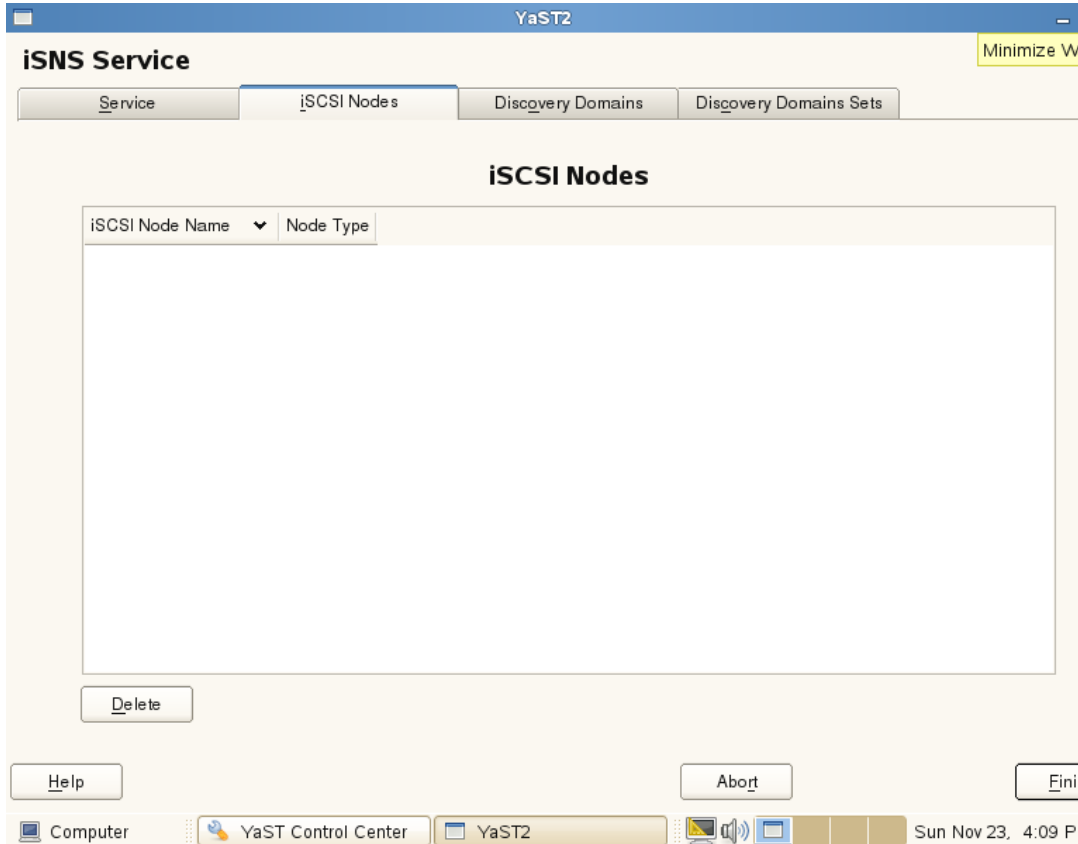
- 3 Click the *Create Discovery Domain Set* button.

You can also select an existing discovery domain set and click the *Delete* button to remove that discovery domain set.

- 4 Specify the name of the discovery domain set you are creating, then click *OK*.
- 5 Continue with Section 12.3.3, “Adding iSCSI Nodes to a Discovery Domain” (page 202).

12.3.3 Adding iSCSI Nodes to a Discovery Domain

- 1 Start YaST and under *Network Services*, select *iSNS Server*.
- 2 Click the *iSCSI Nodes* tab.



- 3 Review the list of nodes to ensure that the iSCSI targets and initiators that you want to use the iSNS service are listed.

If an iSCSI target or initiator is not listed, you might need to restart the iSCSI service on the node. You can do this by running the `rcopen-iscsi restart` command to restart an initiator or the `rciscsitarget restart` command to restart a target.

You can select an iSCSI node and click the *Delete* button to remove that node from the iSNS database. This is useful if you are no longer using an iSCSI node or have renamed it.

The iSCSI node is automatically added to the list (iSNS database) again when you restart the iSCSI service or reboot the server unless you remove or comment out the iSNS portion of the iSCSI configuration file.

- 4 Click the *Discovery Domains* tab, select the desired discovery domain, then click the *Display Members* button.
- 5 Click *Add existing iSCSI Node*, select the node you want to add to the domain, then click *Add Node*.
- 6 Repeat Step 5 (page 204) for as many nodes as you want to add to the discovery domain, then click *Done* when you are finished adding nodes.

An iSCSI node can belong to more than one discovery domain.

- 7 Continue with Section 12.3.4, “Adding Discovery Domains to a Discovery Domain Set” (page 203).

12.3.4 Adding Discovery Domains to a Discovery Domain Set

- 1 Start YaST and under *Network Services*, select *iSNS Server*.
- 2 Click the *Discovery Domains Set* tab.
- 3 Select *Create Discovery Domain Set* to add a new set to the list of discovery domain sets.
- 4 Choose a discovery domain set to modify.

- 5 Click *Add Discovery Domain*, select the discovery domain you want to add to the discovery domain set, then click *Add Discovery Domain*.
- 6 Repeat the last step for as many discovery domains as you want to add to the discovery domain set, then click *Done*.

A discovery domain can belong to more than one discovery domain set.

12.4 Starting iSNS

iSNS must be started at the server where you install it. Enter one of the following commands at a terminal console as the `root` user:

```
rcisns start
```

```
/etc/init.d/isns start
```

You can also use the `stop`, `status`, and `restart` options with iSNS.

iSNS can also be configured to start automatically each time the server is rebooted:

- 1 Start YaST and under *Network Services*, select *iSNS Server*.
- 2 With the *Service* tab selected, specify the IP address of your iSNS server, then click *Save Address*.
- 3 In the *Service Start* section of the screen, select *When Booting*.

You can also choose to start the iSNS server manually. You must then use the `rcisns start` command to start the service each time the server is restarted.

12.5 Stopping iSNS

iSNS must be stopped at the server where it is running. Enter one of the following commands at a terminal console as the `root` user:

```
rcisns stop
```

```
/etc/init.d/isns stop
```

12.6 For More Information

For information, see the *Linux iSNS for iSCSI project* [<http://sourceforge.net/projects/linuxisns/>]. The electronic mailing list for this project is *Linux iSNS - Discussion* [http://sourceforge.net/mailarchive/forum.php?forum_name=linuxisns-discussion].

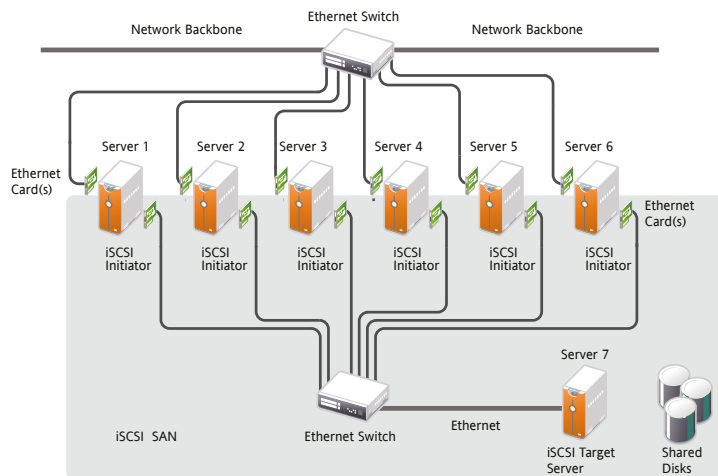
General information about iSNS is available in *RFC 4171: Internet Storage Name Service* [<http://www.ietf.org/rfc/rfc4171>].

Mass Storage over IP Networks: iSCSI

13

One of the central tasks in computer centers and when operating servers is providing hard disk capacity for server systems. Fibre Channel is often used for this purpose. iSCSI (Internet SCSI) solutions provide a lower-cost alternative to Fibre Channel that can leverage commodity servers and Ethernet networking equipment. Linux iSCSI provides iSCSI initiator and target software for connecting Linux servers to central storage systems.

Figure 13.1 *iSCSI SAN with an iSNS Server*



iSCSI is a storage networking protocol that facilitates data transfers of SCSI packets over TCP/IP networks between block storage devices and servers. iSCSI target software runs on the target server and defines the logical units as iSCSI target devices. iSCSI initiator software runs on different servers and connects to the target devices to make the storage devices available on that server.

IMPORTANT

It is not supported to run iSCSI target software and iSCSI initiator software on the same server in a production environment.

The iSCSI target and initiator servers communicate by sending SCSI packets at the IP level in your LAN. When an application running on the initiator server starts an inquiry for an iSCSI target device, the operating system produces the necessary SCSI commands. The SCSI commands are then embedded in IP packets and encrypted as necessary by software that is commonly known as the *iSCSI initiator*. The packets are transferred across the internal IP network to the corresponding iSCSI remote station, called the *iSCSI target*.

Many storage solutions provide access over iSCSI, but it is also possible to run a Linux server that provides an iSCSI target. In this case, it is important to set up a Linux server that is optimized for file system services. The iSCSI target accesses block devices in Linux. Therefore, it is possible to use RAID solutions to increase disk space as well as a lot of memory to improve data caching. For more information about RAID, also see Chapter 8, *Software RAID Configuration* (page 145).

- Section 13.1, “Installing iSCSI” (page 210)
- Section 13.2, “Setting Up an iSCSI Target” (page 211)
- Section 13.3, “Configuring iSCSI Initiator” (page 222)
- Section 13.4, “Using iSCSI Disks when Installing” (page 229)
- Section 13.5, “Troubleshooting iSCSI” (page 229)
- Section 13.6, “Additional Information” (page 232)

13.1 Installing iSCSI

YaST includes entries for iSCSI Target and iSCSI Initiator software, but the packages are not installed by default.

IMPORTANT

It is not supported to run iSCSI target software and iSCSI initiator software on the same server in a production environment.

- Section 13.1.1, “Installing iSCSI Target Software” (page 210)
- Section 13.1.2, “Installing the iSCSI Initiator Software” (page 210)

13.1.1 Installing iSCSI Target Software

Install the iSCSI target software on the server where you want to create iSCSI target devices.

- 1 Open YaST, and log in as the `root` user.
- 2 Select *Network Services* *iSCSI Target*.
- 3 When you are prompted to install the `iscsitarget` package, click *Install*.
- 4 Follow the on-screen install instructions, and provide the installation media as needed.

When the installation is complete, YaST opens to the iSCSI Target Overview page with the *Service* tab selected.

- 5 Continue with Section 13.2, “Setting Up an iSCSI Target” (page 211).

13.1.2 Installing the iSCSI Initiator Software

Install the iSCSI initiator software on each server where you want to access the target devices that you set up on the iSCSI target server.

- 1 Open YaST, and log in as the `root` user.
- 2 Select *Network Services* *iSCSI Initiator*.
- 3 When you are prompted to install the `open-iscsi` package, click *Install*.
- 4 Follow the on-screen install instructions, and provide the installation media as needed.

When the installation is complete, YaST opens to the iSCSI Initiator Overview page with the *Service* tab selected.

- 5 Continue with Section 13.3, “Configuring iSCSI Initiator” (page 222).

13.2 Setting Up an iSCSI Target

SUSE Linux Enterprise Server comes with an open source iSCSI target solution that evolved from the Ardis iSCSI target. A basic setup can be done with YaST, but to take full advantage of iSCSI, a manual setup is required.

- Section 13.2.1, “Preparing the Storage Space” (page 211)
- Section 13.2.2, “Creating iSCSI Targets with YaST” (page 213)
- Section 13.2.3, “Configuring an iSCSI Target Manually” (page 218)
- Section 13.2.4, “Configuring Online Targets with ietadm” (page 220)

13.2.1 Preparing the Storage Space

The iSCSI target configuration exports existing block devices to iSCSI initiators. You must prepare the storage space you want to use in the target devices by setting up unformatted partitions or devices by using the Partitioner in YaST, or by partitioning the devices from the command line.

IMPORTANT

After you set up a device or partition for use as an iSCSI target, you never access it directly via its local path. Do not specify a mount point for it when you create it.

- Section “Partitioning Devices” (page 212)
- Section “Partitioning Devices in a Virtual Environment” (page 213)

Partitioning Devices

- 1 Log in as the `root` user, then open YaST.
- 2 Select *SystemPartitioner*.
- 3 Click *Yes* to continue through the warning about using the Partitioner.
- 4 Click *Add* to create a partition, but do not format it, and do not mount it.

iSCSI targets can use unformatted partitions with Linux, Linux LVM, or Linux RAID file system IDs.

- 4a Select *Primary Partition*, then click *Next*.
 - 4b Specify the amount of space to use, then click *Next*.
 - 4c Select *Do not format*, then specify the file system ID type.
 - 4d Select *Do not mount*.
 - 4e Click *Finish*.
- 5 Repeat Step 4 (page 212) for each area that you want to use later as an iSCSI LUN.
- 6 Click *Accept* to keep your changes, then close YaST.

Partitioning Devices in a Virtual Environment

You can use a Xen guest server as the iSCSI target server. You must assign the storage space you want to use for the iSCSI storage devices to the guest virtual machine, then access the space as virtual disks within the guest environment. Each virtual disk can be a physical block device, such as an entire disk, partition, or volume, or it can be a file-backed disk image where the virtual disk is a single image file on a larger physical disk on the Xen host server. For the best performance, create each virtual disk from a physical disk or a partition. After you set up the virtual disks for the guest virtual machine, start the guest server, then configure the new blank virtual disks as iSCSI target devices by following the same process as for a physical server.

File-backed disk images are created on the Xen host server, then assigned to the Xen guest server. By default, Xen stores file-backed disk images in the `/var/lib/xen/images/vm_name` directory, where `vm_name` is the name of the virtual machine.

For example, if you want to create the disk image `/var/lib/xen/images/vm_one/xen-0` with a size of 4 GB, first ensure that the directory is there, then create the image itself.

- 1 Log in to the host server as the `root` user.
- 2 At a terminal console prompt, enter the following commands

```
mkdir -p /var/lib/xen/images/vm_one
dd if=/dev/zero of=/var/lib/xen/images/vm_one/xen-0 seek=1M bs=4096 count=1
```

- 3 Assign the file system image to the guest virtual machine in the Xen configuration file.
- 4 Log in as the `root` user on the guest server, then use YaST to set up the virtual block device by using the process in Section “Partitioning Devices” (page 212).

13.2.2 Creating iSCSI Targets with YaST

To configure the iSCSI target, run the *iSCSI Target* module in YaST. The configuration is split into three tabs. In the *Service* tab, select the start mode and the firewall settings. If you want to access the iSCSI target from a remote machine, select *Open Port in*

Firewall. If an iSNS server should manage the discovery and access control, activate *iSNS Access Control* and enter the IP address of your iSNS server. You cannot use hostnames, but must use the IP address. For more about iSNS, read Chapter 12, *iSNS for Linux* (page 193).

The *Global* tab provides settings for the iSCSI server. The authentication set here is used for the discovery of services, not for accessing the targets. If you do not want to restrict the access to the discovery, use *No Authentication*.

If authentication is needed, there are two possibilities to consider. One is that an initiator must prove that it has the permissions to run a discovery on the iSCSI target. This is done with *Incoming Authentication*. The other is that the iSCSI target must prove to the initiator that it is the expected target. Therefore, the iSCSI target can also provide a user name and password. This is done with *Outgoing Authentication*. Find more information about authentication in *RFC 3720* [<http://www.ietf.org/rfc/rfc3720.txt>].

The targets are defined in the *Targets* tab. Use *Add* to create a new iSCSI target. The first dialog asks for information about the device to export.

Target

The *Target* line has a fixed syntax that looks like the following:

```
iqn.yyyy-mm.<reversed domain name>:unique_id
```

It always starts with *iqn*. *yyyy-mm* is the format of the date when this target is activated. Find more about naming conventions in *RFC 3722* [<http://www.ietf.org/rfc/rfc3722.txt>].

Identifier

The *Identifier* is freely selectable. It should follow some scheme to make the whole system more structured.

LUN

It is possible to assign several LUNs to a target. To do this, select a target in the *Targets* tab, then click *Edit*. Then, add new LUNs to an existing target.

Path

Add the path to the block device or file system image to export.

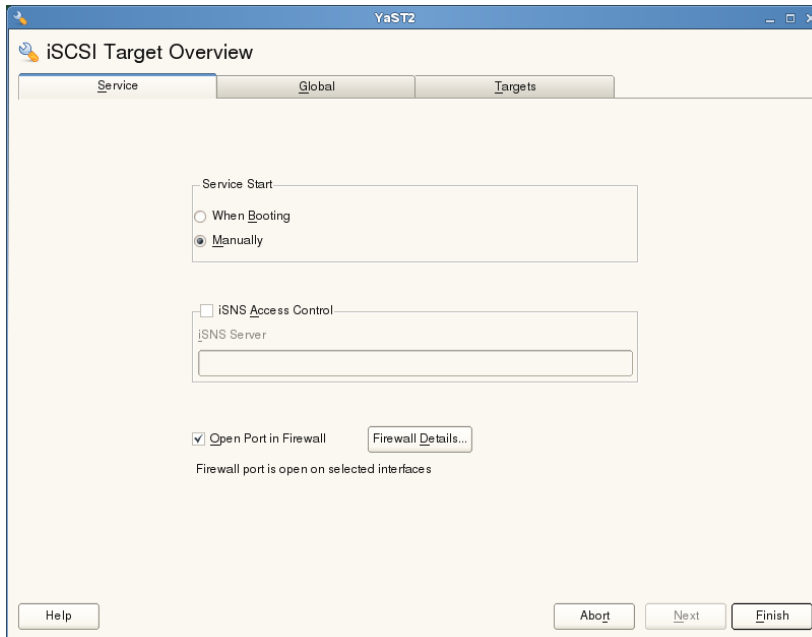
The next menu configures the access restrictions of the target. The configuration is very similar to the configuration of the discovery authentication. In this case, at least an incoming authentication should be setup.

Next finishes the configuration of the new target, and brings you back to the overview page of the *Target* tab. Activate your changes by clicking *Finish*.

To create a target device:

- 1 Open YaST, and log in as the `root` user.
- 2 Select *Network Services* *iSCSI Target*.

YaST opens to the iSCSI Target Overview page with the *Service* tab selected.

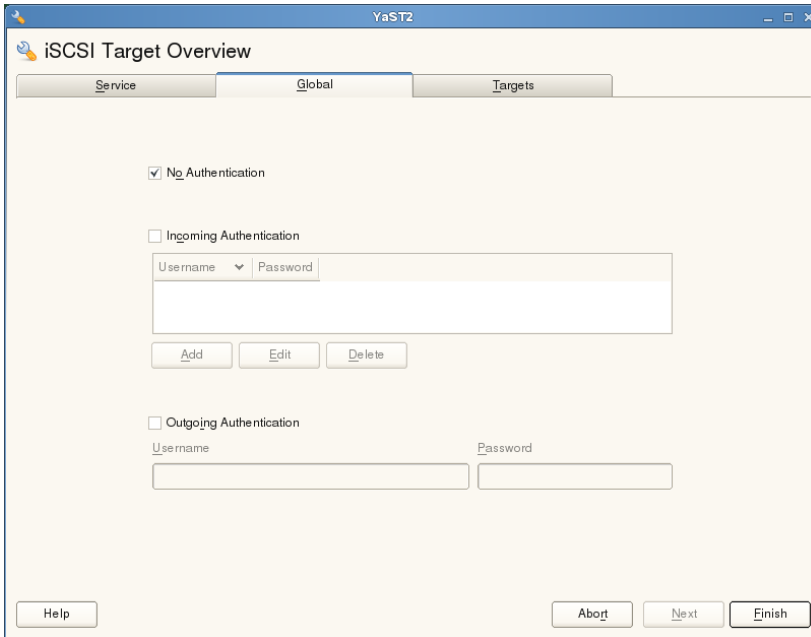


- 3 In the *Service Start* area, select one of the following:

- **When booting:** Automatically start the initiator service on subsequent server re-boots.

- **Manually (default):** Start the service manually.
- 4 If you are using iSNS for target advertising, select the *iSNS Access Control* check box, then type the IP address.
 - 5 If desired, open the firewall ports to allow access to the server from remote computers.
 - 5a Select the *Open Port in Firewall* check box.
 - 5b Specify the network interfaces where you want to open the port by clicking *Firewall Details*, selecting the check box next to a network interface to enable it, then clicking *OK* to accept the settings.
 - 6 If authentication is required to connect to target devices you set up on this server, select the *Global* tab, deselect *No Authentication* to enable authentication, then specify the necessary credentials for incoming and outgoing authentication.

The *No Authentication* option is enabled by default. For a more secure configuration, you can specify authentication for incoming, outgoing, or both incoming and outgoing. You can also specify multiple sets of credentials for incoming authentication by adding pairs of user names and passwords to the list under *Incoming Authentication*.



7 Configure the iSCSI target devices.

7a Select the *Targets* tab.

7b If you have not already done so, select and delete the example iSCSI target from the list, then confirm the deletion by clicking *Continue*.

7c Click *Add* to add a new iSCSI target.

The iSCSI target automatically presents an unformatted partition or block device and completes the Target and Identifier fields.

7d You can accept this, or browse to select a different space.

You can also subdivide the space to create LUNs on the device by clicking *Add* and specifying sectors to allocate to that LUN. If you need additional options for these LUNs, select *Expert Settings*.

7e Click *Next*

7f Repeat Step 7c (page 217) to Step 7e (page 217) for each iSCSI target device you want to create.

7g (Optional) On the *Service* tab, click *Save* to export the information about the configured iSCSI targets to a file.

This makes it easier to later provide this information to consumers of the resources.

7h Click *Finish* to create the devices, then click *Yes* to restart the iSCSI software stack.

13.2.3 Configuring an iSCSI Target Manually

Configure an iSCSI target in `/etc/ietd.conf`. All parameters in this file before the first *Target* declaration are global for the file. Authentication information in this portion has a special meaning—it is not global, but is used for the discovery of the iSCSI target.

If you have access to an iSNS server, you should first configure the file to tell the target about this server. The address of the iSNS server must always be given as an IP address. You cannot specify the DNS name for the server. The configuration for this functionality looks like the following:

```
iSNSServer 192.168.1.111
iSNSAccessControl no
```

This configuration makes the iSCSI target register itself with the iSNS server, which in turn provides the discovery for initiators. For more about iSNS, see Chapter 12, *iSNS for Linux* (page 193). The access control for the iSNS discovery is not supported. Just keep `iSNSAccessControl no`.

All direct iSCSI authentication can be done in two directions. The iSCSI target can require the iSCSI initiator to authenticate with the `IncomingUser`, which can be added multiple times. The iSCSI initiator can also require the iSCSI target to authenticate. Use `OutgoingUser` for this. Both have the same syntax:

```
IncomingUser <username> <password>
OutgoingUser <username> <password>
```

The authentication is followed by one or more target definitions. For each target, add a `Target` section. This section always starts with a `Target` identifier followed, by definitions of logical unit numbers:

```
Target iqn.yyyy-mm.<reversed domain name>[:identifier]
    Lun 0 Path=/dev/mapper/system-v3
    Lun 1 Path=/dev/hda4
    Lun 2 Path=/var/lib/xen/images/xen-1,Type=fileio
```

In the `Target` line, `yyyy-mm` is the date when this target is activated, and `identifier` is freely selectable. Find more about naming conventions in *RFC 3722* [<http://www.ietf.org/rfc/rfc3722.txt>]. Three different block devices are exported in this example. The first block device is a logical volume (see also Chapter 4, *LVM Configuration* (page 33)), the second is an IDE partition, and the third is an image available in the local file system. All these look like block devices to an iSCSI initiator.

Before activating the iSCSI target, add at least one `IncomingUser` after the `Lun` definitions. It does the authentication for the use of this target.

To activate all your changes, restart the `iscsitarget` daemon with `rcopen-iscsi restart`. Check your configuration in the `/proc` file system:

```
cat /proc/net/iet/volume
tid:1 name:iqn.2006-02.com.example.iserv:systems
    lun:0 state:0 iotype:fileio path:/dev/mapper/system-v3
    lun:1 state:0 iotype:fileio path:/dev/hda4
    lun:2 state:0 iotype:fileio path:/var/lib/xen/images/xen-1
```

There are many more options that control the behavior of the iSCSI target. For more information, see the man page of `ietd.conf`.

Active sessions are also displayed in the `/proc` file system. For each connected initiator, an extra entry is added to `/proc/net/iet/session`:

```
cat /proc/net/iet/session
tid:1 name:iqn.2006-02.com.example.iserv:system-v3
    sid:562949957419520
initiator:iqn.2005-11.de.suse:cn=rome.example.com,01.9ff842f5645
    cid:0 ip:192.168.178.42 state:active hd:none dd:none
    sid:281474980708864 initiator:iqn.2006-02.de.suse:01.6f7259c88b70
    cid:0 ip:192.168.178.72 state:active hd:none dd:none
```

13.2.4 Configuring Online Targets with ietadm

When changes to the iSCSI target configuration are necessary, you must always restart the target to activate changes that are done in the configuration file. Unfortunately, all active sessions are interrupted in this process. To maintain an undisturbed operation, the changes should be done in the main configuration file `/etc/ietd.conf`, but also made manually to the current configuration with the administration utility `ietadm`.

To create a new iSCSI target with a LUN, first update your configuration file. The additional entry could be:

```
Target ign.2006-02.com.example.iserv:system2
    Lun 0 Path=/dev/mapper/system-swap2
    IncomingUser joe secret
```

To set up this configuration manually, proceed as follows:

- 1 Create a new target with the command `ietadm --op new --tid=2 --params Name=ign.2006-02.com.example.iserv:system2`.
- 2 Add a logical unit with `ietadm --op new --tid=2 --lun=0 --params Path=/dev/mapper/system-swap2`.
- 3 Set the user name and password combination on this target with `ietadm --op new --tid=2 --user --params=IncomingUser=joe,Password=secret`.
- 4 Check the configuration with `cat /proc/net/iet/volume`.

It is also possible to delete active connections. First, check all active connections with the command `cat /proc/net/iet/session`. This might look like:

```
cat /proc/net/iet/session
tid:1 name:ign.2006-03.com.example.iserv:system
    sid:281474980708864 initiator:ign.1996-04.com.example:01.82725735af5
    cid:0 ip:192.168.178.72 state:active hd:none dd:none
```

To delete the session with the session ID `281474980708864`, use the command `ietadm --op delete --tid=1 --sid=281474980708864 --cid=0`. Be aware

that this makes the device inaccessible on the client system and processes accessing this device are likely to hang.

`ietadm` can also be used to change various configuration parameters. Obtain a list of the global variables with `ietadm --op show --tid=1 --sid=0`. The output looks like:

```
InitialR2T=Yes
ImmediateData=Yes
MaxConnections=1
MaxRecvDataSegmentLength=8192
MaxXmitDataSegmentLength=8192
MaxBurstLength=262144
FirstBurstLength=65536
DefaultTime2Wait=2
DefaultTime2Retain=20
MaxOutstandingR2T=1
DataPDUIInOrder=Yes
DataSequenceInOrder=Yes
ErrorRecoveryLevel=0
HeaderDigest=None
DataDigest=None
OFMarker=No
IFMarker=No
OFMarkInt=Reject
IFMarkInt=Reject
```

All of these parameters can be easily changed. For example, if you want to change the maximum number of connections to two, use

```
ietadm --op update --tid=1 --params=MaxConnections=2.
```

In the file `/etc/ietd.conf`, the associated line should look like `MaxConnections 2`.

WARNING

The changes that you make with the `ietadm` utility are not permanent for the system. These changes are lost at the next reboot if they are not added to the `/etc/ietd.conf` configuration file. Depending on the usage of iSCSI in your network, this might lead to severe problems.

There are several more options available for the `ietadm` utility. Use `ietadm -h` to find an overview. The abbreviations there are target ID (`tid`), session ID (`sid`), and connection ID (`cid`). They can also be found in `/proc/net/iet/session`.

13.3 Configuring iSCSI Initiator

The iSCSI initiator, also called an iSCSI client, can be used to connect to any iSCSI target. This is not restricted to the iSCSI target solution explained in Section 13.2, “Setting Up an iSCSI Target” (page 211). The configuration of iSCSI initiator involves two major steps: the discovery of available iSCSI targets and the setup of an iSCSI session. Both can be done with YaST.

- Section 13.3.1, “Using YaST for the iSCSI Initiator Configuration” (page 222)
- Section 13.3.2, “Setting Up the iSCSI Initiator Manually” (page 227)
- Section 13.3.3, “The iSCSI Client Databases” (page 228)

13.3.1 Using YaST for the iSCSI Initiator Configuration

The iSCSI Initiator Overview in YaST is divided into three tabs:

- **Service:** The *Service* tab can be used to enable the iSCSI initiator at boot time. It also offers to set a unique *Initiator Name* and an iSNS server to use for the discovery. The default port for iSNS is 3205.
- **Connected Targets:** The *Connected Targets* tab gives an overview of the currently connected iSCSI targets. Like the *Discovered Targets* tab, it also gives the option to add new targets to the system.

On this page, you can select a target device, then toggle the start-up setting for each iSCSI target device:

- **Automatic:** This option is used for iSCSI targets that are to be connected when the iSCSI service itself starts up. This is the typical configuration.

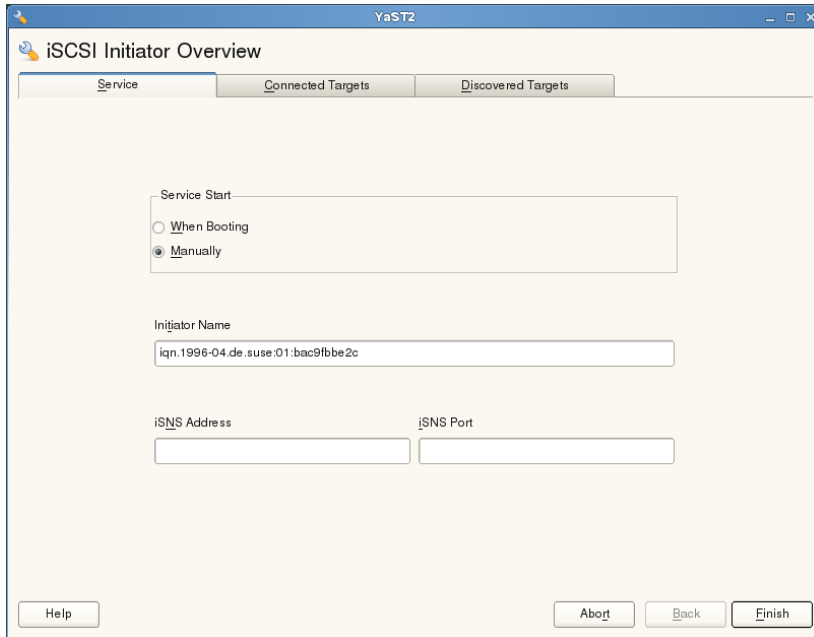
- **Onboot:** This option is used for iSCSI targets that are to be connected during boot; that is, when root (/) is on iSCSI. As such, the iSCSI target device will be evaluated from the initrd on server boots.
- **Discovered Targets:** *Discovered Targets* provides the possibility of manually discovering iSCSI targets in the network.
- Section “Configuring the iSCSI Initiator” (page 223)
- Section “Discovering iSCSI Targets by Using iSNS” (page 225)
- Section “Discovering iSCSI Targets Manually” (page 225)
- Section “Setting the Start-up Preference for iSCSI Target Devices” (page 226)

Configuring the iSCSI Initiator

1 Open YaST, and log in as the `root` user.

2 Select *Network Services* *iSCSI Initiator*.

YaST opens to the iSCSI Initiator Overview page with the *Service* tab selected.



3 In the *Service Start* area, select one of the following:

- **When booting:** Automatically start the initiator service on subsequent server re-boots.
- **Manually (default):** Start the service manually.

4 Specify or verify the *Initiator Name*.

Specify a well-formed iSCSI qualified name (IQN) for the iSCSI initiator on this server. The initiator name must be globally unique on your network. The IQN uses the following general format:

```
iqn.yyyy-mm.com.mycompany:n1:n2
```

where n1 and n2 are alphanumeric characters. For example:

```
iqn.1996-04.de.suse:01:9c83a3e15f64
```

The *Initiator Name* is automatically completed with the corresponding value from the `/etc/iscsi/initiatorname.iscsi` file on the server.

If the server has iBFT (iSCSI Boot Firmware Table) support, the *Initiator Name* is completed with the corresponding value in the IBFT, and you are not able to change the initiator name in this interface. Use the BIOS Setup to modify it instead. The iBFT is a block of information containing various parameters useful to the iSCSI boot process, including iSCSI target and initiator descriptions for the server.

- 5 Use either of the following methods to discover iSCSI targets on the network.
 - **iSNS:** To use iSNS (Internet Storage Name Service) for discovering iSCSI targets, continue with Section “Discovering iSCSI Targets by Using iSNS” (page 225).
 - **Discovered Targets:** To discover iSCSI target devices manually, continue with Section “Discovering iSCSI Targets Manually” (page 225).

Discovering iSCSI Targets by Using iSNS

Before you can use this option, you must have already installed and configured an iSNS server in your environment. For information, see Chapter 12, *iSNS for Linux* (page 193).

- 1 In YaST, select *iSCSI Initiator*, then select the *Service* tab.
- 2 Specify the IP address of the iSNS server and port.

The default port is 3205.
- 3 On the iSCSI Initiator Overview page, click *Finish* to save and apply your changes.

Discovering iSCSI Targets Manually

Repeat the following process for each of the iSCSI target servers that you want to access from the server where you are setting up the iSCSI initiator.

- 1 In YaST, select *iSCSI Initiator*, then select the *Discovered Targets* tab.
- 2 Click *Discovery* to open the *iSCSI Initiator Discovery* dialog.

- 3 Enter the IP address and change the port if needed.

The default port is 3260.

- 4 If authentication is required, deselect *No Authentication*, then specify the credentials the *Incoming* or *Outgoing* authentication.
- 5 Click *Next* to start the discovery and connect to the iSCSI target server.
- 6 If credentials are required, after a successful discovery, use *Login* to activate the target.

You are prompted for authentication credentials to use the selected iSCSI target.

- 7 Click *Next* to finish the configuration.

If everything went well, the target now appears in *Connected Targets*.

The virtual iSCSI device is now available.

- 8 On the iSCSI Initiator Overview page, click *Finish* to save and apply your changes.
- 9 You can find the local device path for the iSCSI target device by using the `lsscsi` command:

```
lsscsi
[1:0:0:0]    disk      IET          VIRTUAL-DISK    0          /dev/sda
```

Setting the Start-up Preference for iSCSI Target Devices

- 1 In YaST, select *iSCSI Initiator*, then select the *Connected Targets* tab to view a list of the iSCSI target devices that are currently connected to the server.
- 2 Select the iSCSI target device that you want to manage.
- 3 Click *Toggle Start-Up* to modify the setting:
 - **Automatic:** This option is used for iSCSI targets that are to be connected when the iSCSI service itself starts up. This is the typical configuration.

- **Onboot:** This option is used for iSCSI targets that are to be connected during boot; that is, when root (/) is on iSCSI. As such, the iSCSI target device will be evaluated from the `initrd` on server boots.

4 Click *Finish* to save and apply your changes.

13.3.2 Setting Up the iSCSI Initiator Manually

Both the discovery and the configuration of iSCSI connections require a running `iscsid`. When running the discovery the first time, the internal database of the iSCSI initiator is created in the directory `/var/lib/open-iscsi`.

If your discovery is password protected, provide the authentication information to `iscsid`. Because the internal database does not exist when doing the first discovery, it cannot be used at this time. Instead, the configuration file `/etc/iscsid.conf` must be edited to provide the information. To add your password information for the discovery, add the following lines to the end of `/etc/iscsid.conf`:

```
discovery.sendtargets.auth.authmethod = CHAP
discovery.sendtargets.auth.username = <username>
discovery.sendtargets.auth.password = <password>
```

The discovery stores all received values in an internal persistent database. In addition, it displays all detected targets. Run this discovery with the command `iscsiadm -m discovery --type=st --portal=<targetip>`. The output should look like:

```
149.44.171.99:3260,1 iqn.2006-02.com.example.iserv:systems
```

To discover the available targets on a iSNS server, use the command `iscsiadm --mode discovery --type isns --portal <targetip>`

For each target defined on the iSCSI target, one line appears. For more information about the stored data, see Section 13.3.3, “The iSCSI Client Databases” (page 228).

The special `--login` option of `iscsiadm` creates all needed devices:

```
iscsiadm -m node -n iqn.2006-02.com.example.iserv:systems --login
```

The newly generated devices show up in the output of `lsscsi` and can now be accessed by mount.

13.3.3 The iSCSI Client Databases

All information that was discovered by the iSCSI initiator is stored in two database files that reside in `/var/lib/open-iscsi`. There is one database for the discovery of targets and one for the discovered nodes. When accessing a database, you first must select if you want to get your data from the discovery or from the node database. Do this with the `-m discovery` and `-m node` parameters of `iscsiadm`. Using `iscsiadm` just with one of these parameters gives an overview of the stored records:

```
iscsiadm -m discovery
149.44.171.99:3260,1 iqn.2006-02.com.example.iserv:systems
```

The target name in this example is

`iqn.2006-02.com.example.iserv:systems`. This name is needed for all actions that relate to this special data set. To examine the content of the data record with the ID `iqn.2006-02.com.example.iserv:systems`, use the following command:

```
iscsiadm -m node --targetname iqn.2006-02.com.example.iserv:systems
node.name = iqn.2006-02.com.example.iserv:systems
node.transport_name = tcp
node.tpgt = 1
node.active_conn = 1
node.startup = manual
node.session.initial_cmds_n = 0
node.session.reopen_max = 32
node.session.auth.authmethod = CHAP
node.session.auth.username = joe
node.session.auth.password = *****
node.session.auth.username_in = <empty>
node.session.auth.password_in = <empty>
node.session.timeo.replacement_timeout = 0
node.session.err_timeo.abort_timeout = 10
node.session.err_timeo.reset_timeout = 30
node.session.iscsi.InitialR2T = No
node.session.iscsi.ImmediateData = Yes
....
```

To edit the value of one of these variables, use the command `iscsiadm` with the `update` operation. For example, if you want `iscsid` to log in to the iSCSI target when it initializes, set the variable `node.startup` to the value `automatic`:

```
iscsiadm -m node -n iqn.2006-02.com.example.iserv:systems --op=update
--name=node.startup --value=automatic
```

Remove obsolete data sets with the `delete` operation If the target `iqn.2006-02.com.example.iserv:systems` is no longer a valid record, delete this record with the following command:

```
iscsiadm -m node -n iqn.2006-02.com.example.iserv:systems --op=delete
```

IMPORTANT

Use this option with caution because it deletes the record without any additional confirmation prompt.

To get a list of all discovered targets, run the `iscsiadm -m node` command.

13.4 Using iSCSI Disks when Installing

Bootting from an iSCSI disk on i386, x86_64, and ppc64 architectures is supported, when iSCSI enabled firmware is used.

To use iSCSI disks during installation, it is necessary to add the following parameter to the boot option line:

```
withiscsi=1
```

During installation, an additional screen appears that provides the option to attach iSCSI disks to the system and use them in the installation process.

13.5 Troubleshooting iSCSI

- Section 13.5.1, “Hotplug Doesn’t Work for Mounting iSCSI Targets” (page 230)

- Section 13.5.2, “Data Packets Dropped for iSCSI Traffic” (page 230)
- Section 13.5.3, “Using iSCSI Volumes with LVM” (page 230)
- Section 13.5.4, “iSCSI Targets Are Mounted When the Configuration File Is Set to Manual” (page 231)

13.5.1 Hotplug Doesn’t Work for Mounting iSCSI Targets

In SLES 10, you could add the `hotplug` option to your device in the `/etc/fstab` file to mount iSCSI targets. For example:

```
/dev/disk/by-uuid-blah /oracle/db ext3 hotplug,rw 0 2
```

For SLES 11, the `hotplug` option no longer works. Use the `nofail` option instead. For example:

```
/dev/sdb1 /mnt/mountpoint ext3 acl,user,nofail 0 0
```

For information, see *TID 7004427: /etc/fstab entry does not mount iSCSI device on boot up* [<http://www.novell.com/support/php/search.do?cmd=displayKC&docType=kc&externalId=7004427>].

13.5.2 Data Packets Dropped for iSCSI Traffic

A firewall might drop packets if it gets too busy. The default for the SUSE Firewall is to drop packets after three minutes. If you find that iSCSI traffic packets are being dropped, you can consider configuring the SUSE Firewall to queue packets instead of dropping them when it gets too busy.

13.5.3 Using iSCSI Volumes with LVM

Use the troubleshooting tips in this section when using LVM on iSCSI targets.

- Section “Check the iSCSI Initiator Discovery Occurs at Boot” (page 231)
- Section “Check that iSCSI Target Discovery Occurs at Boot” (page 231)

Check the iSCSI Initiator Discovery Occurs at Boot

When you set up the iSCSI Initiator, ensure that you enable discovery at boot time so that `udev` can discover the iSCSI devices at boot time and set up the devices to be used by LVM.

Check that iSCSI Target Discovery Occurs at Boot

Remember that `udev` provides the default setup for devices in SLES 11. Ensure that all of the applications that create devices have a Runlevel setting to run at boot so that `udev` can recognize and assign devices for them at system startup. If the application or service is not started until later, `udev` does not create the device automatically as it would at boot time.

You can check your runlevel settings for LVM2 and iSCSI in *YaST* by going to *SystemSystem Services (Runlevel)Expert Mode*. The following services should be enabled at boot (B):

```
boot.lvm
boot.open-iscsi
open-iscsi
```

13.5.4 iSCSI Targets Are Mounted When the Configuration File Is Set to Manual

When Open-iSCSI starts, it can mount the targets even if the option `node.startup` option is set to manual in the `/etc/iscsi/iscsid.conf` file if you manually modified the configuration file.

Check the `/etc/iscsi/nodes/<target_name>/<ip_address,port>/default` file. It contains a `node.startup` setting that overrides the `/etc/iscsi/iscsid.conf` file. Setting the mount option to manual by using the YaST interface

also sets the `node.startup = manual` in the `/etc/iscsi/nodes/<target_name>/<ip_address,port>/default` files.

13.6 Additional Information

The iSCSI protocol has been available for several years. There are many reviews and additional documentation comparing iSCSI with SAN solutions, doing performance benchmarks, or just describing hardware solutions. Important pages for more information about open-iscsi are:

- *Open-iSCSI Project* [<http://www.open-iscsi.org/>]
- *AppNote: iFolder on Open Enterprise Server Linux Cluster using iSCSI* [<http://www.novell.com/coolsolutions/appnote/15394.html>]

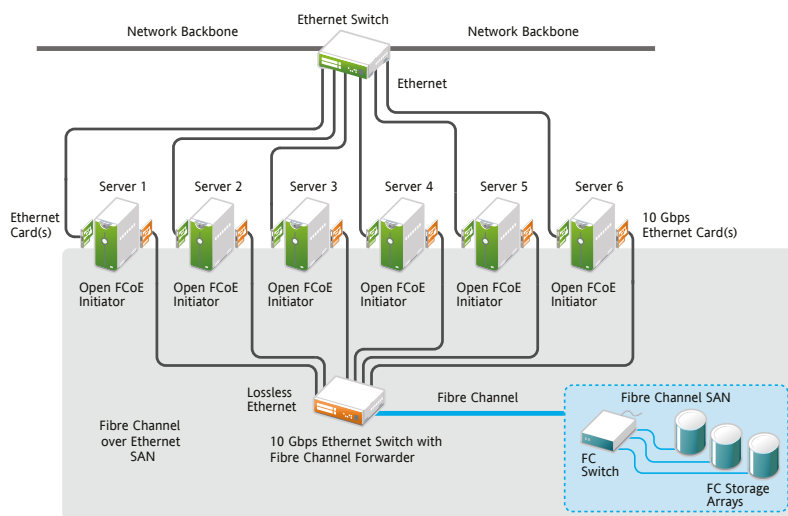
There is also some online documentation available. See the man pages for `iscsiadm`, `iscsid`, `ietd.conf`, and `ietd` and the example configuration file `/etc/iscsid.conf`.

Fibre Channel Storage over Ethernet Networks: FCoE

14

Many enterprise data centers rely on Ethernet for their LAN and data traffic, and on Fibre Channel networks for their storage infrastructure. Open Fibre Channel over Ethernet (FCoE) Initiator software allows servers with Ethernet adapters to connect to a Fibre Channel storage subsystem over an Ethernet network. This connectivity was previously reserved exclusively for systems with Fibre Channel adapters over a Fibre Channel fabric. The FCoE technology reduces complexity in the data center by aiding network convergence. This helps to preserve your existing investments in a Fibre Channel storage infrastructure and to simplify network management.

Figure 14.1 *Open Fibre Channel over Ethernet SAN*



Open-FCoE allows you to run the Fibre Channel protocols on the host, instead of on proprietary hardware on the host bus adapter. It is targeted for 10 Gbps (gigabit per second) Ethernet adapters, but can work on any Ethernet adapter that supports pause frames. The initiator software provides a Fibre Channel protocol processing module as well as an Ethernet based transport module. The Open-FCoE module acts as a low-level driver for SCSI. The Open-FCoE transport uses `net_device` to send and receive packets. Data Center Bridging (DCB) drivers provide the quality of service for FCoE.

FCoE is an encapsulation protocol that moves the Fibre Channel protocol traffic over Ethernet connections without changing the Fibre Channel frame. This allows your network security and traffic management infrastructure to work the same with FCoE as it does with Fibre Channel.

You might choose to deploy Open-FCoE in your enterprise if the following conditions exist:

- Your enterprise already has a Fibre Channel storage subsystem and administrators with Fibre Channel skills and knowledge.
- You are deploying 10 Gbps Ethernet in the network.

This section describes how to set up FCoE in your network.

- Section 14.1, “Installing FCoE and the YaST2 FCoE Client” (page 236)
- Section 14.2, “Configuring FCoE Interfaces during the Installation” (page 237)
- Section 14.3, “Managing FCoE Services with YaST” (page 238)
- Section 14.4, “Configuring FCoE with Commands” (page 243)
- Section 14.5, “Managing FCoE Instances with the FCoE Administration Tool” (page 245)
- Section 14.6, “Setting Up Partitions for an FCoE Initiator Disk” (page 249)
- Section 14.7, “Creating a File System on an FCoE Initiator Disk” (page 250)
- Section 14.8, “Additional Information” (page 251)

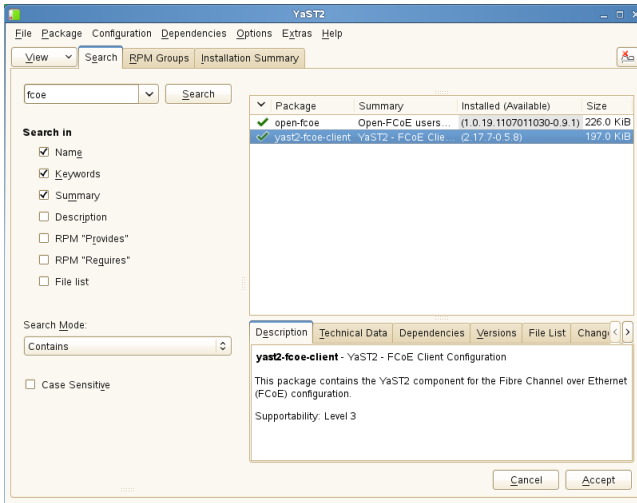
14.1 Installing FCoE and the YaST2 FCoE Client

You can set up FCoE disks in your storage infrastructure by enabling FCoE at the switch for the connections to a server. If FCoE disks are available when the SUSE Linux Enterprise Server operating system is installed, the FCoE Initiator software is automatically installed at that time.

If the FCoE Initiator software and YaST2 FCoE Client software are not installed, use the following procedure to manually install them on an existing system:

- 1 Log in to the server as the `root` user.
- 2 In YaST2, select *Software Management*.
- 3 Search for and select the following FCoE packages:
 - `open-fcoe`
 - `yast2-fcoe-client`

For example, type `fcoe` in the *Search* field, click *Search* to locate the software packages, then select the check box next to each software package that you want to install.



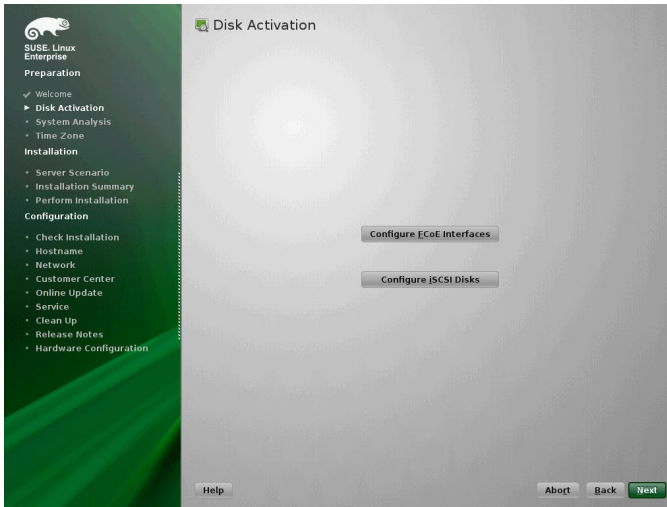
- 4 Click *Accept*, then click *Continue* to accept the automatic changes.

14.2 Configuring FCoE Interfaces during the Installation

The YaST2 installation for SUSE Linux Enterprise Server allows you to configure FCoE disks during the operating system installation if FCoE is enabled at the switch for the connections between the server and the Fibre Channel storage infrastructure. Some system BIOS types can automatically detect the FCoE disks, and report the disks to the YaST2 Installation software. However, automatic detection of FCoE disks is not supported by all BIOS types. To enable automatic detection in this case, you can add the `withfcoe` option to the kernel command line when you begin the installation:

```
withfcoe=1
```

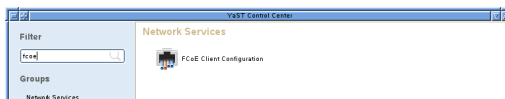
When the FCoE disks are detected, the YaST2 installation offers the option to configure FCoE instances at that time. On the Disk Activation page, select *Configure FCoE Interfaces* to access the FCoE configuration. For information about configuring the FCoE interfaces, see Section 14.3, “Managing FCoE Services with YaST” (page 238).



14.3 Managing FCoE Services with YaST

You can use the YaST2 FCoE Client Configuration option to create, configure, and remove FCoE interfaces for the FCoE disks in your Fibre Channel storage infrastructure. To use this option, the FCoE Initiator service (the `fcoemon` daemon) and the Link Layer Discovery Protocol agent daemon (`lldpad`) must be installed and running, and the FCoE connections must be enabled at the FCoE-capable switch.

- 1 Log in as the `root` user, then launch YaST2.
- 2 In YaST, select *Network ServicesFCoE Client Configuration*.

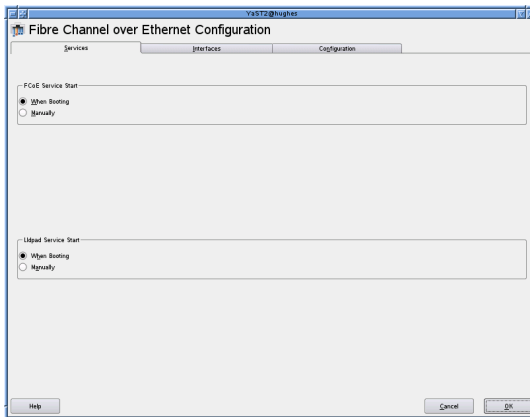


The *Fibre Channel over Ethernet Configuration* dialog box provides three tabs:

- *Services*

- *Interfaces*
- *Configuration*

- 3 On the *Services* tab, view or modify the FCoE service and Lldpad (Link Layer Discovery Protocol agent daemon) service start time as necessary.



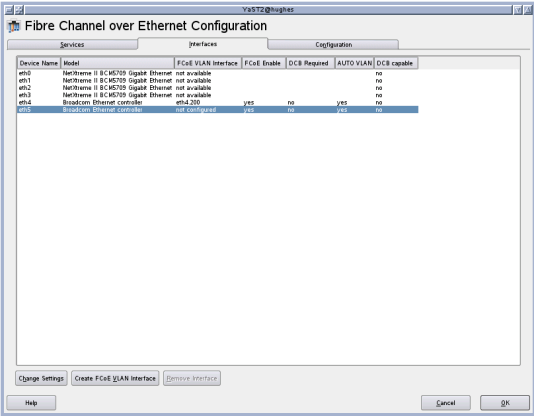
- **FCoE Service Start:** Specifies whether to start the Fibre Channel over Ethernet service `fcoemon` daemon at the server boot time or manually. The daemon controls the FCoE interfaces and establishes a connection with the `lldpad` daemon. The values are *When Booting* (default) or *Manually*.
- **Lldpad Service Start:** Specifies whether to start the Link Layer Discovery Protocol agent `lldpad` daemon at the server boot time or manually. The `lldpad` daemon informs the `fcoemon` daemon about the Data Center Bridging features and the configuration of the FCoE interfaces. The values are *When Booting* (default) or *Manually*.

If you modify a setting, click *OK* to save and apply the change.

- 4 On the *Interfaces* tab, view information about all of the detected network adapters on the server, including information about VLAN and FCoE configuration. You can also create an FCoE VLAN interface, change settings for an existing FCoE interface, or remove an FCoE interface.

View FCoE Information

The *FCoE Interfaces* table displays the following information about each adapter:



The screenshot shows a window titled 'Fibre Channel over Ethernet Configuration' with tabs for 'Services', 'Interfaces', and 'Configuration'. The 'Interfaces' tab is active, displaying a table with the following data:

Device Name	Model	FCoE VLAN Interface	FCoE Enable	DCB Required	AUTO VLAN	DCB capable
eth0	Realtek 10/100/1000 Ethernet					no
eth1	Realtek 10/100/1000 Ethernet					no
eth2	Realtek 10/100/1000 Ethernet					no
eth3	Realtek 10/100/1000 Ethernet					no
eth4	Broadcom Ethernet controller	eth4.200	yes	no	yes	no
eth5	Broadcom Ethernet controller	eth5.200	yes	no	yes	no

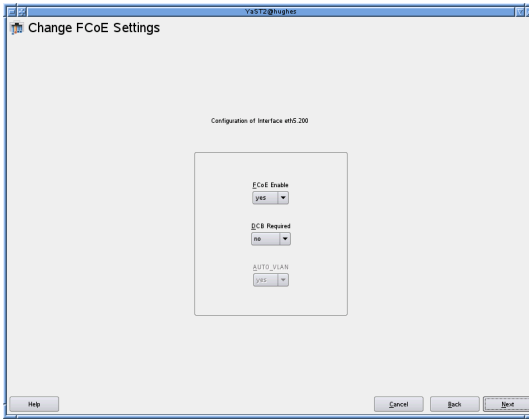
At the bottom of the window, there are buttons for 'Change Settings', 'Create FCoE VLAN Interface', 'Remove Interface', 'Help', 'Cancel', and 'OK'.

- **Device Name:** Specifies the adapter name such as `eth4`.
- **Model:** Specifies the adapter model information.
- **FCoE VLAN Interface**
 - **Interface Name:** If a name is assigned to the interface, such as `eth4.200`, FCoE is available on the switch, and the FCoE interface is activated for the adapter.
 - **Not Configured:** If the status is *not configured*, FCoE is enabled on the switch, but an FCoE interface has not been activated for the adapter. Select the adapter, then click *Create FCoE VLAN Interface* to activate the interface on the adapter.
 - **Not Available:** If the status is *not available*, FCoE is not possible for the adapter because FCoE has not been enabled for that connection on the switch.
- **FCoE Enable:** Specifies whether FCoE is enabled on the switch for the adapter connection. (*yes* or *no*)
- **DCB Required:** Specifies whether the adapter requires Data Center Bridging. (*yes* or *no*)
- **Auto VLAN:** Specifies whether automatic VLAN configuration is enabled for the adapter. (*yes* or *no*)

- **DCB Capable:** Specifies whether the adapter supports Data Center Bridging. (*yes* or *no*)

Change FCoE Settings

Select an FCoE VLAN interface, then click *Change Settings* at the bottom of the page to open the *Change FCoE Settings* dialog box.

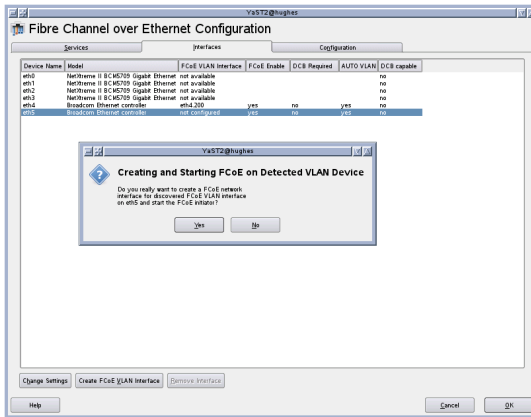


- **FCoE Enable:** Enable or disable the creation of FCoE instances for the adapter. Values are *yes* or *no*.
- **DCB Required:** Specifies whether Data Center Bridging is required for the adapter. Values are *yes* (default) or *no*. DCB is usually required.
- **Auto VLAN:** Specifies whether the `fcoemon` daemon creates the VLAN interfaces automatically. Values are *yes* or *no*.

If you modify a setting, click *Next* to save and apply the change. The settings are written to the `/etc/fcoe/ethX` file. The `fcoemon` daemon reads the configuration files for each FCoE interface when the daemon is initialized. There is a file for every FCoE interface.

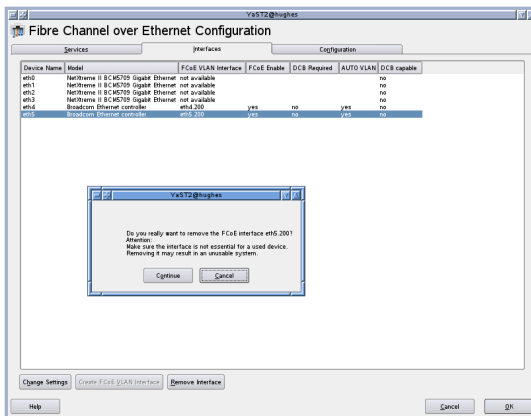
Create FCoE VLAN Interfaces

Select an adapter that has FCoE enabled but is not configured, then click *Yes* to configure the FCoE interface. The assigned interface name appears in the list, such as `eth5.200`.

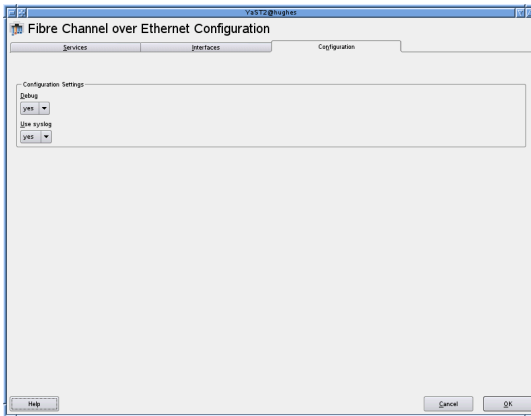


Remove FCoE Interface

Select the FCoE interface that you want to remove, click *Remove Interface* at the bottom of the page, then click *Continue* to confirm. The FCoE Interface value changes to *not configured*.



- 5 On the *Configuration* tab, view or modify the general settings for the FCoE system service.



- **Debug:** Enables or disables debugging messages from the FCoE service script and `fcoemon` daemon. The values are *Yes* or *No* (default).
- **Use syslog:** Specifies whether messages are sent to the system log (`/var/log/syslog`). The values are *Yes* (default) or *No*. (Data is logged in `/var/log/messages`.)

If you modify a setting, click *OK* to save and apply the change. The settings are written to the `/etc/fcoe/config` file.

- 6 Click *OK* to save and apply changes.

14.4 Configuring FCoE with Commands

- 1 Log in to the server as the `root` user, then open a terminal console.
- 2 Use YaST to configure the Ethernet network interface card, such as `eth2`.
- 3 Start the Link Layer Discovery Protocol agent daemon (`lldpad`).

```
lldpad start
```

4 Enable Data Center Bridging on your Ethernet adapter.

```
dcbtool sc eth2 dcb on
Version:      2
Command:      Set Config
Feature:      DCB State
Port:         eth2
Status:       Successful
```

5 Enable and set the Priority Flow Control (PFC) settings for Data Center Bridging.

```
dcbtool sc eth<x> pfc e:1 a:1 w:1
```

Argument setting values are:

e:<0|1>

Controls feature enable.

a:<0|1>

Controls whether the feature is advertised via Data Center Bridging Exchange protocol to the peer.

w:<0|1>

Controls whether the feature is willing to change its operational configuration based on what is received from the peer.

6 Enable the Data Center Bridging to accept the switch's priority setting for FCoE.

```
dcbtool sc eth2 app:fcoe e:1
Version:      2
Command:      Set Config
Feature:      Application FCoE
Port:         eth2
Status:       Successful
```

7 Copy the default FCoE configuration file to /etc/fcoe/cfg-eth2.

```
cp /etc/fcoe/cfg-ethx /etc/fcoe/cfg-eth2
```

8 Start the FCoE Initiator service.

```
rcfcoe start
Starting FCoE initiator service
```

- 9 Set up the Link Layer Discovery Protocol agent daemon (`lldpad`) and the FCoE Initiator service to start when booting.

```
chkconfig boot.lldpad on
chkconfig boot.fcoe on
```

14.5 Managing FCoE Instances with the FCoE Administration Tool

The `fcoeadm` utility is the Fibre Channel over Ethernet (FCoE) management tool for the Open-FCoE project. It can be used to create, destroy, and reset an FCoE instance on a given network interface. The `fcoeadm` utility sends commands to a running `fcoemon` process via a socket interface. For information about `fcoemon`, see the `fcoemon(8)` man page.

The `fcoeadm` utility allows you to query the FCoE instances about the following:

- Interfaces
- Target LUNs
- Port statistics

The `fcoeadm` utility is part of the `fcoe-utils` package. It is maintained by the Open-FCoE project [<http://open-fcoe.org/>].

Syntax

Fibre Channel over Ethernet Administration version 1.0.12.

```
fcoeadm
[-c|--create] [<ethX>]
[-d|--destroy] [<ethX>]
[-r|--reset] [<ethX>]
[-S|--Scan] [<ethX>]
```

```

[-i|--interface] [<ethX>]
[-t|--target] [<ethX>]
[-l|--lun] [<ethX>]
[-s|--stats <ethX>] [<interval>]
[-v|--version]
[-h|--help]

```

Options

-c , --create <ethX>

Creates an FCoE instance based on the specified network interface. If an `fcoemon` configuration file does not exist for the Open-FCoE service daemon interface (`/etc/fcoe/cfg-ethx`; see `fcoemon(8)` man page), the created FCoE instance does not require Data Center Bridging.

Example: To create an FCoE instance on `eth2.101`:

```
fcoeadm -c eth2.101
```

-d , --destroy <ethX>

Destroys an FCoE instance on the specified network interface. This does not destroy FCoE instances created by `fipvlan`.

Example: To destroy an FCoE instance on `eth2.101`:

```
fcoeadm -d eth2.101
```

-h , --help

Displays the usage message of the `fcoeadm` command.

-i , --interface [<ethX>]

Shows information about the FCoE instance on the specified network interface. If no network interface is specified, it shows information for all FCoE instances.

Examples

To show information about all of the adapters and their ports that have FCoE instances created:

```
fcoeadm -i
```

To show information about all of the FCoE instances on interface `eth3`:


```
fcoeadm -i eth3
```

```
-l , --lun [<ethX>]
```

Shows detailed information about the discovered SCSI LUNs associated with the FCoE instance on the specified network interface. If no network interface is specified, it shows information about SCSI LUNs from all FCoE instances.

Examples

To show detailed information about all of the LUNs discovered on all FCoE connections:

```
fcoeadm -l
```

To show detailed information about all of the LUNs discovered on a specific connections, such as `eth3.101`:

```
fcoeadm -l eth3.101
```

```
-r , --reset <ethX>
```

Resets the FCoE instance on the specified network interface. This does not reset FCoE instances created by `fipvlan`.

Example: To reset the FCoE instance on `eth2.101`:

```
fcoeadm -r eth2.101
```

```
-s , --stats <ethX> [interval]
```

Shows the statistics (including FC4 statistics) of the FCoE instance on the specified network interface. It displays one line per given time interval. Specify the interval value in whole integers greater than 0. The interval value is the elapsed time of the interval in seconds. If an interval is not specified, the default interval is 1 second.

Examples:

You can show statistics information about a specific `eth3` port that has FCoE instances. The statistics are displayed one line per time interval. The default interval of one second is not specified in the command.

```
fcoeadm -s eth3
```

To show statistics information about a specific `eth3` port that has FCoE instances, at an interval of 3 seconds. The statistics are displayed one line per time interval.

```
fcoeadm -s eth3 3
```

`-S , --Scan <ethX>`

Rescans for new targets and LUN for the specified network interface. This does not rescan any NPIV (N_Port ID Virtualization) instances created on the same port, and does not rescan any FCoE instances created by `fipvlan`.

`-t , --target [<ethX>]`

Shows information about the discovered targets associated with the FCoE instance on the specified network interface. If no network interface is specified, it shows information about discovered targets from all FCoE instances.

Examples

You can show information about all of the discovered targets from all of the ports that have FCoE instances. They might be on different adapter cards. After each discovered target, any associated LUNs are listed.

```
fcoeadm -t
```

You can show information about all of the discovered targets from a given `eth3` port having FCoE instance. After each discovered target, any associated LUNs are listed.

```
fcoeadm -t eth3
```

`-v , --version`

Displays the version of the `fcoeadm` command.

FCoE Response Examples

View FCoE Initiator Status for FC-ID Node/Port Number

```
fcoeadm -i eth0.201
```

```

Description:      82599EB 10-Gigabit SFI/SFP+ Network Connection
Revision:        01
Manufacturer:    Intel Corporation
Serial Number:   001B219B258C
Driver:          ixgbe 3.3.8-k2
Number of Ports: 1

```

```

Symbolic Name:   fcoe v0.1 over eth0.201
OS Device Name:  host8
Node Name:       0x1000001B219B258E
Port Name:       0x2000001B219B258E
FabricName:      0x2001000573D38141
Speed:           10 Gbit
Supported Speed: 10 Gbit
MaxFrameSize:    2112
FC-ID (Port ID): 0x790003
State:           Online

```

View FCoE Targets for FC-ID Node/Port Number

```

fcoeadm -t eth0.201
Interface:      eth0.201
Roles:          FCP Target
Node Name:      0x200000D0231B5C72
Port Name:      0x210000D0231B5C72
Target ID:      0
MaxFrameSize:   2048
OS Device Name: rport-8:0-7
FC-ID (Port ID): 0x79000C
State:          Online

```

LUN ID	Device Name	Capacity	Block Size	Description
-----	-----	-----	-----	
40	/dev/sdqi	792.84 GB	512	IFT DS S24F-R2840-4 (rev
386C)				
72	/dev/sdpk	650.00 GB	512	IFT DS S24F-R2840-4 (rev
386C)				
168	/dev/sdgy	1.30 TB	512	IFT DS S24F-R2840-4 (rev
386C)				

14.6 Setting Up Partitions for an FCoE Initiator Disk

You can use the `fdisk (8)` command to set up partitions for an FCoE initiator disk.

```

fdisk /dev/sdc
  Device contains neither a valid DOS partition table, nor Sun, SGI or OSF
  disklabel.
  Building a new DOS disklabel with disk identifier 0xfc691889.
  Changes will remain in memory only, until you decide to write them.
  After that, of course, the previous content won't be recoverable.

  Warning: Invalid flag 0x0000 of partition table 4 will be corrected by
  w(rite)

  Command (n for help): n
  Command action
    e    extended
    p    primary partition (1-4)

  p
    Partition number (1-4): 4
    First cylinder (1-1017, default 1):
      Using default value 1
    Last cylinder, *cylinders or *size(K,M,G) (1-1017, default 1017):
      Using default value 1017

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

    Calling ioctl() to re-read partition table.
    Syncing disks.

```

14.7 Creating a File System on an FCoE Initiator Disk

You can use the `mkfs (8)` command to create a file system on an FCoE initiator disk.

```

mkfs /dev/sdc
  mke2fs 1.41.9 (22-Aug-2011)
  /dev/sdc is entire device, not just one partition!
  Proceed anyway? (y, n) y
  Filesystem label=
  OS type: Linux
  Block size=4096 (log-2)
  262144 inodes, 1048576 blocks
  52428 blocks (5.00%) reserved for the super user
  First data block=0
  Maximum filesystem blocks=1073741824
  32 block groups
  32768 blocks per group, 32768 fragments per group

```

```
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 804736

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 27 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

14.8 Additional Information

For information, see the follow documentation:

- For information about the Open-FCoE service daemon, see the `fcoemon(8)` man page.
- For information about the Open-FCoE Administration tool, see the `fcoeadm(8)` man page.
- For information about the Data Center Bridging Configuration tool, see the `dcbtool(8)` man page.
- For information about the Link Layer Discovery Protocol agent daemon, see the `lldpad(8)` man page.
- *Open Fibre Channel over Ethernet Quick Start* [<http://www.open-fcoe.org/open-fcoe/wiki/quickstart>].

Volume Snapshots

A file system snapshot is a copy-on-write technology that monitors changes to an existing volume's data blocks so that when a write is made to one of the blocks, the block's value at the snapshot time is copied to a snapshot volume. In this way, a point-in-time copy of the data is preserved until the snapshot volume is deleted.

- Section 15.1, “Understanding Volume Snapshots” (page 253)
- Section 15.2, “Creating Linux Snapshots with LVM” (page 255)
- Section 15.3, “Monitoring a Snapshot” (page 255)
- Section 15.4, “Deleting Linux Snapshots” (page 256)

15.1 Understanding Volume Snapshots

A file system snapshot contains metadata about and data blocks from an original volume that have changed since the snapshot was taken. When you access data via the snapshot, you see a point-in-time copy the original volume. There is no need to restore data from backup media or to overwrite the changed data.

In a Xen host environment, the virtual machine must be using an LVM logical volume as its storage back-end, as opposed to using a virtual disk file.

Linux snapshots allow you to create a backup from a point-in-time view of the file system. The snapshot is created instantly and persists until you delete it. You can backup the file system from the snapshot while the volume itself continues to be available for users. The snapshot initially contains some metadata about the snapshot, but no actual data from the original volume. Snapshot uses copy-on-write technology to detect when data changes in an original data block. It copies the value it held when the snapshot was taken to a block in the snapshot volume, then allows the new data to be stored in the original block. As blocks change from their original value, the snapshot size grows.

When you are sizing the snapshot, consider how much data is expected to change on the original volume and how long you plan to keep the snapshot. The amount of space that you allocate for a snapshot volume can vary, depending on the size of the original volume, how long you plan to keep the snapshot, and the number of data blocks that are expected to change during the snapshot's lifetime. The snapshot volume cannot be resized after it is created. As a guide, create a snapshot volume that is about 10% of the size of the original logical volume. If you anticipate that every block in the original volume will change at least one time before you delete the snapshot, then the snapshot volume should be at least as large as the original volume plus some additional space for metadata about the snapshot volume. Less space is required if the data changes infrequently or if the expected lifetime is sufficiently brief.

IMPORTANT

During the snapshot's lifetime, the snapshot must be mounted before its original volume can be mounted.

When you are done with the snapshot, it is important to remove it from the system. A snapshot eventually fills up completely as data blocks change on the original volume. When the snapshot is full, it is disabled, which prevents you from remounting the original volume.

Remove snapshots in a last created, first deleted order.

15.2 Creating Linux Snapshots with LVM

The Logical Volume Manager (LVM) can be used for creating snapshots of your file system.

- Open a terminal console, log in as the `root` user, then enter

```
lvcreate -s -L 1G -n snap_volume source_volume_path
```

For example:

```
lvcreate -s -L 1G -n linux01-snap /dev/lvm/linux01
```

The snapshot is created as the `/dev/lvm/linux01-snap` volume.

15.3 Monitoring a Snapshot

- Open a terminal console, log in as the `root` user, then enter

```
lvdisplay snap_volume
```

For example:

```
lvdisplay /dev/vg01/linux01-snap
```

```
--- Logical volume ---
LV Name                /dev/lvm/linux01
VG Name                vg01
LV UUID                QHVJYh-PR3s-A4SG-s4Aa-MyWN-Ra7a-HL47KL
LV Write Access        read/write
LV snapshot status     active destination for /dev/lvm/linux01
LV Status              available
# open                 0
LV Size                80.00 GB
Current LE             1024
COW-table size         8.00 GB
COW-table LE           512
Allocated to snapshot  30%
Snapshot chunk size    8.00 KB
Segments               1
```

Allocation	inherit
Read ahead sectors	0
Block device	254:5

15.4 Deleting Linux Snapshots

- Open a terminal console, log in as the `root` user, then enter

```
lvremove snap_volume_path
```

For example:

```
lvremove /dev/lvm/linux01-snap
```

Managing Access Control Lists over NFSv4

16

There is no single standard for Access Control Lists (ACLs) in Linux beyond the simple user-group-others read, write, and execute (`rwX`) flags. One option for finer control are the *Draft POSIX ACLs*, which were never formally standardised by POSIX. Another is the NFSv4 ACLs, which were designed to be part of the NFSv4 network filesystem with the goal of making something that provided reasonable compatibility between POSIX systems on Linux and WIN32 systems on Microsoft Windows.

NFSv4 ACLs are not sufficient to correctly implement Draft POSIX ACLs so no attempt has been made to map ACL accesses on an NFSv4 client (such as using `setfacl`).

When using NFSv4, Draft POSIX ACLs cannot be used even in emulation and NFSv4 ACLs need to be used directly; i.e., while `setfacl` can work on NFSv3, it cannot work on NFSv4. To allow NFSv4 ACLs to be used on an NFSv4 filesystem, SUSE Linux Enterprise Server provides the `nfs4-acl-tools` package which contains the following:

- `nfs4-getfacl`
- `nfs4-setfacl`
- `nfs4-editacl`

These operate in a generally similar way to `getfacl` and `setfacl` for examining and modifying NFSv4 ACLs. These commands are effective only if the file system on the NFS server provides full support for NFSv4 ACLs. Any limitation imposed by the server is felt by these programs running on the client in that some particular combinations of Access Control Entries (ACEs) might not be possible.

Additional Information

For information, see “ACLs” on the Linux-nfs.org Web site [http://wiki.linux-nfs.org/wiki/index.php/ACLs#Introduction_to_NFSv4_ACLs].

Troubleshooting Storage Issues

17

This section describes how to work around known issues for devices, software RAIDs, multipath I/O, and volumes.

- Section 17.1, “Is DM-MPIO Available for the Boot Partition?” (page 259)
- Section 17.2, “Issues for iSCSI” (page 259)
- Section 17.3, “Issues for Multipath I/O” (page 260)
- Section 17.4, “Issues for Software RAIDs” (page 260)

17.1 Is DM-MPIO Available for the Boot Partition?

Device Mapper Multipath I/O (DM-MPIO) is supported for the boot partition, beginning in SUSE Linux Enterprise Server 10 Support Pack 1.

17.2 Issues for iSCSI

See Chapter 17, *Troubleshooting Storage Issues* (page 259).

17.3 Issues for Multipath I/O

See Section 7.14, “Troubleshooting MPIO” (page 142).

17.4 Issues for Software RAIDs

See Section 8.3, “Troubleshooting Software RAIDs” (page 150).



GNU Licenses

This appendix contains the GNU General Public License Version 2 and the GNU Free Documentation License Version 1.2.

- Section A.1, “GNU General Public License” (page 261)
- Section A.2, “GNU Free Documentation License” (page 264)

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to ensure that the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to ensure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must ensure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a)** You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b)** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c)** If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a)** Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b)** Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c)** Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and an idea of what it does.

Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'. This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright
interest in the program `Gnomovision'
(which makes passes at compilers) written
by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D.** Preserve all the copyright notices of the Document.
- E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H.** Include an unaltered copy of this License.
- I.** Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K.** For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M.** Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N.** Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O.** Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has

been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

B

Documentation Updates

This section contains information about documentation content changes made to the *SUSE Linux Enterprise Server Storage Administration Guide* since the initial release of SUSE Linux Enterprise Server 11.

This document was updated on the following dates:

- Section B.1, “January 18, 2012 (SLES 11 SP2)” (page 270)
- Section B.2, “July 12, 2011” (page 274)
- Section B.3, “June 14, 2011” (page 274)
- Section B.4, “May 5, 2011” (page 275)
- Section B.5, “January 2011” (page 275)
- Section B.6, “September 16, 2010” (page 276)
- Section B.7, “June 21, 2010” (page 277)
- Section B.8, “May 2010 (SLES 11 SP1)” (page 279)
- Section B.9, “February 23, 2010” (page 281)
- Section B.10, “December 1, 2009” (page 282)
- Section B.11, “October 20, 2009” (page 284)

- Section B.12, “August 3, 2009” (page 285)
- Section B.13, “June 22, 2009” (page 286)
- Section B.14, “May 21, 2009” (page 288)

B.1 January 18, 2012 (SLES 11 SP2)

Updates were made to the following sections. The changes are explained below.

- Section B.1.1, “Fibre Channel Storage over Ethernet Networks: FCoE” (page 270)
- Section B.1.2, “GNU Licenses” (page 270)
- Section B.1.3, “LVM Configuration ” (page 271)
- Section B.1.4, “Managing Access Control Lists over NFSv4” (page 271)
- Section B.1.5, “Managing Multipath I/O for Devices” (page 271)
- Section B.1.6, “Mass Storage over IP Networks: iSCSI ” (page 272)
- Section B.1.7, “Overview of File Systems on Linux” (page 272)
- Section B.1.8, “Resizing File Systems” (page 273)
- Section B.1.9, “Resizing Software RAID Arrays with mdadm” (page 274)

B.1.1 Fibre Channel Storage over Ethernet Networks: FCoE

This section is new. Open Fibre Channel over Ethernet (OpenFCoE) is supported beginning in SLES 11.

B.1.2 GNU Licenses

This section is new.

B.1.3 LVM Configuration

Location	Change
Section 4.6, “Tagging LVM2 Storage Objects” (page 46)	This section is new.

B.1.4 Managing Access Control Lists over NFSv4

This section is new.

B.1.5 Managing Multipath I/O for Devices

Location	Change
Section “Configuring Default Multipath Behavior in <code>/etc/multipath.conf</code> ” (page 110)	The default <code>getuid</code> path for SLES 11 is <code>/lib/udev/scsi_id</code> .
Section 7.12, “Managing I/O in Error Situations” (page 140)	In the <code>dmsetup message</code> commands, the 0 value represents the sector and is used when sector information is not needed.
Section 7.13, “Resolving Stalled I/O” (page 142)	
Table 7.6, “Multipath Attributes” (page 114)	Recommendations were added for the <code>no_path_retry</code> and <code>failback</code> settings when multipath I/O is used in a cluster environment.
Table 7.6, “Multipath Attributes” (page 114)	The path-selector option names and settings were corrected:

Location	Change
	round-robin 0 least-pending 0 service-time 0 queue-length 0

B.1.6 Mass Storage over IP Networks: iSCSI

Location	Change
Section 13.4, “Using iSCSI Disks when Installing” (page 229)	This section is new.
Section 13.5.4, “iSCSI Targets Are Mounted When the Configuration File Is Set to Manual” (page 231)	This section is new.

B.1.7 Overview of File Systems on Linux

Location	Change
Section 1.2.1, “Btrfs” (page 3)	This section is new. Btrfs is fully supported in SLES 11 SP2 and later.
Section 1.2.4, “ReiserFS” (page 8)	<p>IMPORTANT</p> <p>The ReiserFS file system is fully supported for the lifetime of SUSE Linux Enterprise Server 11 specifically for migration purposes. SUSE plans to remove support for creating</p>

Location	Change
	new ReiserFS file systems starting with SUSE Linux Enterprise Server 12.
Section 1.2.6, “Feature Comparison” (page 10)	This section is new.
Section 1.4, “Large File Support in Linux” (page 12)	The values in this section were updated to current standards.
Section 1.5, “Managing Devices with the YaST2 Partitioner” (page 13)	This section is new.

B.1.8 Resizing File Systems

Location	Change
Section 5.1, “Guidelines for Resizing” (page 61)	The <code>resize2fs</code> command allows only the Ext3 file system to be resized if mounted. The size of an Ext3 volume can be increased or decreased when the volume is mounted or unmounted. The Ext2/4 file systems must be unmounted for increasing or decreasing the volume size.
Section 5.4, “Decreasing the Size of an Ext2 or Ext3 File System” (page 66)	

B.1.9 Resizing Software RAID Arrays with mdadm

Location	Change
Section 11.2.2, “Increasing the Size of the RAID Array” (page 182)	The <code>--assume-clean</code> option is new.

B.2 July 12, 2011

Updates were made to the following section. The changes are explained below.

- Section B.2.1, “Managing Multipath I/O for Devices” (page 274)

B.2.1 Managing Multipath I/O for Devices

Location	Change
Section 7.2.3, “Using LVM2 on Multipath Devices” (page 82)	Running <code>mkinitrd</code> is needed only if the root (/) device or any parts of it (such as <code>/var</code> , <code>/etc</code> , <code>/log</code>) are on the SAN and multipath is needed to boot.
Section 7.8, “Configuring Multipath I/O for an Existing Software RAID” (page 131)	

B.3 June 14, 2011

Updates were made to the following section. The changes are explained below.

- Section B.3.1, “Managing Multipath I/O for Devices” (page 275)

- Section B.3.2, “What’s New for Storage in SLES 11” (page 275)

B.3.1 Managing Multipath I/O for Devices

Location	Change
??? (page 122) in Table 7.6, “Multipath Attributes” (page 114)	The default setting changed from multibus to failover in SLES 11.

B.3.2 What’s New for Storage in SLES 11

Location	Change
Section 2.3.12, “Change from Multibus to Failover as the Default Setting for the MPIO Path Grouping Policy” (page 27)	This section is new.

B.4 May 5, 2011

This release fixes broken links and removes obsolete references.

B.5 January 2011

Updates were made to the following section. The changes are explained below.

- Section B.5.1, “LVM Configuration” (page 276)
- Section B.5.2, “Managing Multipath I/O for Devices” (page 276)
- Section B.5.3, “Resizing File Systems” (page 276)

B.5.1 LVM Configuration

Location	Change
Section 4.3, “Creating Volume Groups” (page 39)	LVM2 does not restrict the number of physical extents. Having a large number of extents has no impact on I/O performance to the logical volume, but it slows down the LVM tools.

B.5.2 Managing Multipath I/O for Devices

Location	Change
Tuning the Failover for Specific Host Bus Adapters	This section was removed. For HBA failover guidance, refer to your vendor documentation.

B.5.3 Resizing File Systems

Location	Change
Section 11.3.1, “Decreasing the Size of the File System” (page 186)	Decreasing the size of the file system is supported when the file system is unmounted.

B.6 September 16, 2010

Updates were made to the following sections. The changes are explained below.

- Section B.6.1, “LVM Configuration” (page 277)

B.6.1 LVM Configuration

Location	Change
Section 4.2, “Creating LVM Partitions” (page 37)	<p>The discussion and procedure were expanded to explain how to configure a partition that uses the entire disk.</p> <p>The procedure was modified to use the Hard Disk partitioning feature in the YaST Partitioner.</p>
All LVM Management sections	Procedures throughout the chapter were modified to use Volume Management in the YaST Partitioner.
Section 4.7, “Resizing a Volume Group” (page 55)	This section is new.
Section 4.8, “Resizing a Logical Volume with YaST” (page 57)	This section is new.
Section 4.10, “Deleting a Volume Group” (page 59)	This section is new.
Section 4.11, “Deleting an LVM Partition (Physical Volume)” (page 60)	This section is new.

B.7 June 21, 2010

Updates were made to the following sections. The changes are explained below.

- Section B.7.1, “LVM Configuration” (page 278)
- Section B.7.2, “Managing Multipath I/O” (page 278)
- Section B.7.3, “Managing Software RAID6 and 10 with mdadm” (page 278)

- Section B.7.4, “Mass Storage on IP NetWork: iSCSI” (page 279)

B.7.1 LVM Configuration

Location	Change
Section 4.2, “Creating LVM Partitions” (page 37)	Details were added to the procedure.

B.7.2 Managing Multipath I/O

Location	Change
Section “Configuring User-Friendly Names or Alias Names in /etc/multipath.conf” (page 104)	Using user-friendly names for the root device can result in data loss. Added alternatives from <i>TID 7001133: Recommendations for the usage of user_friendly_names in multipath configurations</i> [http://www.novell.com/support/search.do?cmd=displayKC&docType=k&externalId=7001133].

B.7.3 Managing Software RAIDs 6 and 10 with mdadm

Location	Change
Section “Far Layout” (page 171)	Errata in the example were corrected.

B.7.4 Mass Storage on IP NetWork: iSCSI

Location	Change
Section 13.5.1, “Hotplug Doesn’t Work for Mounting iSCSI Targets” (page 230)	This section is new.

B.8 May 2010 (SLES 11 SP1)

Updates were made to the following sections. The changes are explained below.

- Section B.8.1, “Managing Multipath I/O for Devices” (page 279)
- Section B.8.2, “Mass Storage over IP Networks: iSCSI” (page 280)
- Section B.8.3, “Software RAID Configuration” (page 281)
- Section B.8.4, “What’s New” (page 281)

B.8.1 Managing Multipath I/O for Devices

Location	Change
Section 7.2.3, “Using LVM2 on Multipath Devices” (page 82)	The example in Step 3 (page 82) was corrected.
Section 7.2.6, “SAN Timeout Settings When the Root Device Is Multipathed” (page 84)	This section is new.
Section 7.3.2, “Multipath I/O Management Tools” (page 92)	The file list for a package can vary for different server architectures. For a list of files included in the multipath-tools package, go to the <i>SUSE Linux Enterprise Server Technical SpecificationsPack-</i>

Location	Change
	<i>age Descriptions</i> Web page [http://www.novell.com/products/server/techspecs.html?tab=1], find your architecture and select <i>Packages Sorted by Name</i> , then search on “multipath-tools” to find the package list for that architecture.
Section 7.4.1, “Preparing SAN Devices for Multipathing” (page 98)	If the SAN device will be used as the root device on the server, modify the timeout settings for the device as described in Section 7.2.6, “SAN Timeout Settings When the Root Device Is Multipathed” (page 84).
Section “Verifying the Setup in the etc/multipath.conf File” (page 102)	Added example output for -v3 verbosity.
Section “Enabling Multipath I/O at Install Time on an Active/Active Multipath Storage LUN” (page 127)	This section is new.
Section “Enabling Multipath I/O at Install Time on an Active/Passive Multipath Storage LUN” (page 127)	This section is new.

B.8.2 Mass Storage over IP Networks: iSCSI

Location	Change
Step 7g (page 218) in Section 13.2.2, “Creating iSCSI Targets with YaST” (page 213)	In the <i>YaSTNetwork ServicesiSCSI Target</i> function, the <i>Save</i> option allows you to export the iSCSI target information, which makes it easier

Location	Change
	to provide this information to consumers of the resources.
Section 13.5, “Troubleshooting iSCSI” (page 229)	This section is new.

B.8.3 Software RAID Configuration

Location	Change
Section 8.4, “For More Information” (page 151)	The Software RAID HOW-TO has been deprecated. Use the <i>Linux RAID</i> wiki [https://raid.wiki.kernel.org/index.php/Linux_Raid] instead.

B.8.4 What’s New

Location	Change
Section 2.2, “What’s New in SLES 11 SP1” (page 18)	This section is new.

B.9 February 23, 2010

Updates were made to the following sections. The changes are explained below.

- Section B.9.1, “Configuring Software RAID for the Root Partition” (page 282)
- Section B.9.2, “Managing Multipath I/O” (page 282)

B.9.1 Configuring Software RAID for the Root Partition

Location	Change
Section 9.1, “Prerequisites for the Software RAID” (page 153)	Corrected an error in the RAID 0 definition..

B.9.2 Managing Multipath I/O

Location	Change
Section 7.9, “Scanning for New Devices without Rebooting” (page 134)	Added information about using the <code>rescan -scsi-bus.sh</code> script to scan for devices without rebooting.
Section 7.10, “Scanning for New Partitioned Devices without Rebooting” (page 137)	Added information about using the <code>rescan -scsi-bus.sh</code> script to scan for devices without rebooting.

B.10 December 1, 2009

Updates were made to the following sections. The changes are explained below.

- Section B.10.1, “Managing Multipath I/O for Devices” (page 283)
- Section B.10.2, “Resizing File Systems” (page 283)
- Section B.10.3, “What’s New” (page 284)

B.10.1 Managing Multipath I/O for Devices

Location	Change
Section 7.2.3, “Using LVM2 on Multipath Devices” (page 82)	The -f mpath option changed to -f multipath: mkinitrd -f multipath
Section 7.8, “Configuring Multipath I/O for an Existing Software RAID” (page 131)	
prio_callout in Table 7.6, “Multipath Attributes” (page 114)	Multipath prio_callouts are located in shared libraries in <code>/lib/libmultipath/lib*</code> . By using shared libraries, the callouts are loaded into memory on daemon startup.

B.10.2 Resizing File Systems

Location	Change
Section 5.1.1, “File Systems that Support Resizing” (page 62)	The <code>resize2fs</code> utility supports online or offline resizing for the <code>ext3</code> file system.

B.10.3 What's New

Location	Change
Section 2.3.10, “Location Change for Multipath Tool Callouts” (page 26)	This section is new.
Section 2.3.11, “Change from mpath to multipath for the mkinitrd -f Option” (page 26)	This section is new.

B.11 October 20, 2009

Updates were made to the following sections. The changes are explained below.

- Section B.11.1, “LVM Configuration” (page 284)
- Section B.11.2, “Managing Multipath I/O for Devices” (page 285)
- Section B.11.3, “What’s New” (page 285)

B.11.1 LVM Configuration

Location	Change
Section 4.1, “Understanding the Logical Volume Manager” (page 34)	In the YaST Control Center, select <i>System > Partitioner</i> .

B.11.2 Managing Multipath I/O for Devices

Location	Change
Section “Blacklisting Non-Multipathed Devices in /etc/multipath.conf” (page 109)	The keyword <code>devnode_blacklist</code> has been deprecated and replaced with the keyword <code>blacklist</code> .
Section “Configuring Default Multipath Behavior in /etc/multipath.conf” (page 110)	Changed <code>getuid_callout</code> to <code>getuid</code> .
Section “Understanding Priority Groups and Attributes” (page 114)	Changed <code>getuid_callout</code> to <code>getuid</code> .
??? (page 122)	Added descriptions of least-pending, length-load-balancing, and service-time options.

B.11.3 What’s New

Location	Change
Section 2.3.9, “Advanced I/O Load-Balancing Options for Multipath” (page 26)	This section is new.

B.12 August 3, 2009

Updates were made to the following section. The change is explained below.

- Section B.12.1, “Managing Multipath I/O” (page 286)

B.12.1 Managing Multipath I/O

Location	Change
Section 7.2.5, “Using --noflush with Multipath Devices” (page 83)	This section is new.

B.13 June 22, 2009

Updates were made to the following sections. The changes are explained below.

- Section B.13.1, “Managing Multipath I/O” (page 286)
- Section B.13.2, “Managing Software RAIDs 6 and 10 with mdadm” (page 287)
- Section B.13.3, “Mass Storage over IP Networks: iSCSI ” (page 287)

B.13.1 Managing Multipath I/O

Location	Change
Section 7.7, “Configuring Multipath I/O for the Root Device” (page 126)	Added Step 4 (page 130) and Step 6 (page 130) for System Z.
Section 7.10, “Scanning for New Partitioned Devices without Re-booting” (page 137)	Corrected the syntax for the command lines in Step 2.

Location	Change
Section 7.10, “Scanning for New Partitioned Devices without Re-booting” (page 137)	Step 7 (page 138) replaces old Step 7 and Step 8.

B.13.2 Managing Software RAIDs 6 and 10 with mdadm

Location	Change
Section 10.4, “Creating a Degraded RAID Array” (page 173)	To see the rebuild progress while being refreshed every second, enter <pre>watch -n 1 cat /proc/mdstat</pre>

B.13.3 Mass Storage over IP Networks: iSCSI

Location	Change
Section 13.3.1, “Using YaST for the iSCSI Initiator Configuration” (page 222)	<p>Re-organized material for clarity.</p> <p>Added information about how to use the settings for the Start-up option for iSCSI target devices:</p> <ul style="list-style-type: none"> • Automatic: This option is used for iSCSI targets that are to be connected when the iSCSI service itself starts up. This is the typical configuration. • Onboot: This option is used for iSCSI targets that are to be connected during boot; that is, when root (/) is on iSCSI. As such, the iSCSI

Location	Change
	target device will be evaluated from the initrd on server boots.

B.14 May 21, 2009

Updates were made to the following section. The changes are explained below.

- Section B.14.1, “Managing Multipath I/O” (page 288)

B.14.1 Managing Multipath I/O

Location	Change
Section “Storage Arrays That Are Automatically Detected for Multipathing” (page 86)	Testing of the IBM zSeries device with multipathing has shown that the <code>dev_loss_tmo</code> parameter should be set to 90 seconds, and the <code>fast_io_fail_tmo</code> parameter should be set to 5 seconds. If you are using zSeries devices, you must manually create and configure the <code>/etc/multipath.conf</code> file to specify the values. For information, see Section “Configuring Default Settings for zSeries in <code>/etc/multipath.conf</code> ” (page 110).
Section 7.3.1, “Device Mapper Multipath Module” (page 90)	Multipathing is supported for the <code>/boot</code> device in SUSE Linux Enterprise Server 11 and later.
Section “Configuring Default Settings for zSeries in <code>/etc/multipath.conf</code> ” (page 110)	This section is new.

Location	Change
Section 7.7, “Configuring Multipath I/O for the Root Device” (page 126)	DM-MP is now available and supported for <code>/boot</code> and <code>/root</code> in SUSE Linux Enterprise Server 11.

