



Storage Administration Guide

SUSE Linux Enterprise Server 12

Storage Administration Guide

SUSE Linux Enterprise Server 12

Provides information about how to manage storage devices on a SUSE Linux Enterprise Server.

Publication date: Sep 30 2014

SUSE Linux Products GmbH
Maxfeldstr. 5
90409 Nürnberg
GERMANY

<https://www.suse.com/documentation> 

© 2006–2017 SUSE LLC

GNU ##### (GNU Free Documentation License) 1.2 ## 1.3 # (####)#####/#####
#####“GNU #####”##### 1.2 #####

SUSE # Novell ##### Novell #####<http://www.novell.com/company/legal/trade-marks/tmlist.html>  ##### (®™ #) ## SUSE # Novell ##### (*) #####
#####

#####SUSE LLC #####
#####

Contents

About This Guide xv

1 Overview of File Systems in Linux 1

- 1.1 Terminology 1
- 1.2 Major File Systems in Linux 2
 - Btrfs 3 • Ext2 7 • Ext3 8 • ReiserFS 14 • XFS 16 • File System Feature Comparison 17
- 1.3 Other Supported File Systems 17
- 1.4 Large File Support in Linux 18
- 1.5 Linux Kernel Storage Limitations 20
- 1.6 Managing Devices with the YaST Partitioner 21
- 1.7 Additional Information 21

2 What's New for Storage in SLES 11 22

- 2.1 What's New in SLES 11 SP3 22
 - Btrfs Quotas 22 • iSCSI LIO Target Server 22 • Linux Software RAID5 23 • LVM2 23 • Multipath I/O 24
- 2.2 What's New in SLES 11 SP2 27
- 2.3 What's New in SLES 11 SP1 28
 - Saving iSCSI Target Information 29 • Modifying Authentication Parameters in the iSCSI Initiator 29 • Allowing Persistent Reservations for MPIO Devices 29 • MDADM 3.0.2 29 • Boot Loader Support for MDRAID External Metadata 30 • YaST Install and Boot Support for MDRAID External Metadata 30 • Improved Shutdown for MDRAID Arrays that Contain the Root File System 30 • MD over iSCSI Devices 31 • MD-SGPIIO 31 • Resizing LVM 2 Mirrors 31 • Updating Storage Drivers for Adapters on IBM Servers 31

- 2.4 What's New in SLES 11 32
- EVMS2 Is Deprecated 33
 - Ext3 as the Default File System 34
 - Default Inode Size Increased for Ext3 34
 - JFS File System Is Deprecated 34
 - OCFS2 File System Is in the High Availability Release 34
 - /dev/disk/by-name Is Deprecated 34
 - Device Name Persistence in the /dev/disk/by-id Directory 34
 - Filters for Multipathed Devices 35
 - User-Friendly Names for Multipathed Devices 35
 - Advanced I/O Load-Balancing Options for Multipath 36
 - Location Change for Multipath Tool Callouts 36
 - Change from mpmath to multipath for the mkinitrd -f Option 36
 - Change from Multibus to Failover as the Default Setting for the MPIO Path Grouping Policy 36

3 Planning a Storage Solution 38

- 3.1 Partitioning Devices 38
- 3.2 Multipath Support 38
- 3.3 Software RAID Support 38
- 3.4 File System Snapshots 39
- 3.5 Backup and Antivirus Support 39
- Open Source Backup 39
 - Commercial Backup and Antivirus Support 39

4 LVM Configuration 40

- 4.1 Understanding the Logical Volume Manager 41
- 4.2 Creating LVM Partitions 43
- 4.3 Creating Volume Groups 45
- 4.4 Configuring Physical Volumes 46
- 4.5 Configuring Logical Volumes 48
- 4.6 Automatically Activating Non-Root LVM Volume Groups 52
- 4.7 Tagging LVM2 Storage Objects 53
- Using LVM2 Tags 53
 - Requirements for Creating LVM2 Tags 54
 - Command Line Tag Syntax 54
 - Configuration File Syntax 55
 - Using Tags for a

Simple Activation Control in a Cluster 57 • Using Tags to Activate On Preferred Hosts in a Cluster 58

- 4.8 Resizing a Volume Group 61
- 4.9 Resizing a Logical Volume with YaST 62
- 4.10 Resizing a Logical Volume with Commands 63
- 4.11 Deleting a Volume Group 65
- 4.12 Deleting an LVM Partition (Physical Volume) 65
- 4.13 Using LVM Commands 66

5 Resizing File Systems 70

- 5.1 Guidelines for Resizing 70
 - File Systems that Support Resizing 71 • Increasing the Size of a File System 71 • Decreasing the Size of a File System 72
- 5.2 Increasing the Size of an Ext2, Ext3, or Ext4 File System 72
- 5.3 Increasing the Size of a Reiser File System 73
- 5.4 Decreasing the Size of an Ext2 or Ext3 File System 75
- 5.5 Decreasing the Size of a Reiser File System 75

6 Using UUIDs to Mount Devices 77

- 6.1 Naming Devices with udev 77
- 6.2 Understanding UUIDs 78
 - Using UUIDs to Assemble or Activate File System Devices 78 • Finding the UUID for a File System Device 78
- 6.3 Using UUIDs in the Boot Loader and /etc/fstab File (x86) 79
- 6.4 Using UUIDs in the Boot Loader and /etc/fstab File (IA64) 81
- 6.5 Additional Information 82

7 Managing Multipath I/O for Devices 83

7.1 Understanding Multipath I/O 84

7.2 Planning for Multipathing 84

Guidelines for Multipathing 85 • PRIO Settings in multipath-tools-0.4.9 88 • Using WWID, User-Friendly, and Alias Names for Multipath Devices 89 • Using LVM2 on Multipath Devices 89 • Using mdadm with Multipath Devices 93 • Using Multipath with NetApp Devices 93 • Using --noflush with Multipath Devices 94 • SAN Timeout Settings When the Root Device Is Multipath 94 • Partitioning Multipath Devices 95 • Supported Architectures for Multipath I/O 96 • Supported Storage Arrays for Multipathing 96

7.3 Multipath Management Tools 101

Device Mapper Multipath Module 101 • Multipath I/O Management Tools 104 • Using MDADM for Multipath Devices 105 • Linux multipath(8) Command 106 • Linux mpathpersist(8) Utility 111

7.4 Configuring the System for Multipathing 115

Preparing SAN Devices for Multipathing 115 • Partitioning Multipath Devices 116 • Configuring the Device Drivers in initrd for Multipathing 117 • Adding multipathd to the Boot Sequence 118

7.5 Enabling and Starting Multipath I/O Services 119

7.6 Creating or Modifying the /etc/multipath.conf File 119

Creating the /etc/multipath.conf File 120 • Sections in the /etc/multipath.conf File 121 • Verifying the Multipath Setup in the /etc/multipath.conf File 122 • Applying the /etc/multipath.conf File Changes to Update the Multipath Maps 125

7.7 Configuring Default Policies for Polling, Queueing, and Failback 125

7.8 Blacklisting Non-Multipath Devices 127

7.9 Configuring User-Friendly Names or Alias Names 130

7.10 Configuring Default Settings for zSeries Devices 134

- 7.11 Configuring Path Failover Policies and Priorities **135**
 - Configuring the Path Failover Policies **135** • Configuring Failover Priorities **136** • Using a Script to Set Path Priorities **148** • Configuring ALUA (mpath_prio_alua) **149** • Reporting Target Path Groups **152**
- 7.12 Configuring Multipath I/O for the Root Device **152**
 - Enabling Multipath I/O at Install Time **152** • Enabling Multipath I/O for an Existing Root Device **156** • Disabling Multipath I/O on the Root Device **156**
- 7.13 Configuring Multipath I/O for an Existing Software RAID **156**
- 7.14 Scanning for New Devices without Rebooting **159**
- 7.15 Scanning for New Partitioned Devices without Rebooting **163**
- 7.16 Viewing Multipath I/O Status **166**
- 7.17 Managing I/O in Error Situations **168**
- 7.18 Resolving Stalled I/O **169**
- 7.19 Troubleshooting MPIO **170**
 - PRIO Settings for Individual Devices Fail After Upgrading to Multipath 0.4.9 **170** • PRIO Settings with Arguments Fail After Upgrading to multipath-tools-0.4.9 **171** • Technical Information Documents **172**
- 7.20 What's Next **172**

8 Software RAID Configuration 173

- 8.1 Understanding RAID Levels **173**
 - RAID 0 **174** • RAID 1 **174** • RAID 2 and RAID 3 **174** • RAID 4 **175** • RAID 5 **175** • Nested RAID Levels **175**
- 8.2 Soft RAID Configuration with YaST **175**
- 8.3 Troubleshooting Software RAIDs **177**
- 8.4 For More Information **178**

9 Configuring Software RAID1 for the Root Partition 179

- 9.1 Prerequisites for Using a Software RAID1 Device for the Root Partition 179
- 9.2 Enabling iSCSI Initiator Support at Install Time 180
- 9.3 Enabling Multipath I/O Support at Install Time 180
- 9.4 Creating a Software RAID1 Device for the Root (/) Partition 181

10 Managing Software RAID6 and 10 with mdadm 185

- 10.1 Creating a RAID 6 185
 - Understanding RAID 6 185 • Creating a RAID 6 186
- 10.2 Creating Nested RAID 10 Devices with mdadm 187
 - Understanding Nested RAID Devices 187 • Creating Nested RAID 10 (1+0) with mdadm 188 • Creating Nested RAID 10 (0+1) with mdadm 190
- 10.3 Creating a Complex RAID 10 192
 - Understanding the Complex RAID10 192 • Creating a Complex RAID 10 with mdadm 196 • Creating a Complex RAID10 with the YaST Partitioner 197
- 10.4 Creating a Degraded RAID Array 201

11 Resizing Software RAID Arrays with mdadm 204

- 11.1 Understanding the Resizing Process 204
 - Guidelines for Resizing a Software RAID 204 • Overview of Tasks 205
- 11.2 Increasing the Size of a Software RAID 206
 - Increasing the Size of Component Partitions 206 • Increasing the Size of the RAID Array 208 • Increasing the Size of the File System 209
- 11.3 Decreasing the Size of a Software RAID 212
 - Decreasing the Size of the File System 213 • Decreasing the Size of the RAID Array 215 • Decreasing the Size of Component Partitions 216

12 Storage Enclosure LED Utilities for MD Software RAIDs 219

- 12.1 Supported LED Management Protocols 219
- 12.2 Supported Storage Enclosure Systems 220
- 12.3 Storage Enclosure LED Monitor Service (ledmon(8)) 220
 - Syntax 220 • Options 221 • Files 222 • Known Issues 222
- 12.4 Storage Enclosure LED Control Application (ledctl(8)) 222
 - Syntax 223 • Pattern Names 223 • Pattern Translation 225 • List of Devices 226 • Options 227 • Files 227 • Examples 228
- 12.5 Enclosure LED Utilities Configuration File (ledctl.conf(5)) 228
- 12.6 Additional Information 229

13 iSNS for Linux 230

- 13.1 How iSNS Works 230
- 13.2 Installing iSNS Server for Linux 232
- 13.3 Configuring iSNS Discovery Domains 233
 - Creating iSNS Discovery Domains 234 • Creating iSNS Discovery Domain Sets 235 • Adding iSCSI Nodes to a Discovery Domain 236 • Adding Discovery Domains to a Discovery Domain Set 238
- 13.4 Starting iSNS 238
- 13.5 Stopping iSNS 239
- 13.6 For More Information 239

14 Mass Storage over IP Networks: iSCSI 240

- 14.1 Installing iSCSI Target and Initiator 241
 - Installing iSCSI Target Software 242 • Installing the iSCSI Initiator Software 242

- 14.2 Setting Up an iSCSI Target 242
 - Preparing the Storage Space 243 • Creating iSCSI Targets with YaST 245 • Configuring an iSCSI Target Manually 248 • Configuring Online Targets with ietadm 250
- 14.3 Configuring iSCSI Initiator 252
 - Using YaST for the iSCSI Initiator Configuration 253 • Setting Up the iSCSI Initiator Manually 257 • The iSCSI Client Databases 258
- 14.4 Using iSCSI Disks when Installing 259
- 14.5 Troubleshooting iSCSI 260
 - Hotplug Doesn't Work for Mounting iSCSI Targets 260 • Data Packets Dropped for iSCSI Traffic 260 • Using iSCSI Volumes with LVM 261 • iSCSI Targets Are Mounted When the Configuration File Is Set to Manual 261
- 14.6 Additional Information 262
- 15 Mass Storage over IP Networks: iSCSI LIO Target Server 263**
 - 15.1 Installing the iSCSI LIO Target Server Software 263
 - 15.2 Starting the iSCSI LIO Target Service 266
 - Configuring iSCSI LIO Startup Preferences 266 • Manually Starting iSCSI LIO Target at the Command Line 268 • Manually Starting iSCSI LIO Target in YaST 269
 - 15.3 Configuring Authentication for Discovery of iSCSI LIO Targets and Clients 269
 - 15.4 Preparing the Storage Space 272
 - Partitioning Devices 273 • Partitioning Devices in a Virtual Environment 274
 - 15.5 Setting Up an iSCSI LIO Target Group 275
 - 15.6 Modifying an iSCSI LIO Target Group 283
 - 15.7 Deleting an iSCSI LIO Target Group 285

- 15.8 Troubleshooting iSCSI LIO Target Server 286
 - Portal Error When Setting Up Target LUNs 287 • iSCSI LIO Targets Are Not Visible from Other Computers 288
- 15.9 iSCSI LIO Target Terminology 288
- 16 Fibre Channel Storage over Ethernet Networks: FCoE 291**
 - 16.1 Installing FCoE and the YaST FCoE Client 292
 - 16.2 Configuring FCoE Interfaces during the Installation 293
 - 16.3 Managing FCoE Services with YaST 294
 - 16.4 Configuring FCoE with Commands 299
 - 16.5 Managing FCoE Instances with the FCoE Administration Tool 301
 - 16.6 Setting Up Partitions for an FCoE Initiator Disk 306
 - 16.7 Creating a File System on an FCoE Initiator Disk 307
 - 16.8 Additional Information 308
- 17 LVM Volume Snapshots 309**
 - 17.1 Understanding Volume Snapshots 309
 - 17.2 Creating Linux Snapshots with LVM 311
 - 17.3 Monitoring a Snapshot 311
 - 17.4 Deleting Linux Snapshots 312
 - 17.5 Using Snapshots for Virtual Machines on a Virtual Host 313
 - 17.6 Merging a Snapshot with the Source Logical Volume to Revert Changes or Roll Back to a Previous State 315
- 18 Managing Access Control Lists over NFSv4 318**
- 19 Troubleshooting Storage Issues 319**
 - 19.1 Is DM-MPIO Available for the Boot Partition? 319

- 19.2 Btrfs Error: No space is left on device **319**
Disk Space Consumed by Snapper Snapshots **319** • Disk Space Consumed by Log, Crash, and Cache Files **321**
- 19.3 Issues for Multipath I/O **321**
- 19.4 Issues for Software RAIDs **321**
- 19.5 Issues for iSCSI **321**
- 19.6 Issues for iSCSI LIO Target **322**

A GNU Licenses **323**

- A.1 GNU General Public License **323**
- A.2 GNU Free Documentation License **327**

B Documentation Updates **332**

- B.1 November 4, 2013 **333**
Managing Multipath I/O for Devices **333**
- B.2 October 14, 2013 **333**
iSNS for Linux **333**
- B.3 October 4, 2013 **334**
Managing Multipath I/O for Devices **334** • Volume Snapshots **335** • What's New for Storage in SLES 11 **336**
- B.4 June 2013 (SLES 11 SP3) **336**
LVM Configuration **336** • Mass Storage over IP Networks: iSCSI LIO Target Server **337** • Managing Multipath I/O for Devices **337** • Managing Software RAIDs 6 and 10 with mdadm **338** • Overview of File Systems in Linux **338** • Storage Enclosure LED Utilities for Software RAIDs **339** • Volume Snapshots **339** • Troubleshooting Storage Issues **339** • What's New for Storage in SLES 11 **339**
- B.5 March 19, 2013 **340**
Overview of File Systems in Linux **340**
- B.6 March 11, 2013 **340**
LVM Configuration **340** • Overview of File Systems in Linux **341**

- B.7 February 8, 2013 **341**
Configuring Software RAID1 for the Root Partition **341**
- B.8 January 8, 2013 **341**
LVM Configuration **342** • Managing Multipath I/O for Devices **342**
- B.9 November 14, 2012 **343**
Overview of File Systems in Linux **344**
- B.10 October 29, 2012 **344**
- B.11 October 19, 2012 **344**
Managing Multipath I/O for Devices **344** • Overview of File Systems in Linux **345**
- B.12 September 28, 2012 **345**
Managing Multipath I/O for Devices **345** • Overview of File Systems in Linux **346**
- B.13 April 12, 2012 **346**
Managing Multipath I/O for Devices **347** • Resizing Software RAID Arrays with mdadm **347**
- B.14 February 27, 2012 (SLES 11 SP2) **347**
Fibre Channel Storage over Ethernet Networks: FCoE **348** • GNU Licenses **348** • LVM Configuration **348** • Managing Access Control Lists over NFSv4 **348** • Managing Multipath I/O for Devices **348** • Managing Software RAIDs 6 and 10 with mdadm **349** • Mass Storage over IP Networks: iSCSI **349** • Overview of File Systems on Linux **350** • Resizing File Systems **350** • Resizing Software RAID Arrays with mdadm **351**
- B.15 July 12, 2011 **351**
Managing Multipath I/O for Devices **351**
- B.16 June 14, 2011 **351**
Managing Multipath I/O for Devices **352** • What's New for Storage in SLES 11 **352**
- B.17 May 5, 2011 **352**

- B.18 January 2011 **352**
LVM Configuration **353** • Managing Multipath I/O for Devices **353** • Resizing File Systems **353**
- B.19 September 16, 2010 **353**
LVM Configuration **354**
- B.20 June 21, 2010 **354**
LVM Configuration **355** • Managing Multipath I/O **355** • Managing Software RAIDs 6 and 10 with mdadm **355** • Mass Storage on IP Network: iSCSI **355**
- B.21 May 2010 (SLES 11 SP1) **355**
Managing Multipath I/O for Devices **356** • Mass Storage over IP Networks: iSCSI **357** • Software RAID Configuration **357** • What's New **357**
- B.22 February 23, 2010 **358**
Configuring Software RAID for the Root Partition **358** • Managing Multipath I/O **358**
- B.23 December 1, 2009 **359**
Managing Multipath I/O for Devices **359** • Resizing File Systems **359** • What's New **360**
- B.24 October 20, 2009 **360**
LVM Configuration **360** • Managing Multipath I/O for Devices **361** • What's New **361**
- B.25 August 3, 2009 **362**
Managing Multipath I/O **362**
- B.26 June 22, 2009 **362**
Managing Multipath I/O **362** • Managing Software RAIDs 6 and 10 with mdadm **363** • Mass Storage over IP Networks: iSCSI **363**
- B.27 May 21, 2009 **364**
Managing Multipath I/O **364**

About This Guide

This guide provides information about how to manage storage devices on a SUSE Linux Enterprise Server 12 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 <http://www.suse.com/doc/feedback.html> and enter your comments there.

Documentation Updates


For the most recent version of the *SUSE Linux Enterprise Server 12 Storage Administration Guide*, visit the SUSE Documentation Web site for SUSE Linux Enterprise Server 12 (<http://www.suse.com/documentation/sles12>).

Additional Documentation

For information about partitioning and managing devices, see ##### 15 #####.

1 Overview of File Systems in Linux

SUSE Linux Enterprise Server ships with a number of different file systems from which to choose, including Btrfs, 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 Server 12 High Availability Extension Guide* (http://www.suse.com/documentation/sle_ha/book_sleha/data/book_sleha.html) .

- *Section 1.1, “Terminology”*
- *Section 1.2, “Major File Systems in Linux”*
- *Section 1.3, “Other Supported File Systems”*
- *Section 1.4, “Large File Support in Linux”*
- *Section 1.5, “Linux Kernel Storage Limitations”*
- *Section 1.6, “Managing Devices with the YaST Partitioner”*
- *Section 1.7, “Additional Information”*

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"](#)
- [Section 1.2.2, "Ext2"](#)
- [Section 1.2.3, "Ext3"](#)
- [Section 1.2.4, "ReiserFS"](#)
- [Section 1.2.5, "XFS"](#)
- [Section 1.2.6, "File System Feature Comparison"](#)

1.2.1 Btrfs

Btrfs 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 twice the amount of disk space than the standard storage proposal. This is done automatically by the YaST Partitioner in the Btrfs storage proposal for the root file system.

- *Section 1.2.1.1, "Key Features"*
- *Section 1.2.1.2, "Bootloader Support"*
- *Section 1.2.1.3, "Btrfs Subvolumes"*
- *Section 1.2.1.4, "Snapshots for the Root File System"*
- *Section 1.2.1.5, "Online Check and Repair Functionality"*
- *Section 1.2.1.6, "RAID and Multipath Support"*
- *Section 1.2.1.7, "Migration from Ext File Systems to Btrfs"*
- *Section 1.2.1.8, "Btrfs Administration"*
- *Section 1.2.1.9, "Btrfs Quota Support for Subvolumes"*

1.2.1.1 Key Features

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. The feature is planned to be available in a future release of the YaST Partitioner.

Use Btrfs commands to set up transparent compression. Compression and Encryption functionality for Btrfs is currently under development and is currently not supported on SUSE Linux Enterprise Server.

- 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 and AutoYaST on SUSE Linux.
- Offline migration from existing Ext2, Ext3, and Ext4 file systems.

1.2.1.2 Bootloader Support

Bootloader support for `/boot` on Btrfs is planned to be available beginning in SUSE Linux Enterprise 12.

1.2.1.3 Btrfs Subvolumes

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, the YaST Partitioner automatically prepares the Btrfs file system for use with Btrfs subvolumes. You can cover any subdirectory as a subvolume. For example, [Table 1.1, “Default Subvolume Handling for Btrfs in YaST”](#) identifies the subdirectories that we recommend you treat as subvolumes because they contain files that you should not snapshot for the reasons given:

TABLE 1.1: DEFAULT SUBVOLUME HANDLING FOR BTRFS IN YAST

Path	Reason to Cover as a Subvolume
<code>/opt</code>	Contains third-party add-on application software packages.
<code>/srv</code>	Contains <code>http</code> and <code>ftp</code> files.
<code>/tmp</code>	Contains temporary files.
<code>/var/crash</code>	Contains memory dumps of crashed kernels.

Path	Reason to Cover as a Subvolume
<u>/var/log</u>	Contains system and applications' log files, which should never be rolled back.
<u>/var/run</u>	Contains run-time variable data.
<u>/var/spool</u>	Contains data that is awaiting processing by a program, user, or administrator, such as news, mail, and printer queues.
<u>/var/tmp</u>	Contains temporary files or directories that are preserved between system reboots.

After the installation, you can add or remove Btrfs subvolumes by using the YaST Expert Partitioner. For information, see [##### 15 ##### 15.1.2.1.1 ##### YaST ##### Btrfs #####](#).

1.2.1.4 Snapshots for the Root File System

Btrfs provides writable snapshots with the SUSE Snapper infrastructure that allow you to easily roll back your system if needed after applying updates, or to back up files. Snapper allows you to create and delete snapshots, and to compare snapshots and revert the differences between them. If Btrfs is used for the root (/) file system, YaST automatically enables snapshots for the root file system.

For information about Snapper and its integration in ZYpp ([snapper-zypp-plugin](#)) and YaST ([yast2-snapper](#)), see [##### 4 ##### Snapper #####](#).

To prevent snapshots from filling up the system disk, you can change the Snapper cleanup defaults to be more aggressive in the [/etc/snapper/configs/root](#) configuration file, or for other mount points. Snapper provides three algorithms to clean up old snapshots that are executed in a daily cron-job. The cleanup frequency is defined in the Snapper configuration for the mount point. Lower the `TIMELINE_LIMIT` parameters for daily, monthly, and yearly to reduce how long and the number of snapshots to be retained. For information, see [##### 4 ##### Snapper ##### 4.4 ##### Snapper ###](#).

For information about the SUSE Snapper project, see the [Snapper Portal wiki at OpenSUSE.org](http://en.opensuse.org/Portal:Snapper) (<http://en.opensuse.org/Portal:Snapper>) [↗](#).

1.2.1.5 Online Check and Repair Functionality

The **scrub** check and repair functionality is available as part of the Btrfs command line tools. It verifies the integrity of data and metadata, assuming the tree structures is fine. You can run **scrub** periodically on a mounted file system; it runs as a background process during normal operation.

1.2.1.6 RAID and Multipath Support

You can create Btrfs on Multiple Devices (MD) and Device Mapper (DM) storage configurations by using the YaST Partitioner.

1.2.1.7 Migration from Ext File Systems to Btrfs

You can migrate data volumes from existing Ext file systems (Ext2, Ext3, or Ext4) to the Btrfs file system. The conversion process occurs offline and in place on the device. The file system needs least 15% of available free space on the device.

To convert the Ext file system to Btrfs, take the file system offline, then enter:

```
btrfs-convert <device>
```

To roll back the migration to the original Ext file system, take the file system offline, then enter:

```
btrfs-convert -r <device>
```

Important

When rolling back to the original Ext file system, all data will be lost that you added after the conversion to Btrfs. That is, only the original data is converted back to the Ext file system.

1.2.1.8 Btrfs Administration

Btrfs is integrated in the YaST Partitioner and AutoYaST. It is available during the installation to allow you to set up a solution for the root file system. You can use the YaST Partitioner after the install to view and manage Btrfs volumes.

Btrfs administration tools are provided in the `btrfsprogs` package. For information about using Btrfs commands, see the `btrfs(8)`, `btrfsck(8)`, `mkfs.btrfs(8)`, and `btrfsctl(8)` man pages. For information about Btrfs features, see the *Btrfs wiki* (<http://btrfs.wiki.kernel.org>)⁷.

The `fsck.btrfs(8)` tool will soon be available in the SUSE Linux Enterprise update repositories.

1.2.1.9 Btrfs Quota Support for Subvolumes

The Btrfs `root` file system subvolumes `/var/log`, `/var/crash` and `/var/cache` can use all of the available disk space during normal operation, and cause a system malfunction. To help avoid this situation, SUSE Linux Enterprise now offers Btrfs quota support for subvolumes. See the `btrfs(8)` manual page for more details.

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 1.2.2.1, "Solidity and Speed"*
- *Section 1.2.2.2, "Easy Upgradability"*

1.2.2.1 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 only 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.

1.2.2.2 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 1.2.3.1, “Easy and Highly Reliable Upgrades from Ext2”*
- *Section 1.2.3.2, “Reliability and Performance”*
- *Section 1.2.3.3, “Converting an Ext2 File System into Ext3”*
- *Section 1.2.3.4, “Ext3 File System Inode Size and Number of Inodes”*

1.2.3.1 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. Perform a clean unmount of the Ext3 file system and remount it as an Ext2 file system.

1.2.3.2 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.

1.2.3.3 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`.

- a. Edit `/etc/dracut.conf.d/01-dist.conf`, the following line:

```
force_drivers+="ext3 jbd"
```

- b. Run the **dracut -f** command.

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

4. Reboot the system.

1.2.3.4 Ext3 File System Inode Size and Number of Inodes

An inode stores information about the file and its block location in the file system. To allow space in the inode for extended attributes and ACLs, the default inode size for Ext3 was increased from 128 bytes on SLES 10 to 256 bytes on SLES 11. As compared to SLES 10, when you make a new Ext3 file system on SLES 11, the default amount of space pre-allocated for the same number of inodes is doubled, and the usable space for files in the file system is reduced by that amount. Thus, you must use larger partitions to accommodate the same number of inodes and files than were possible for an Ext3 file system on SLES 10.

When you create a new Ext3 file system, the space in the inode table is pre-allocated for the total number of inodes that can be created. The bytes-per-inode ratio and the size of the file system determine how many inodes are possible. When the file system is made, an inode is created for every bytes-per-inode bytes of space:

```
number of inodes = total size of the file system divided by the number of bytes per
inode
```

The number of inodes controls the number of files you can have in the file system: one inode for each file. To address the increased inode size and reduced usable space available, the default for the bytes-per-inode ratio was increased from 8192 bytes on SLES 10 to 16384 bytes on SLES 11. The doubled ratio means that the number of files that can be created is one-half of the number of files possible for an Ext3 file system on SLES 10.

Important

After the inodes are allocated, you cannot change the settings for the inode size or bytes-per-inode ratio. No new inodes are possible without recreating the file system with different settings, or unless the file system gets extended. When you exceed the maximum number of inodes, no new files can be created on the file system until some files are deleted.

When you make a new Ext3 file system, you can specify the inode size and bytes-per-inode ratio to control inode space usage and the number of files possible on the file system. If the blocks size, inode size, and bytes-per-inode ratio values are not specified, the default values in the `/etc/mke2fs.conf` file are applied. For information, see the `mke2fs.conf(5)` man page.

Use the following guidelines:

- **Inode size:** The default inode size is 256 bytes. Specify a value in bytes that is a power of 2 and equal to 128 or larger in bytes and up to the block size, such as 128, 256, 512, and so on. Use 128 bytes only if you do not use extended attributes or ACLs on your Ext3 file systems.
- **Bytes-per-inode ratio:** The default bytes-per-inode ratio is 16384 bytes. Valid bytes-per-inode ratio values must be a power of 2 equal to 1024 or greater in bytes, such as 1024, 2048, 4096, 8192, 16384, 32768, and so on. This value should not be smaller than the block size of the file system, because the block size is the smallest chunk of space used to store data. The default block size for the Ext3 file system is 4 KB.

In addition, you should consider the number of files and the size of files you need to store. For example, if your file system will have many small files, you can specify a smaller bytes-per-inode ratio, which increases the number of inodes. If your file system will have a very large files, you can specify a larger bytes-per-inode ratio, which reduces the number of possible inodes.

Generally, it is better to have too many inodes than to run out of them. If you have too few inodes and very small files, you could reach the maximum number of files on a disk that is practically empty. If you have too many inodes and very large files, you might have free space reported but be unable to use it because you cannot create new files in space reserved for inodes.

If you do not use extended attributes or ACLs on your Ext3 file systems, you can restore the SLES 10 behavior specifying 128 bytes as the inode size and 8192 bytes as the bytes-per-inode ratio when you make the file system. Use any of the following methods to set the inode size and bytes-per-inode ratio:

- **Modifying the default settings for all new Ext3 files:** In a text editor, modify the `defaults` section of the `/etc/mke2fs.conf` file to set the `inode_size` and `inode_ratio` to the desired default values. The values apply to all new Ext3 file systems. For example:

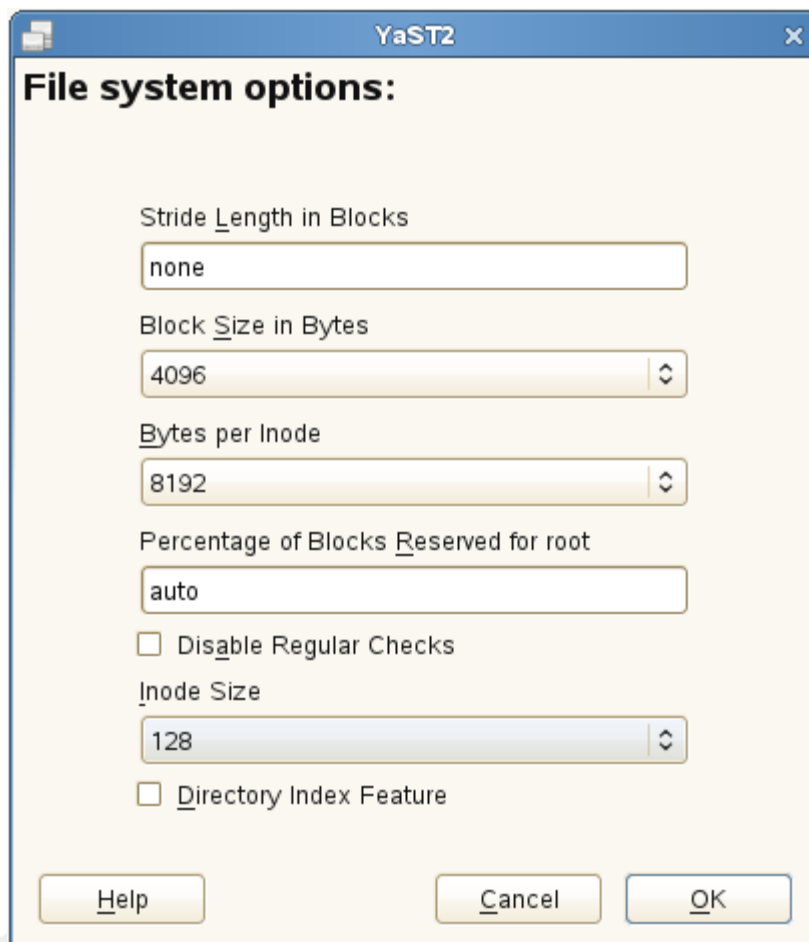
```
blocksize = 4096
inode_size = 128
inode_ratio = 8192
```

- **At the command line:** Pass the inode size (`-I 128`) and the bytes-per-inode ratio (`-i 8192`) to the `mkfs.ext3(8)` command or the `mke2fs(8)` command when you create a new Ext3 file system. For example, use either of the following commands:

```
mkfs.ext3 -b 4096 -i 8092 -I 128 /dev/sda2

mke2fs -t ext3 -b 4096 -i 8192 -I 128 /dev/sda2
```

- **During installation with YaST:** Pass the inode size and bytes-per-inode ratio values when you create a new Ext3 file system during the installation. In the YaST Partitioner on the *Edit Partition* page under *Formatting Options*, select *Format partitionExt3*, then click *Options*. In the *File system options* dialog box, select the desired values from the *Block Size in Bytes*, *Bytes-per-inode*, and *Inode Size* drop-down lists. For example, select 4096 for the *Block Size in Bytes* drop-down list, select 8192 from the *Bytes per inode* drop-down list, select 128 from the *Inode Size* drop-down list, then click *OK*.




- During installation with autoyast: In an autoyast profile, you can use the `fs_options` tag to set the `opt_bytes_per_inode` ratio value of 8192 for `-i` and the `opt_inode_density` value of 128 for `-I`:

```
<partitioning config:type="list">
  <drive>
    <device>/dev/sda</device>
    <initialize config:type="boolean">true</initialize>
    <partitions config:type="list">
      <partition>
        <filesystem config:type="symbol">ext3</filesystem>
        <format config:type="boolean">true</format>
        <fs_options>
          <opt_bytes_per_inode>
```

```

        <option_str>-i</option_str>
        <option_value>8192</option_value>
    </opt_bytes_per_inode>
    <opt_inode_density>
        <option_str>-I</option_str>
        <option_value>128</option_value>
    </opt_inode_density>
</fs_options>
<mount>/</mount>
<partition_id config:type="integer">131</partition_id>
<partition_type>primary</partition_type>
<size>25G</size>
</partition>

```

For information, see *SLES11 ext3 partitions can only store 50% of the files that can be stored on SLES10* [Technical Information Document 7009075] (<http://www.novell.com/support/kb/doc.php?id=7009075>) .

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

Existing ReiserFS partitions are supported for the lifetime of SUSE Linux Enterprise Server 12 specifically for migration purposes. Support for creating new ReiserFS file systems has been removed starting with SUSE Linux Enterprise Server 12.

- *Section 1.2.4.1, "Better Disk Space Utilization"*
- *Section 1.2.4.2, "Better Disk Access Performance"*

- [Section 1.2.4.3, “Fast Crash Recovery”](#)
- [Section 1.2.4.4, “Reliability through Data Journaling”](#)

1.2.4.1 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 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.

1.2.4.2 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.

1.2.4.3 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.

1.2.4.4 Reliability through Data Journaling

ReiserFS also supports data journaling and ordered data modes similar to the concepts outlined in [Section 1.2.3, “Ext3”](#). 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 1.2.5.1, "High Scalability through the Use of Allocation Groups"*
- *Section 1.2.5.2, "High Performance through Efficient Management of Disk Space"*
- *Section 1.2.5.3, "Preallocation to Avoid File System Fragmentation"*

1.2.5.1 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.

1.2.5.2 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.

1.2.5.3 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 File System 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 \(http://www.suse.com/products/server/technical-information/#FileSystem\)](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 1.2, "File System Types in Linux" 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 1.2: FILE SYSTEM TYPES IN LINUX

File System Type	Description
<u>cramfs</u>	Compressed ROM file system: A compressed read-only file system for ROMs.
<u>hpfs</u>	High Performance File System: The IBM OS/2 standard file system. Only supported in read-only mode.
<u>iso9660</u>	Standard file system on CD-ROMs.

File System Type	Description
<u>minix</u>	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.
<u>msdos</u>	<u>fat</u> , the file system originally used by DOS, is today used by various operating systems.
<u>ncpfs</u>	File system for mounting Novell volumes over networks.
<u>nfs</u>	Network File System: Here, data can be stored on any machine in a network and access might be granted via a network.
<u>ntfs</u>	Windows NT file system; read-only.
<u>smbfs</u>	Server Message Block is used by products such as Windows to enable file access over a network.
<u>sysv</u>	Used on SCO UNIX, Xenix, and Coherent (commercial UNIX systems for PCs).
<u>ufs</u>	Used by BSD, SunOS, and NextStep. Only supported in read-only mode.
<u>umsdos</u>	UNIX on MS-DOS: Applied on top of a standard <u>fat</u> file system, achieves UNIX functionality (permissions, links, long filenames) by creating special files.
<u>vfat</u>	Virtual FAT: Extension of the <u>fat</u> file system (supports long filenames).

1.4 Large File Support in Linux

Originally, Linux supported a maximum file size of 2 GiB (2^{31} bytes). Unless a file system comes with large file support, the maximum file size on a 32-bit system is 2 GiB.

Currently, all of our standard file systems have LFS (large file support), which gives a maximum file size of 2^{63} bytes in theory. *Table 1.3, “Maximum Sizes of Files and File Systems (On-Disk Format, 4 KiB Block Size)”* offers an overview of the current on-disk format limitations of Linux files and file systems. The numbers in the table assume that the file systems are using 4 KiB block size, which is a common standard. When using different block sizes, the results are different. The maximum file sizes in *Table 1.3, “Maximum Sizes of Files and File Systems (On-Disk Format, 4 KiB Block Size)”* can be larger than the file system's actual size when using sparse blocks.



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 1.3: MAXIMUM SIZES OF FILES AND FILE SYSTEMS (ON-DISK FORMAT, 4 KIB BLOCK SIZE)

File System (4 KiB Block Size)	Maximum File System Size	Maximum File Size
Btrfs	16 EiB	16 EiB
Ext3	16 TiB	2 TiB
OCFS2 (a cluster-aware file system available in the High Availability Extension)	16 TiB	1 EiB
ReiserFS v3.6	16 TiB	1 EiB
XFS	8 EiB	8 EiB
NFSv2 (client side)	8 EiB	2 GiB
NFSv3 (client side)	8 EiB	8 EiB

Important

Table 1.3, “Maximum Sizes of Files and File Systems (On-Disk Format, 4 KiB Block Size)” 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 Linux Kernel Storage Limitations

Table 1.4, “Storage Limitations” summarizes the kernel limits for storage associated with SUSE Linux Enterprise Server.

TABLE 1.4: STORAGE LIMITATIONS

Storage Feature	Limitation
Maximum number of LUNs supported	16384 LUNs per target.
Maximum number of paths per single LUN	No limit per se. Each path is treated as a normal LUN. The actual limit is given by the number of LUNs per target and the number of targets per HBA (16777215 for a Fibre Channel HBA).
Maximum number of HBAs	Unlimited. The actual limit is determined by the amount of PCI slots of the system.

Storage Feature	Limitation
Maximum number of paths with device-mapper-multipath (in total) per operating system	Approximately 1024. The actual number depends on the length of the device number strings. It is a compile-time variable within multipath-tools, which can be raised if this limit poses to be a problem.
Maximum size per block device	For X86, up to 16 TiB. For x86_64, ia64, s390x, and ppc64, up to 8 EiB.

1.6 Managing Devices with the YaST Partitioner

You can use the YaST Partitioner to create and manage file systems and RAID devices. For information, see [Chapter 15](#).

1.7 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/>)
- *XFS: A High-Performance Journaling Filesystem* (<http://oss.sgi.com/projects/xfs/>)
- *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).

2 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 SP3"*
- *Section 2.2, "What's New in SLES 11 SP2"*
- *Section 2.3, "What's New in SLES 11 SP1"*
- *Section 2.4, "What's New in SLES 11"*

2.1 What's New in SLES 11 SP3

In addition to bug fixes, the features and behavior changes in this section were made for the SUSE Linux Enterprise Server 11 SP3 release:

- *Section 2.1.1, "Btrfs Quotas"*
- *Section 2.1.2, "iSCSI LIO Target Server"*
- *Section 2.1.3, "Linux Software RAID5"*
- *Section 2.1.4, "LVM2"*
- *Section 2.1.5, "Multipath I/O"*

2.1.1 Btrfs Quotas

Btrfs quota support for subvolumes on the root file system has been added in the **btrfs(8)** command.

2.1.2 iSCSI LIO Target Server

YaST supports the iSCSI LIO Target Server software. For information, see *Chapter 15, Mass Storage over IP Networks: iSCSI LIO Target Server*.

2.1.3 Linux Software RAIDs

The following enhancements were added for Linux software RAIDs:

- [Section 2.1.3.1, “Support for Intel RSTe+”](#)
- [Section 2.1.3.2, “LEDMON Utility”](#)
- [Section 2.1.3.3, “Device Order in the Software RAID”](#)

2.1.3.1 Support for Intel RSTe+

The software RAID provides improved support on the Intel RSTe+ (Rapid Storage Technology Enterprise) platform to support RAID level 0, 1, 4, 5, 6, and 10.

2.1.3.2 LEDMON Utility

The LEDMON utility supports PCIe-SSD enclosure LEDs for MD software RAIDs. For information, see [Chapter 12, Storage Enclosure LED Utilities for MD Software RAIDs](#).

2.1.3.3 Device Order in the Software RAID

In the *Add RAID* wizard in the YaST Partitioner, the *Classify* option allows you to specify the order in which the selected devices in a Linux software RAID will be used to ensure that one half of the array resides on one disk subsystem and the other half of the array resides on a different disk subsystem. For example, if one disk subsystem fails, the system keeps running from the second disk subsystem. For information, see [Step 4.d in Section 10.3.3, “Creating a Complex RAID10 with the YaST Partitioner”](#).

2.1.4 LVM2

The following enhancements were added for LVM2:

- [Section 2.1.4.1, “Thin Pool and Thin Volumes”](#)
- [Section 2.1.4.2, “Thin Snapshots”](#)

2.1.4.1 Thin Pool and Thin Volumes

LVM logical volumes can be thinly provisioned. For information, see [Section 4.5, “Configuring Logical Volumes”](#).

- **Thin pool:** The logical volume is a pool of space that is reserved for use with thin volumes. The thin volumes can allocate their needed space from it on demand.
- **Thin volume:** The volume is created as a sparse volume. The volume allocates needed space on demand from a thin pool.

2.1.4.2 Thin Snapshots

LVM logical volume snapshots can be thinly provisioned. Thin provisioning is assumed if you to create a snapshot without a specified size. For information, see [Section 17.2, “Creating Linux Snapshots with LVM”](#).

2.1.5 Multipath I/O

The following changes and enhancements were made for multipath I/O:

- [Section 2.1.5.1, “mpathpersist\(8\)”](#)
- [Section 2.1.5.2, “multipath\(8\)”](#)
- [Section 2.1.5.3, “/etc/multipath.conf”](#)

2.1.5.1 mpathpersist(8)

The **mpathpersist(8)** utility is new. It can be used to manage SCSI persistent reservations on Device Mapper Multipath devices. For information, see [Section 7.3.5, “Linux mpathpersist\(8\) Utility”](#).

2.1.5.2 `multipath(8)`

The following enhancement was added to the `multipath(8)` command:

- The `-r` option allows you to force a device map reload.

2.1.5.3 `/etc/multipath.conf`

The Device Mapper - Multipath tool added the following enhancements for the `/etc/multipath.conf` file:

- `udev_dir`.

The `udev_dir` attribute is deprecated. After you upgrade to SLES 11 SP3, you can remove the following line from the `defaults` section of your `/etc/multipath.conf` file:

```
udev_dir /dev
```

- `getuid_callout`.

In the `defaults` section of the `/etc/multipath.conf` file, the `getuid_callout` attribute is deprecated and replaced by the `uid_attribute` parameter. This parameter is a udev attribute that provides a unique path identifier. The default value is `ID_SERIAL`. After you upgrade to SLES 11 SP3, you can modify the attributes in the `defaults` section of your `/etc/multipath.conf` file:

- Remove the following line from the `defaults` section:

```
getuid_callout "/lib/udev/scsi_id --whitelisted --device=/dev/%n"
```

- Add the following line to the `defaults` section:

```
uid_attribute "ID_SERIAL"
```

- `path_selector`.

In the `defaults` section of the `/etc/multipath.conf` file, the default value for the `path_selector` attribute was changed from `"round-robin 0"` to `"service-time 0"`. The `service-time` option chooses the path for the next bunch of I/O based on the amount of outstanding I/O to the path and its relative throughput.

After you upgrade to SLES 11 SP3, you can modify the attribute value in the `defaults` section of your `/etc/multipath.conf` file to use the recommended default:

```
path_selector "service-time 0"
```

- **user_friendly_names.**

The `user_friendly_names` attribute can be configured in the `devices` section and in the `multipaths` section.

- **max_fds.**

The default setting for the `max_fds` attribute was changed to `max`. This allows the multipath daemon to open as many file descriptors as the system allows when it is monitoring paths.

After you upgrade to SLES 11 SP3, you can modify the attribute value in your `/etc/multipath.conf` file:

```
max_fds "max"
```

- **reservation_key.**

In the `defaults` section or `multipaths` section of the `/etc/multipath.conf` file, the `reservation_key` attribute can be used to assign a Service Action Reservation Key that is used with the `mpathpersist(8)` utility to manage persistent reservations for Device Mapper Multipath devices. The attribute is not used by default. If it is not set, the `multipathd` daemon does not check for persistent reservation for newly discovered paths or reinstated paths.

```
reservation_key <reservation key>
```

For example:

```

multipaths {
    multipath {
        wwid      XXXXXXXXXXXXXXXX
        alias      yellow
        reservation_key  0x123abc
    }
}

```

For information about setting persistent reservations, see [Section 7.3.5, “Linux mpathpersist\(8\) Utility”](#).

- **hardware_handler.**

Four SCSI hardware handlers were added in the SCSI layer that can be used with DM-Multipath:

[scsi_dh_alua](#)
[scsi_dh_rdac](#)
[scsi_dh_hp_sw](#)
[scsi_dh_emc](#)

These handlers are modules created under the SCSI directory in the Linux kernel. Previously, the hardware handler in the Device Mapper layer was used.

Add the modules to the [initrd](#) image, then specify them in the [/etc/multipath.conf](#) file as hardware handler types [alua](#), [rdac](#), [hp_sw](#), and [emc](#). For information about adding the device drivers to the [initrd](#) image, see [Section 7.4.3, “Configuring the Device Drivers in initrd for Multipathing”](#).

2.2 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 File System. See [Section 1.2.1, “Btrfs”](#).
- Open Fibre Channel over Ethernet. See [Chapter 16, Fibre Channel Storage over Ethernet Networks: FCoE](#).
- Tagging for LVM storage objects. See [Section 4.7, “Tagging LVM2 Storage Objects”](#).
- NFSv4 ACLs tools. See [Chapter 18, Managing Access Control Lists over NFSv4](#).

- `--assume-clean` option for `mdadm resize` command. See [Section 11.2.2, “Increasing the Size of the RAID Array”](#).
- In the `defaults` section of the `/etc/multipath.conf` file, the `default_getuid` parameter was obsoleted and replaced by the `getuid_callout` parameter:

The line changed from this:

```
default_getuid "/sbin/scsi_id -g -u -s /block/%n"
```

to this:

```
getuid_callout "/lib/udev/scsi_id --whitelisted --device=/dev/%n"
```

2.3 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 2.3.1, “Saving iSCSI Target Information”](#)
- [Section 2.3.2, “Modifying Authentication Parameters in the iSCSI Initiator”](#)
- [Section 2.3.3, “Allowing Persistent Reservations for MPIO Devices”](#)
- [Section 2.3.4, “MDADM 3.0.2”](#)
- [Section 2.3.5, “Boot Loader Support for MDRAID External Metadata”](#)
- [Section 2.3.6, “YaST Install and Boot Support for MDRAID External Metadata”](#)
- [Section 2.3.7, “Improved Shutdown for MDRAID Arrays that Contain the Root File System”](#)
- [Section 2.3.8, “MD over iSCSI Devices”](#)
- [Section 2.3.9, “MD-SGPIIO”](#)
- [Section 2.3.10, “Resizing LVM 2 Mirrors”](#)
- [Section 2.3.11, “Updating Storage Drivers for Adapters on IBM Servers”](#)

2.3.1 Saving iSCSI Target Information

In the *YaSTNetwork ServicesiSCSI 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.

2.3.2 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.

2.3.3 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.

2.3.4 MDADM 3.0.2

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

2.3.5 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.

2.3.6 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.

2.3.7 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 environment 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.

2.3.8 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`.

2.3.9 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)

2.3.10 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.

2.3.11 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

2.4 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 2.4.1, "EVMS2 Is Deprecated"*
- *Section 2.4.2, "Ext3 as the Default File System"*
- *Section 2.4.3, "Default Inode Size Increased for Ext3"*
- *Section 2.4.4, "JFS File System Is Deprecated"*
- *Section 2.4.5, "OCFS2 File System Is in the High Availability Release"*
- *Section 2.4.6, "/dev/disk/by-name Is Deprecated"*
- *Section 2.4.7, "Device Name Persistence in the /dev/disk/by-id Directory"*
- *Section 2.4.8, "Filters for Multipathed Devices"*
- *Section 2.4.9, "User-Friendly Names for Multipathed Devices"*
- *Section 2.4.10, "Advanced I/O Load-Balancing Options for Multipath"*
- *Section 2.4.11, "Location Change for Multipath Tool Callouts"*
- *Section 2.4.12, "Change from mpath to multipath for the mkinitrd -f Option"*
- *Section 2.4.13, "Change from Multibus to Failover as the Default Setting for the MPIIO Path Grouping Policy"*

2.4.1 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 non-system EVMS-managed devices should be automatically recognized and managed by Linux Volume Manager 2 (LVM2) when you upgrade your system.

If you have EVMS managing the system device (any device that contains the root (/), /boot, or swap), try these things to prepare the SLES 10 server before you reboot the server to upgrade:

1. In the /etc/fstab file, modify the boot and swap disks to the default /dev/system/sys_lx directory:
 - a. Remove /evms/lvm2 from the path for the swap and root (/) partitions.
 - b. Remove /evms from the path for /boot partition.
2. In the /boot/grub/menu.lst file, remove /evms/lvm2 from the path.
3. In the /etc/sysconfig/bootloader file, verify that the path for the boot device is the /dev directory.
4. Ensure that boot.lvm and boot.md are enabled:
 - a. In YaST, click *SystemRunlevel EditorExpert Mode*.
 - b. Select boot.lvm.
 - c. Click *Set/ResetEnable the Service*.
 - d. Select boot.md.
 - e. Click *Set/ResetEnable the Service*.
 - f. Click *Finish*, then click *Yes*.
5. Reboot and start the upgrade.

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.suse.com/documentation/sles10/stor_admin/data/bookinfo.html) .

2.4.2 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.suse.com/products/server/technical-information/?tab=2>) on the *SUSE Linux Enterprise 11 Tech Specs* Web page.

2.4.3 Default Inode Size Increased for Ext3

To allow space for extended attributes and ACLs for a file on Ext3 file systems, the default inode size for Ext3 was increased from 128 bytes on SLES 10 to 256 bytes on SLES11. For information, see *Section 1.2.3.4, "Ext3 File System Inode Size and Number of Inodes"*.

2.4.4 JFS File System Is Deprecated

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

2.4.5 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.

2.4.6 /dev/disk/by-name Is Deprecated

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

2.4.7 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 2.4.8, “Filters for Multipathed Devices”](#)
- [Section 2.4.9, “User-Friendly Names for Multipathed Devices”](#)

2.4.8 Filters for Multipathed Devices

The deprecation of the `/dev/disk/by-name` directory (as described in [Section 2.4.6, “/dev/disk/by-name Is Deprecated”](#)) 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 `pvs` 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 7.2.4, “Using LVM2 on Multipath Devices”](#).

2.4.9 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 2.4.7, “Device Name Persistence in the /dev/disk/by-id Directory”](#)) 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`.

2.4.10 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 *path_selector* in *Section 7.11.2.1, “Understanding Priority Groups and Attributes”*.

2.4.11 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.

2.4.12 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
```

2.4.13 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 7.11, “Configuring Path Failover Policies and Priorities”](#).

Draft

3 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 server.

- *Section 3.1, “Partitioning Devices”*
- *Section 3.2, “Multipath Support”*
- *Section 3.3, “Software RAID Support”*
- *Section 3.4, “File System Snapshots”*
- *Section 3.5, “Backup and Antivirus Support”*

3.1 Partitioning Devices

For information about using the YaST Expert Partitioner, see [##### ## 15 #####](#)
[## 15.1 ##### YaST #####](#).

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”*
- *Section 3.5.2, “Commercial Backup and Antivirus Support”*


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.

4 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"*
- *Section 4.2, "Creating LVM Partitions"*
- *Section 4.3, "Creating Volume Groups"*
- *Section 4.4, "Configuring Physical Volumes"*
- *Section 4.5, "Configuring Logical Volumes"*
- *Section 4.6, "Automatically Activating Non-Root LVM Volume Groups"*
- *Section 4.7, "Tagging LVM2 Storage Objects"*
- *Section 4.8, "Resizing a Volume Group"*
- *Section 4.9, "Resizing a Logical Volume with YaST"*
- *Section 4.10, "Resizing a Logical Volume with Commands"*
- *Section 4.11, "Deleting a Volume Group"*
- *Section 4.12, "Deleting an LVM Partition (Physical Volume)"*
- *Section 4.13, "Using LVM Commands"*

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" 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.

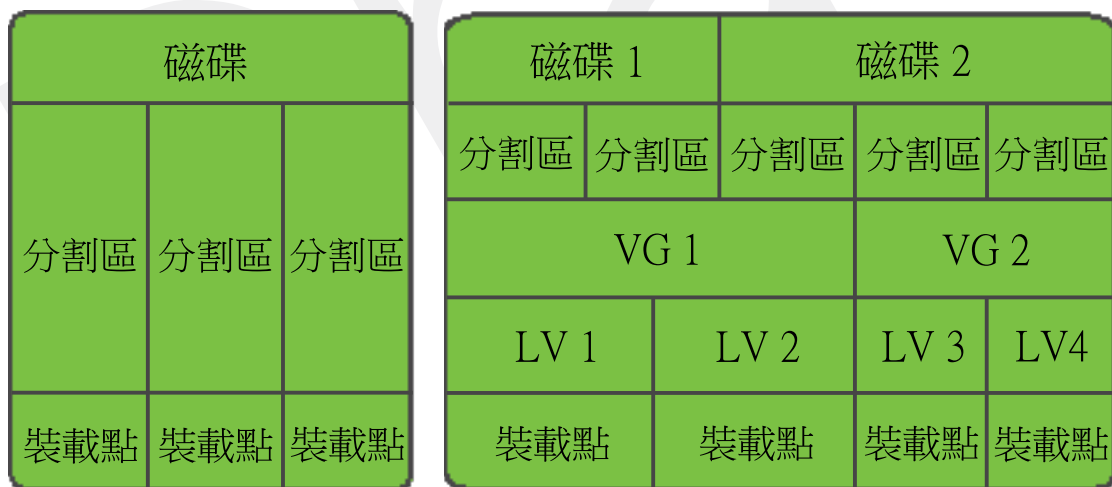


FIGURE 4.1: PHYSICAL PARTITIONING VERSUS LVM


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"*, 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 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/>) .

Important

If you add multipath support after you have configured LVM, you must modify the `/etc/lvm/lvm.conf` file to scan only the multipath device names in the `/dev/disk/by-id` directory as described in *Section 7.2.4, “Using LVM2 on Multipath Devices”*, then reboot the server.

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. Launch YaST as the root user.
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 4.12, “Deleting an LVM Partition \(Physical Volume\)”](#).
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*.

<p>New Partition Size</p> <p><input type="radio"/> Maximum Size (305.92 MB)</p> <p><input checked="" type="radio"/> Custom Size</p> <p>Size (e.g. 9.0 MB or 9.0 GB)</p> <p><input type="text" value="250 MB"/></p> <p><input type="radio"/> Custom Region</p> <p>Start Cylinder</p> <p><input type="text" value="0"/></p> <p>End Cylinder</p> <p><input type="text" value="38"/></p>	<ul style="list-style-type: none"> • 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:

<p>Formatting Options</p> <p><input type="radio"/> Format partition</p> <p>File system</p> <p><input type="text" value="Ext3"/></p> <p><input type="button" value="Options..."/></p> <p><input checked="" type="radio"/> Do not format partition</p> <p>File system ID:</p> <p><input type="text" value="0x8E Linux LVM"/></p> <p><input type="checkbox"/> Encrypt device</p>	<ol style="list-style-type: none"> 1. Under <i>Formatting Options</i>, select <i>Do not format</i>. 2. From the <i>File System ID</i> drop-down list, select <i>0x8E Linux LVM</i> as the partition identifier. 3. Under <i>Mounting Options</i>, select <i>Do not mount partition</i>.
--	--

8. Click *Finish*.

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

9. Repeat [Step 4](#) through [Step 8](#) for each Linux LVM partition you want to add.

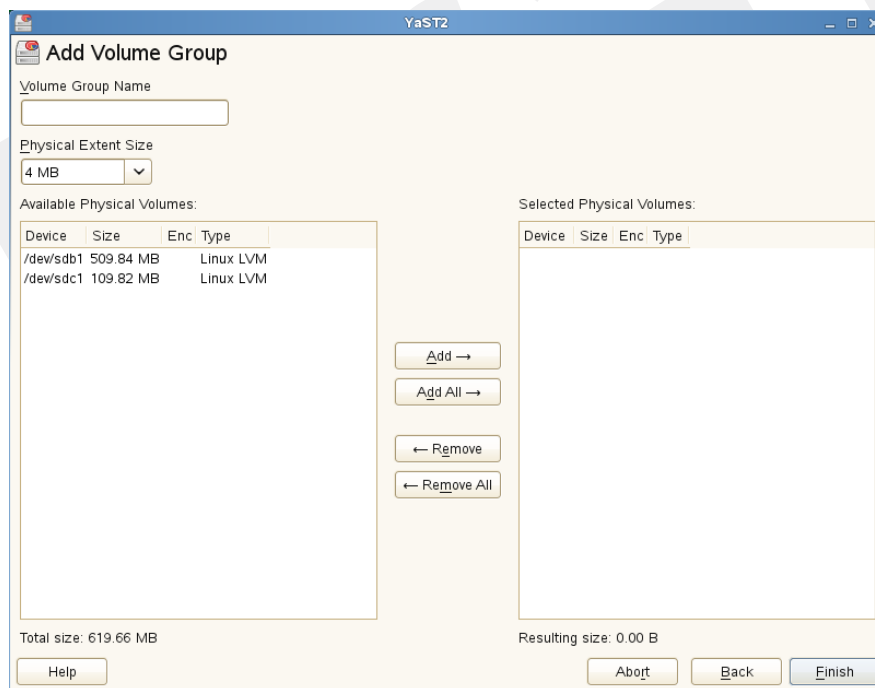
10. Click *Next*, verify that the new Linux LVM partitions are listed, then click *Finish* to exit the partitioner.

11. (Optional) Continue with the Volume Group configuration as described in [Section 4.3, "Creating Volume Groups"](#).

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. Launch YaST as the root user.
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.

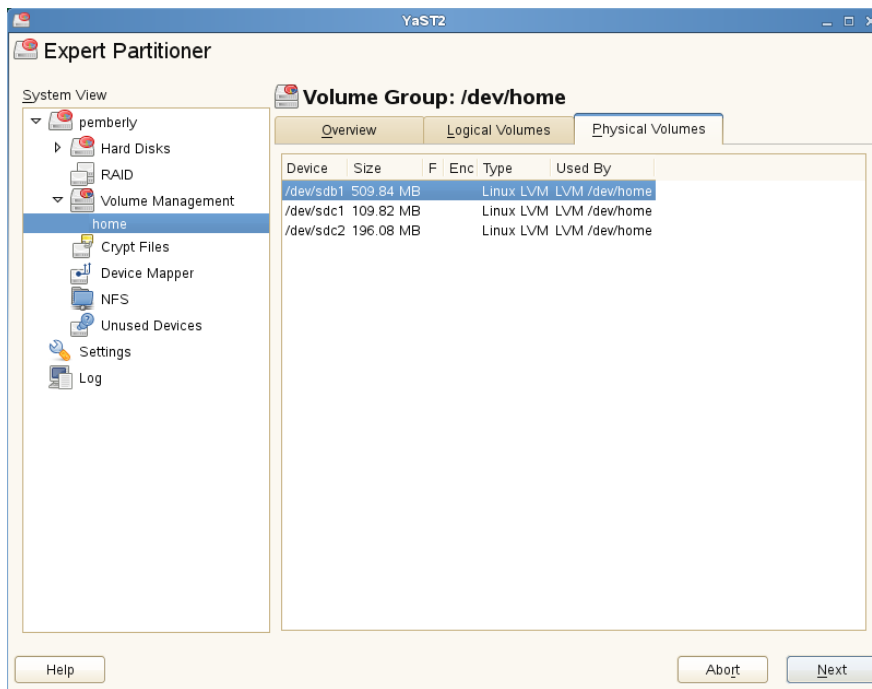
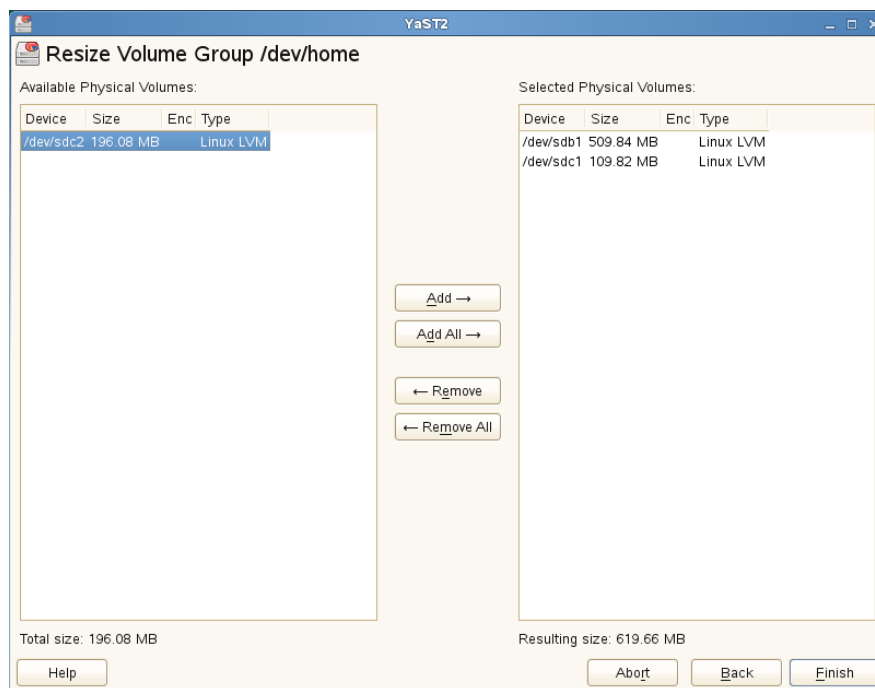


FIGURE 4.2: PHYSICAL VOLUMES IN THE VOLUME GROUP NAMED HOME

To add more physical volumes to an existing volume group:

1. Launch YaST as the root user.
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 box (see [Figure 4.3, "Logical Volume Management"](#)) 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.

Beginning in SLES 11 SP3, an LVM logical volume can be thinly provisioned. Thin provisioning allows you to create logical volumes with sizes that overbook the available free space. You create a thin pool that contains unused space reserved for use with an arbitrary number of thin

volumes. A thin volume is created as a sparse volume and space is allocated from a thin pool as needed. The thin pool can be expanded dynamically when needed for cost-effective allocation of storage space.

! Important

To use thinly provisioned volumes in a cluster, the thin pool and the thin volumes that use it must be managed in a single cluster resource. This allows the thin volumes and thin pool to always be mounted exclusively on the same node.

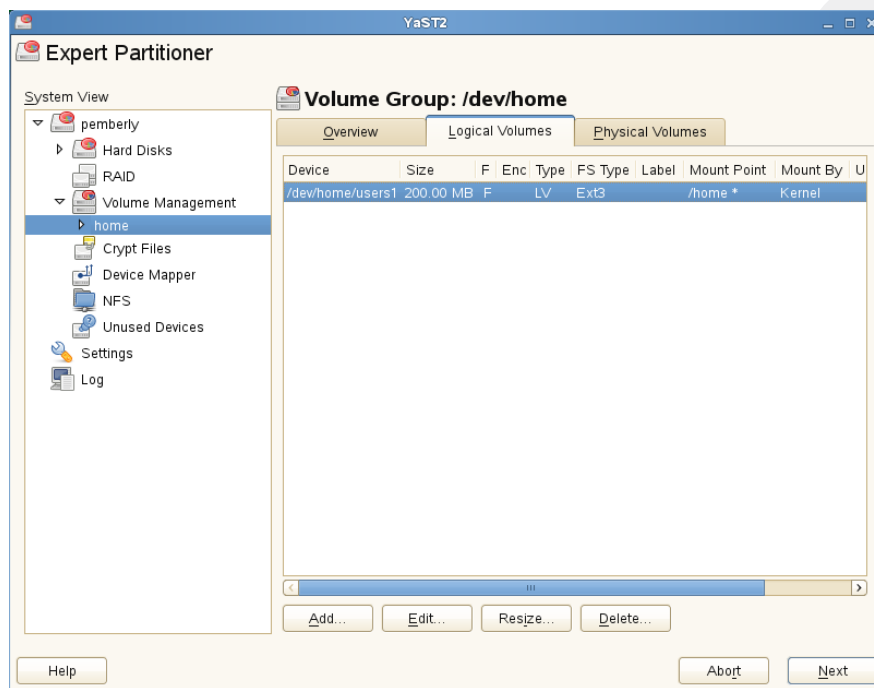
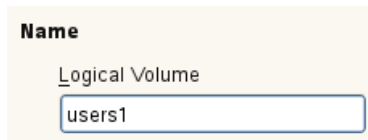


FIGURE 4.3: LOGICAL VOLUME MANAGEMENT

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. Launch YaST as the root user.
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 box.
6. Specify the *Name* for the logical volume, then click *Next*.



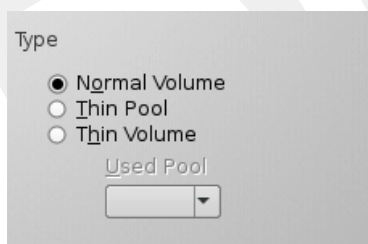
Name

Logical Volume

users1

7. Specify the type of LVM volume:

- **Normal volume:** (Default) The volume's space is allocated immediately.
- **Thin pool:** The logical volume is a pool of space that is reserved for use with thin volumes. The thin volumes can allocate their needed space from it on demand.
- **Thin volume:** The volume is created as a sparse volume. The volume allocates needed space on demand from a thin pool.



Type

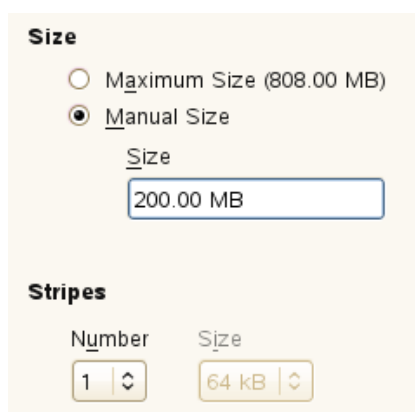
☒ Normal Volume

☐ Thin Pool

☐ Thin Volume

Used Pool

8. 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

1

Size

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.

9. Specify the formatting options for the logical volume:

The screenshot shows a dialog box with two main sections: **Formatting Options** and **Mounting Options**. In the **Formatting Options** section, the **Format partition** radio button is selected. Below it, the **File system** dropdown menu is set to **Ext3**, and there is an **Options...** button. The **Do not format partition** and **Encrypt device** options are unselected. In the **Mounting Options** section, the **Mount partition** radio button is selected. Below it, the **Mount Point** dropdown menu is set to **/home**, and there is an **Fstab Options...** button. The **Do not mount partition** option is unselected.

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.

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

4.6 Automatically Activating Non-Root LVM Volume Groups

Activation behavior for non-root LVM volume groups is controlled by parameter settings in the `/etc/sysconfig/lvm` file.

By default, non-root LVM volume groups are automatically activated on system restart by `/etc/rc.d/boot.lvm`, according to the setting for the `LVM_VGS_ACTIVATED_ON_BOOT` parameter in the `/etc/sysconfig/lvm` file. This parameter allows you to activate all volume groups on system restart, or to activate only specified non-root LVM volume groups.

To activate all non-root LVM volume groups on system restart, ensure that the value for the `LVM_VGS_ACTIVATED_ON_BOOT` parameter in the `/etc/sysconfig/lvm` file is empty (`" "`). This is the default setting. For almost all standard LVM installations, it can safely stay empty.

```
LVM_VGS_ACTIVATED_ON_BOOT=""
```

To activate only a specified non-root LVM volume group on system restart, specify the volume group name as the value for the `LVM_VGS_ACTIVATED_ON_BOOT` parameter:

```
LVM_VGS_ACTIVATED_ON_BOOT="vg1"
```

By default, newly discovered LVM volume groups are not automatically activated. The `LVM_ACTIVATED_ON_DISCOVERED` parameter is disabled in the `/etc/sysconfig/lvm` file:

```
LVM_ACTIVATED_ON_DISCOVERED="disable"
```

You can enable the `LVM_ACTIVATED_ON_DISCOVERED` parameter to allow newly discovered LVM volume groups to be activated via udev rules:

```
LVM_ACTIVATED_ON_DISCOVERED="enable"
```

4.7 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 4.7.1, “Using LVM2 Tags”*
- *Section 4.7.2, “Requirements for Creating LVM2 Tags”*
- *Section 4.7.3, “Command Line Tag Syntax”*
- *Section 4.7.4, “Configuration File Syntax”*
- *Section 4.7.5, “Using Tags for a Simple Activation Control in a Cluster”*
- *Section 4.7.6, “Using Tags to Activate On Preferred Hosts in a Cluster”*

4.7.1 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.

4.7.2 Requirements for Creating LVM2 Tags

Consider the following requirements when using tags with LVM:

- *Section 4.7.2.1, “Supported Characters”*
- *Section 4.7.2.2, “Supported Storage Objects”*

4.7.2.1 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.

4.7.2.2 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.

4.7.3 Command Line Tag Syntax

`--addtag <tag_info>`

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

Example.

```
vgchange --addtag @db1 vg1
```

`--deltag <tag_info>`

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

Example.

```
vgchange --deltag @db1 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
```

4.7.4 Configuration File Syntax

- [Section 4.7.4.1, “Enabling Hostname Tags in the lvm.conf File”](#)
- [Section 4.7.4.2, “Defining Tags for Hostnames in the lvm.conf File”](#)
- [Section 4.7.4.3, “Defining Activation”](#)
- [Section 4.7.4.4, “Defining Activation in Multiple Hostname Configuration Files”](#)

4.7.4.1 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 4.7.4.3, “Defining Activation”](#).

4.7.4.2 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" ]
    }
}
```

4.7.4.3 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.

4.7.4.4 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.

4.7.5 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
```

4.7.6 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 4.7.6.1, "Option 1: Centralized Admin and Static Configuration Replicated Between Hosts"*
- *Section 4.7.6.2, "Option 2: Localized Admin and Configuration"*

4.7.6.1 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

```

4.7.6.2 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 @filesver vg2
```

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

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

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

- b. 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 = [ "@filesver" ]  
}
```

```
}
```

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

- a. 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" ]  
}
```

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

```
vgchange -ay vg2  
  
vgchange -ay @fileserver
```

4.8 Resizing a Volume Group

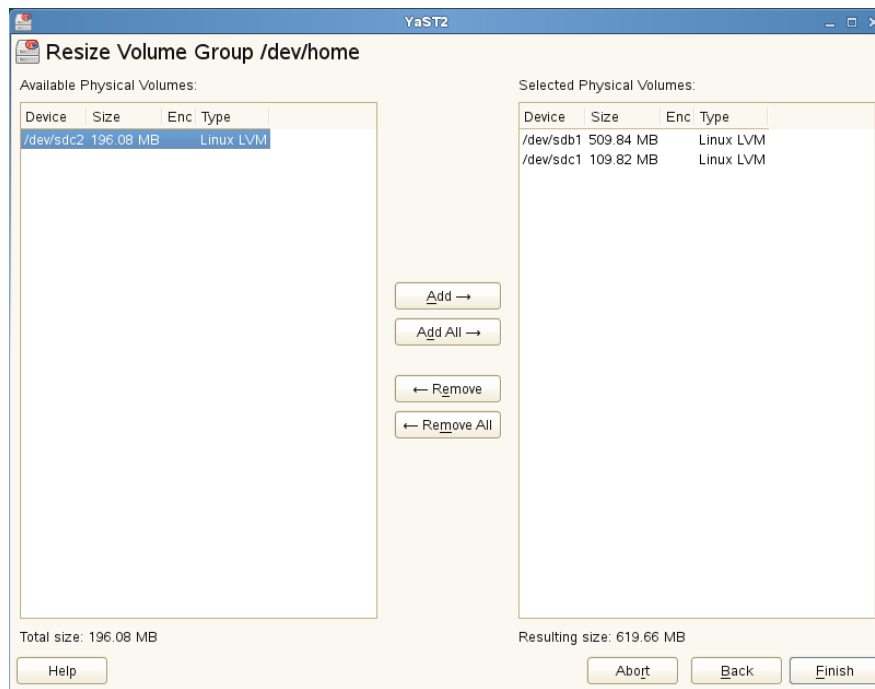
You can add and remove Linux LVM partitions from a volume group to expand or reduce its size.



Warning

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

1. Launch YaST as the `root` user.
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.

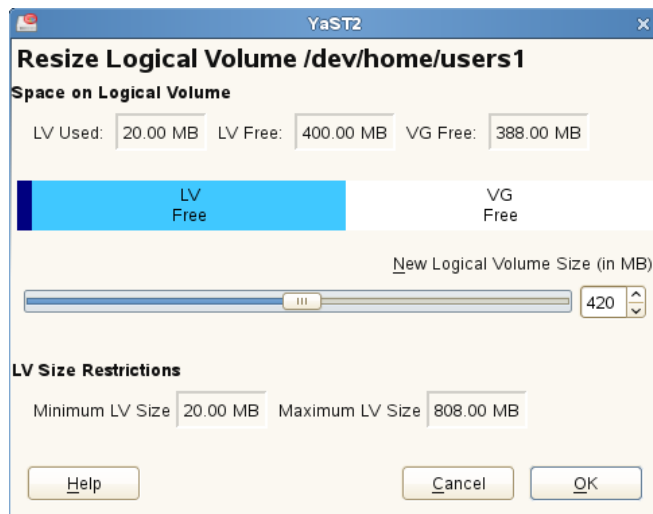
7. Click *Finish*.

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

4.9 Resizing a Logical Volume with YaST

1. Launch YaST as the root user.
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 box.



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*.

4.10 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](#).

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
```

4.11 Deleting a Volume Group



Warning

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

1. Launch YaST as the root user.
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*.

4.12 Deleting an LVM Partition (Physical Volume)



Warning

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


1. Launch YaST as the root user.
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*.


4.13 Using LVM Commands

For information about using LVM commands, see the man pages for the commands described in [Table 4.1, “LVM Commands”](#). Perform the commands as the `root` user.

TABLE 4.1: LVM COMMANDS

Command	Description
<code>pvcreate <device></code>	Initializes a device (such as <code>/dev/sdb</code>) for use by LVM as a physical volume.
<code>pvdisplay <device></code>	Displays information about the LVM physical volume, such as whether it is currently being used in a logical volume.
<code>vgcreate -c y <vg_name> <dev1> [dev2...]</code>	Creates a clustered volume group with one or more specified devices.
<code>vgchange -a [ey n] <vg_name></code>	Activates (<code>-a ey</code>) or deactivates (<code>-a n</code>) a volume group and its logical volumes for input/output. <div> Important Ensure that you use the <code>ey</code> option to exclusively activate a volume group on a cluster node. This option is used by default in the load script.</div>
<code>vgremove <vg_name></code>	Removes a volume group. Before using this command, remove the logical volumes, then deactivate the volume group.

Command	Description
<code>vgdisplay <vg_name></code>	<p>Displays information about a specified volume group.</p> <p>To find the total physical extent of a volume group, enter</p> <pre>vgdisplay vg_name grep "Total PE"</pre>
<code>lvcreate -L size -n <lv_name> <vg_name></code>	Creates a logical volume of the specified size.
<pre>lvcreate -L <size> <-T --thinpool> <vg_name/thin_pool_name> lvcreate -virtualsize <size> <-T --thin> <vg_name/thin_pool_name> -n <thin_lv_name> lvcreate -L <size> -T <vg_name/thin_pool_name> -virtualsize <size> -n <thin_lv_name></pre>	<p>Creates a thin logical volume or a thin pool of the specified size.</p> <p>Creates thin pool or thin logical volume or both. Specifying the optional argument <code>--size</code> will cause the creation of the thin pool logical volume. Specifying the optional argument <code>--virtualsize</code> will cause the creation of the thin logical volume from given thin pool volume. Specifying both arguments will cause the creation of both thin pool and thin volume using this pool.</p>
<code>lvcreate -s [-L size] -n <snap_volume> <source_volume_path></code>	Creates a snapshot volume for the specified logical volume. If the size option (<code>-L</code> , <code>--size</code>) is not included, the snapshot is created as a thin snapshot.
<code>lvremove</dev/vg_name/lv_name></code>	Removes a logical volume, such as <u>/dev/vg_name/lv_name</u> .

Command	Description
	Before using this command, close the logical volume by dismounting it with the <u>umount</u> command.
<code>lvremove snap_volume_path</code>	Removes a snapshot volume.
<code>lvconvert --merge [-b] [-i <seconds>] [<snap_volume_path>[...<snapN>] @<volume_tag>]</code>	Reverts (rolls back or merges) the snapshot data into the original volume.
<code>vgextend <vg_name><device></code>	Adds a specified physical volume to an existing volume group
<code>vgreduce <vg_name> <device></code>	Removes a specified physical volume from an existing volume group. <div>  Important Ensure that the physical volume is not currently being used by a logical volume. If it is, you must move the data to another physical volume by using the <u>pvmove</u> command. </div>
<code>lvextend -L size</dev/vg_name/lv_name></code>	Extends the size of a specified logical volume. Afterwards, you must also expand the file system to take advantage of the newly available space.
<code>lvreduce -L size </dev/vg_name/lv_name></code>	Reduces the size of a specified logical volume.

Command	Description
	<div data-bbox="798 241 861 304"></div> <div data-bbox="877 241 1082 291">Important</div> <p data-bbox="877 309 1401 488">Ensure that you reduce the size of the file system first before shrinking the volume, otherwise you risk losing data.</p>
<pre data-bbox="188 600 778 689">lvrename </dev/vg_name/old_lv_name> </dev/vg_name/new_lv_name></pre>	<p data-bbox="798 555 1401 734">Renames an existing LVM logical volume in a volume group from the old volume name to the new volume name. It does not change the volume group name.</p>

5 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”*
- *Section 5.2, “Increasing the Size of an Ext2, Ext3, or Ext4 File System”*
- *Section 5.3, “Increasing the Size of a Reiser File System”*
- *Section 5.4, “Decreasing the Size of an Ext2 or Ext3 File System”*
- *Section 5.5, “Decreasing the Size of a Reiser File System”*

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”*
- *Section 5.1.2, “Increasing the Size of a File System”*
- *Section 5.1.3, “Decreasing the Size of a File System”*

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, 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 removable device.

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.


6 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”*
- *Section 6.2, “Understanding UUIDs”*
- *Section 6.3, “Using UUIDs in the Boot Loader and /etc/fstab File (x86)”*
- *Section 6.4, “Using UUIDs in the Boot Loader and /etc/fstab File (IA64)”*
- *Section 6.5, “Additional Information”*

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”*
- *Section 6.2.2, “Finding the UUID for a File System Device”*

6.2.1 Using UUIDs to Assemble or Activate File System Devices

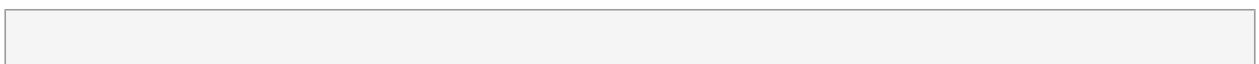
The UUID is always unique to the partition and does not depend on the order in which it appears or where it is mounted. With certain SAN devices attached to the server, the system partitions are renamed and moved to be the last device. For example, if root (/) is assigned to /dev/sda1 during the install, 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:



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 `/etc/default/grub` 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`.

- a. At the terminal console prompt, enter

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

- b. List all partitions by entering

```
ll
```

- c. Find the UUID, such as

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

5. Edit the boot loader configuration using the *Boot Loader* option in YaST.
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/grub2/grub.cfg file as a guide.

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:

- Launch YaST as the root user, select *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`.

- a. At the terminal console prompt, enter

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

- b. List all partitions by entering

```
ll
```

- c. Find the UUID, such as

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

5. Edit the boot loader file, using the Boot Loader option in YaST.

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 ##### 16 #####
`udev #####`

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

```
man 8 udev
```

7 Managing Multipath I/O for Devices

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 Multipath I/O”*
- *Section 7.2, “Planning for Multipathing”*
- *Section 7.3, “Multipath Management Tools”*
- *Section 7.4, “Configuring the System for Multipathing”*
- *Section 7.5, “Enabling and Starting Multipath I/O Services”*
- *Section 7.6, “Creating or Modifying the /etc/multipath.conf File”*
- *Section 7.7, “Configuring Default Policies for Polling, Queueing, and Failback”*
- *Section 7.8, “Blacklisting Non-Multipath Devices”*
- *Section 7.9, “Configuring User-Friendly Names or Alias Names”*
- *Section 7.10, “Configuring Default Settings for zSeries Devices”*
- *Section 7.11, “Configuring Path Failover Policies and Priorities”*
- *Section 7.12, “Configuring Multipath I/O for the Root Device”*
- *Section 7.13, “Configuring Multipath I/O for an Existing Software RAID”*
- *Section 7.14, “Scanning for New Devices without Rebooting”*
- *Section 7.15, “Scanning for New Partitioned Devices without Rebooting”*
- *Section 7.16, “Viewing Multipath I/O Status”*
- *Section 7.17, “Managing I/O in Error Situations”*
- *Section 7.18, “Resolving Stalled I/O”*
- *Section 7.19, “Troubleshooting MPIO”*
- *Section 7.20, “What’s Next”*

7.1 Understanding Multipath I/O

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.

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"*
- *Section 7.2.2, "PRIO Settings in multipath-tools-0.4.9"*
- *Section 7.2.3, "Using WWID, User-Friendly, and Alias Names for Multipath Devices"*
- *Section 7.2.4, "Using LVM2 on Multipath Devices"*
- *Section 7.2.5, "Using mdadm with Multipath Devices"*
- *Section 7.2.6, "Using Multipath with NetApp Devices"*
- *Section 7.2.7, "Using --noflush with Multipath Devices"*
- *Section 7.2.8, "SAN Timeout Settings When the Root Device Is Multipathed"*
- *Section 7.2.9, "Partitioning Multipath Devices"*
- *Section 7.2.10, "Supported Architectures for Multipath I/O"*
- *Section 7.2.11, "Supported Storage Arrays for Multipathing"*

7.2.1 Guidelines for Multipathing

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

- [Section 7.2.1.1, “Prerequisites”](#)
- [Section 7.2.1.2, “Vendor-Provided Multipath Solutions”](#)
- [Section 7.2.1.3, “Disk Management Tasks”](#)
- [Section 7.2.1.4, “Software RAIDs”](#)
- [Section 7.2.1.5, “High-Availability Solutions”](#)
- [Section 7.2.1.6, “Volume Managers”](#)
- [Section 7.2.1.7, “Virtualization Environments”](#)

7.2.1.1 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 7.2.11, “Supported Storage Arrays for Multipathing”](#).
- 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.

7.2.1.2 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.

7.2.1.3 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 re-initialized, 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.

7.2.1.4 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.

For information about setting up multipathing for existing software RAIDs, see [Section 7.13, “Configuring Multipath I/O for an Existing Software RAID”](#).

7.2.1.5 High-Availability Solutions

High-availability solutions for clustering storage resources run on top of the multipathing service on each node. Ensure that the configuration settings in the /etc/multipath.conf file on each node are consistent across the cluster.

Ensure that multipath devices the same name across all devices by doing the following:

- Use UUID and alias names to ensure that multipath device names are consistent across all nodes in the cluster. Alias names must be unique across all nodes. Copy the /etc/multipath.conf file from the node to the /etc/ directory all of the other nodes in the cluster.
- When using links to multipath-mapped devices, ensure that you specify the dm-uuid* name or alias name in the /dev/disk/by-id directory, and not a fixed path instance of the device. For information, see *Section 7.2.3, “Using WWID, User-Friendly, and Alias Names for Multipathed Devices”*.
- Set the user_friendly_names configuration option to no to disable it. A user-friendly name is unique to a node, but a device might not be assigned the same user-friendly name on every node in the cluster.

You can force the system-defined user-friendly names to be consistent across all nodes in the cluster by doing the following:

1. In the /etc/multipath.conf file on one node:
 - a. Set the user_friendly_names configuration option to yes to enable it. Multipath uses the /var/lib/multipath/bindings file to assign a persistent and unique name to the device in the form of mpath<n> in the /dev/mapper directory.
 - b. (Optional) Set the bindings_file option in the defaults section of the /etc/multipath.conf file to specify an alternate location for the bindings file.
The default location is /var/lib/multipath/bindings.
2. Set up all of the multipath devices on the node.
3. Copy the /etc/multipath.conf file from the node to the /etc/ directory all of the other nodes in the cluster.
4. Copy the bindings file from the node to the bindings_file path on all of the other nodes in the cluster.

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.

7.2.1.6 Volume Managers

Volume managers such as LVM2 and Clustered LVM2 run on top of multipathing. You must configure multipathing for a device before you use LVM2 or cLVM2 to create segment managers and file systems on it. For information, see [Section 7.2.4, “Using LVM2 on Multipath Devices”](#).

7.2.1.7 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 PRIO Settings in multipath-tools-0.4.9

SLES 11 SP2 upgrades the multipath-tools from 0.4.8 to 0.4.9. Some changes in PRIO syntax require that you manually modify the `/etc/multipath.conf` file as needed to comply with the new syntax.

The syntax for the `prio` keyword in the `/etc/multipath.conf` file is changed in `multipath-tools-0.4.9`. The `prio` line specifies the prioritizer. If the prioritizer requires an argument, you specify the argument by using the `prio_args` keyword on a second line. Previously, the prioritizer and its arguments were included on the `prio` line.

Multipath Tools 0.4.9 and later uses the `prio` setting in the `defaults{}` or `devices{}` section of the `/etc/multipath.conf` file. It silently ignores the keyword `prio` when it is specified for an individual `multipath` definition in the `multipaths{}` section. Multipath Tools 0.4.8 for SLES 11 SP1 and earlier allows the `prio` setting in the individual `multipath` definition in the `multipaths{}` section to override the `prio` settings in the `defaults{}` or `devices{}` section.

7.2.3 Using WWID, User-Friendly, and Alias Names for Multipathed Devices

A multipath device can be uniquely identified by its WWID, by a user-friendly name, or by an alias that you assign for it. Device node names in the form of `/dev/sdn` and `/dev/dm-n` can change on reboot and might be assigned to different devices each time. A device's WWID, user-friendly name, and alias name persist across reboots, and are the preferred way to identify the device.

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. Partitioned devices have `_part<n>` appended to the device name, such as `/dev/disk/by-id/xxx_part1`.

In the `/dev/disk/by-id` directory, the multipath-mapped devices are represented by the device's `dm-uuid*` name or alias name (if you assign an alias for it in the `/etc/multipath.conf` file). The `scsi-` and `wwn-` device names represent physical paths to the devices.



Important

When using links to multipath-mapped devices, ensure that you specify the `dm-uuid*` name or alias name in the `/dev/disk/by-id` directory, and not a fixed path instance of the device.

When you define device aliases in the `/etc/multipath.conf` file, ensure that you use each device's WWID (such as `3600508e0000000009e6baa6f609e7908`) and not its WWN, which replaces the first character of a device ID with `0x`, such as `0x600508e00000000009e6baa6f609e7908`.

For information about using user-friendly names and aliases for multipathed devices, see [Section 7.9, "Configuring User-Friendly Names or Alias Names"](#).

7.2.4 Using LVM2 on Multipath Devices

Ensure that the configuration file for `lvm.conf` points to the multipath-device names instead of fixed path names. This should happen automatically if `boot.multipath` is enabled and loads before `boot.lvm`.

- [Section 7.2.4.1, “Adding a Multipath Device Filter in the /etc/lvm/lvm.conf File”](#)
- [Section 7.2.4.2, “Enabling boot.multipath”](#)
- [Section 7.2.4.3, “Troubleshooting MPIO Mapping for LVM Devices”](#)

7.2.4.1 Adding a Multipath Device Filter in the /etc/lvm/lvm.conf File

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` to scan multipathed devices through the multipath I/O layer.

Adding a multipath filter prevents LVM from scanning and using the physical paths for raw device nodes that represent individual paths to the SAN (`/dev/sd*`). Ensure that you specify the filter path so that LVM 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 filter path so that only the Device Mapper names are scanned after multipathing is configured. The following filter path accepts only partitions on a multipathed device:

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

To accept both raw disks and partitions for Device Mapper names, specify the path as follows, with no hyphen (-) before mpath:

```
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 **pvscan** 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/dracut.conf.d/01-dist.conf by adding the following line:

```
force_drivers+="dm-multipath"
```

6. Make a new initrd to ensure that the Device Mapper Multipath services are loaded with the changed settings. Running **dracut** 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:

```
dracut -f --add-drivers multipath
```

7. Reboot the server to apply the changes.

7.2.4.2 Enabling `boot.multipath`

Multipath must be loaded before LVM to ensure that multipath maps are built correctly. Loading multipath after LVM can result in incomplete device maps for a multipath device because LVM locks the device, and MPIO cannot create the maps properly.

If the system device is a local device that does not use MPIO and LVM, you can disable both `boot.multipath` and `boot.lvm`. After the server starts, you can manually start multipath before you start LVM, then run a `pvscan` command to recognize the LVM objects.

7.2.4.3 Troubleshooting MPIO Mapping for LVM Devices

Timing is important for starting the LVM process. If LVM starts before MPIO maps are done, LVM might use a fixed path for the device instead of its multipath. The device works, so you might not be aware that the device's MPIO map is incomplete until that fixed path fails. You can help prevent the problem by enabling `boot.multipath` and following the instructions in [Section 7.2.4.1, "Adding a Multipath Device Filter in the `/etc/lvm/lvm.conf` File"](#).

To troubleshoot a mapping problem, you can use `dmsetup` to check that the expected number of paths are present for each multipath device. As the `root` user, enter the following at a command prompt:

```
dmsetup ls --tree
```

In the following sample response, the first device has four paths. The second device is a local device with a single path. The third device has two paths. The distinction between active and passive paths is not reported through this tool.

```
vg910-lv00 (253:23)
└─ 360a980006465576657346d4b6c593362 (253:10)
   │ - (65:96)
   │ - (8:128)
   │ - (8:240)
   └─ (8:16)
vg00-lv08 (253:9)
└─ (8:3)
```

```
system_vg-data_lv (253:1)
└─ 36006016088d014007e0d0d2213ecdf11 (253:0)
   └─ (8:32)
      └─ (8:48)
```

An incorrect mapping typically returns too few paths and does not have a major number of 253. For example, the following shows what an incorrect mapping looks like for the third device:

```
system_vg-data_lv (8:31)
└─ (8:32)
```

7.2.5 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** file should be set as follows to ensure that only device mapper names are scanned after multipathing is configured:

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

If you are using user-friendly names or multipath aliases, specify the path as follows:

```
DEVICE /dev/disk/by-id/dm-name-.*
```

7.2.6 Using Multipath with NetApp Devices

When using multipath for NetApp devices, we recommend the following settings in the **/etc/multipath.conf** file:

- Set the default values for the following parameters globally for NetApp devices:

```
max_fds max
```

```
queue_without_daemon no
```

- Set the default values for the following parameters for NetApp devices in the hardware table:

```
dev_loss_tmo infinity
```

```
fast_io_fail_tmo 5
```

```
features "3 queue_if_no_path pg_init_retries 50"
```

7.2.7 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
```

7.2.8 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 7.6, "Creating or Modifying the /etc/multipath.conf File"](#).

7.2.9 Partitioning Multipath Devices

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

- [Section 7.2.9.1, "SUSE Linux Enterprise Server 11"](#)
- [Section 7.2.9.2, "SUSE Linux Enterprise Server 10"](#)
- [Section 7.2.9.3, "SUSE Linux Enterprise Server 9"](#)

7.2.9.1 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.

Ensure that the configuration files for `lvm.conf` and `md.conf` point to the multipath-device names. This should happen automatically if `boot.multipath` is enabled and loads before `boot.lvm` and `boot.md`. Otherwise, the LVM and MD configuration files might contain

fixed paths for multipath-devices, and you must correct those paths to use the multipath-device names. For LVM2 and cLVM information, see [Section 7.2.4, “Using LVM2 on Multipath Devices”](#). For software RAID information, see [Section 7.13, “Configuring Multipath I/O for an Existing Software RAID”](#).

7.2.9.2 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 symbolic links to the line `/dev/dm-*` 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-*` symbolic links. The main benefit is that you can call `kpartx` with the new parameters without rebooting the server.

7.2.9.3 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 re-initialized, which usually requires a reboot.

7.2.10 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).

7.2.11 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 7.2.11.1, “Storage Arrays That Are Automatically Detected for Multipathing”*
- *Section 7.2.11.2, “Tested Storage Arrays for Multipathing Support”*
- *Section 7.2.11.3, “Storage Arrays that Require Specific Hardware Handlers”*

7.2.11.1 Storage Arrays That Are Automatically Detected for Multipathing

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

3PARdata VV
 AIX NVDISK
 AIX VDASD
 APPLE Xserve RAID
 COMPELNT Compellent Vol
 COMPAQ/HP HSV101, HSV111, HSV200, HSV210, HSV300, HSV400, HSV 450
 COMPAQ/HP MSA, HSV
 COMPAQ/HP MSA VOLUME
 DataCore SANmelody
 DDN SAN DataDirector
 DEC HSG80
 DELL MD3000
 DELL MD3000i
 DELL MD32xx
 DELL MD32xxi
 DGC
 EMC Clariion
 EMC Invista
 EMC SYMMETRIX
 EUROLOGC FC2502
 FSC CentricStor
 FUJITSU ETERNUS_DX, DXL, DX400, DX8000
 HITACHI DF
 HITACHI/HP OPEN
 HP A6189A
 HP HSVX700

HP LOGICAL VOLUME

HP MSA2012fc, MSA 2212fc, MSA2012i

HP MSA2012sa, MSA2312 fc/i/sa, MSA2324 fc/i/sa, MSA2000s VOLUME

HP P2000 G3 FC|P2000G3 FC/iSCSI|P2000 G3 SAS|P2000 G3 iSCSI

IBM 1722-600

IBM 1724

IBM 1726

IBM 1742

IBM 1745, 1746

IBM 1750500

IBM 1814

IBM 1815

IBM 1818

IBM 1820N00

IBM 2105800

IBM 2105F20

IBM 2107900

IBM 2145

IBM 2810XIV

IBM 3303 NVDISK

IBM 3526

IBM 3542

IBM IPR

IBM Nseries

IBM ProFibre 4000R

IBM S/390 DASD ECKD

IBM S/390 DASD FBA

Intel Multi-Flex

LSI/ENGENIO INF-01-00

NEC DISK ARRAY

NETAPP LUN

NEXENTA COMSTAR

Pillar Axiom

PIVOT3 RAIGE VOLUME

SGI IS

SGI TP9100, TP 9300

SGI TP9400, TP9500
STK FLEXLINE 380
STK OPENstorage D280
SUN CSM200_R
SUN LCSM100_[IEFS]
SUN STK6580, STK6780
SUN StorEdge 3510, T4
SUN SUN_6180

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 7.6, “Creating or Modifying the /etc/multipath.conf File”](#).

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 7.10, “Configuring Default Settings for zSeries Devices”](#).

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 7.6, “Creating or Modifying the /etc/multipath.conf File”](#).

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 7.2.11.2, “Tested Storage Arrays for Multipathing Support”](#).
- 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 7.2.11.3, “Storage Arrays that Require Specific Hardware Handlers”](#).
- After you modify the `/etc/multipath.conf` file, you must run **`dracut -f`** to re-create the INITRD on your system, then reboot in order for the changes to take effect.

7.2.11.2 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 7.2.11.1, "Storage Arrays That Are Automatically Detected for Multipathing"*.

7.2.11.3 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 7.2.11.2, “Tested Storage Arrays for Multipathing Support”](#).

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”](#)
- [Section 7.3.2, “Multipath I/O Management Tools”](#)
- [Section 7.3.3, “Using MDADM for Multipathed Devices”](#)
- [Section 7.3.4, “Linux multipath\(8\) Command”](#)
- [Section 7.3.5, “Linux mpathpersist\(8\) Utility”](#)

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. 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 7.2.11.3, “Storage Arrays that Require Specific Hardware Handlers”](#).
- 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</p>

Features of Storage Arrays	Description
	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.
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.
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 7.6, "Creating or Modifying the `/etc/multipath.conf` File"](#).

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
multipath	Scans the system for multipathed devices and assembles them.
multipathd	Waits for maps events, then executes multipath .
devmap-name	Provides a meaningful device name to udev for device maps (devmaps).
kpartx	Maps linear devmaps to partitions on the multipathed device, which makes it possible to create multipath monitoring for partitions on the device.
mpathpersist	Manages SCSI persistent reservations on Device Mapper Multipath devices.

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.suse.com/products/server/?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.

As with 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 7.2.4, “Using LVM2 on Multipath Devices”](#).

7.3.4 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_level] [-b bindings_file] [-d] [-h|-l|-ll|-f|-F|-B|-c|-q|-r|-w|-W] [-p failover|multibus|group_by_serial|group_by_prio|group_by_node_name] [devicename]
```

Options

`-v verbosity_level`

Prints all paths and multipaths.

0

No output.

1

Prints only the created or updated multipath names. Used to feed other tools like kpartx.

2+

Print all information: Detected paths, multipaths, and device maps.

-h

Print usage text.

-d

Dry run; do not create or update device maps.

-l

Show the current multipath topology from information fetched in sysfs and the device mapper.

-ll

Show the current multipath topology from all available information (sysfs, the device mapper, path checkers, and so on).

-f

Flush a multipath device map specified as a parameter, if unused.

-F

Flush all unused multipath device maps.

-t

Print internal hardware table to stdout.

-r

Force device map reload.

-B

Treat the bindings file as read only.

-b bindings_file

Set the user_friendly_names bindings file location. The default is /etc/multipath/bindings.

-c

Check if a block device should be a path in a multipath device.

-q

Allow device tables with queue_if_no_path when multipathd is not running.

-w

Remove the WWID for the specified device from the WWIDs file.

-W

Reset the WWIDs file to only include the current multipath devices.

-p policy

Force new maps to use the specified policy. Existing maps are not modified.

failover

One path per priority group.

multibus

All paths in one priority group.

group_by_serial

One priority group per serial.

group_by_prio

One priority group per priority value. Priorities are determined by callout programs specified as a global, per-controller or per-multipath option in the configuration file.

group_by_node_name

One priority group per target node name. Target node names are fetched in /sys/class/fc_transport/target*/node_name.

device_name

Update only the device map for the specified device. Specify the name as the device path such as /dev/sdb, or in major:minor format, or the multipath map 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. The device may alternatively be a multipath map name.

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. The device may alternatively be a multipath map name.

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. The device may alternatively be a multipath map name.

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. The device may alternatively be a multipath map name.

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. The device may alternatively be a multipath map name.

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"*:

TABLE 7.3: GROUP POLICY OPTIONS FOR THE MULTIPATH -P COMMAND

Policy Option	Description
failover	(Default) 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).

Policy Option	Description
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 <u>/etc/multipath.conf</u> configuration file.
group_by_node_name	One priority group per target node name. Target node names are fetched in the <u>/sys/class/fc_transport/target*/node_name</u> location.

7.3.5 Linux mpathpersist(8) Utility

The **mpathpersist(8)** utility can be used to manage SCSI persistent reservations on Device Mapper Multipath devices.

General syntax for the **mpathpersist(8)** command:

```
mpathpersist [options] [device]
```

Use this utility with the service action reservation key (reservation_key attribute) in the /etc/multipath.conf file to set persistent reservations for SCSI devices. The attribute is not used by default. If it is not set, the **multipathd** daemon does not check for persistent reservation for newly discovered paths or reinstated paths.

```
reservation_key <reservation key>
```

You can add the attribute to the defaults section or the multipaths section. For example:

```

multipaths {
    multipath {
        wwid      XXXXXXXXXXXXXXXXX
        alias      yellow
        reservation_key  0x123abc
    }
}

```

```
}  
}
```

Set the `reservation_key` parameter for all mpath devices applicable for persistent management, then restart the `multipathd` daemon. After it is set up, you can specify the reservation key in the `mpathpersist` commands.

Options

`-h,`

`--help`

Outputs the command usage information, then exits.

`-d <device>,`

`--device=<device>`

Query or change the device.

`-H,`

`--hex`

Display the output response in hexadecimal.

`-X <tids>,`

`--transportID=<tids>`

Transport IDs can be mentioned in several forms.

`-v <level>,`

`--verbose <level>`

Specifies the verbosity level.

0

Critical and error messages.

1

Warning messages.

2

Informational messages.

3

Informational messages with trace enabled.

PR In Options

-i,

--in

Request PR In command.

-k,

--read-keys

PR In: Read Keys.

-s,

--read-status

PR In: Read Full Status.

-r,

--read-reservation

PR In: Read Reservation.

-c,

--report-capabilities

PR In: Report Capabilities.

PR Out Options

-o,

--out

Request PR Out command.

-c,

--clear

PR Out: Clear

-Z,

--param-aptpl

PR Out parameter APTPL.

-S SARK,

--param-sark=SARK

PR Out parameter Service Action Reservation Key (SARK) in hexadecimal.

-P,

--preempt

PR Out: Preempt.

-A,

--preempt-abort

PR Out: Preempt and Abort.

-T < type > ,

--prout-type=< type >

PR Out command type.

-G,

--register

PR Out: Register.

-I,

--register-ignore

PR Out: Register and Ignore.

-L,

--release

PR Out: Release

-R,

--reserve

PR Out: Reserve.

Examples

Register the Service Action Reservation Key for the /dev/mapper/mpath9 device.

```
mpathpersist --out --register --param-sark=123abc --prout-type=5 -d /dev/mapper/  
mpath9
```

Read the Service Action Reservation Key for the /dev/mapper/mpath9 device.

```
mpathpersist -i -k -d /dev/mapper/mpath9
```

Reserve the Service Action Reservation Key for the `/dev/mapper/mpath9` device.

```
mpathpersist --out --reserve --param-sark=123abc --prout-type=8 -d /dev/mapper/  
mpath9
```

Read the reservation status of the `/dev/mapper/mpath9` device.

```
mpathpersist -i -s -d /dev/mapper/mpath9
```

7.4 Configuring the System for Multipathing

- *Section 7.4.1, “Preparing SAN Devices for Multipathing”*
- *Section 7.4.2, “Partitioning Multipath Devices”*
- *Section 7.4.3, “Configuring the Device Drivers in initrd for Multipathing”*
- *Section 7.4.4, “Adding multipathd to the Boot Sequence”*

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 7.2.8, "SAN Timeout Settings When the Root Device Is Multipathed"*.

If the LUNs are not seen by the HBA driver, `lsscsi` 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 SUSE Knowledgebase (<http://www.suse.com/support/kb/>) .

7.4.2 Partitioning Multipath Devices

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

Multipath devices are device-mapper devices. Modifying device-mapper devices with command line tools (such as `parted`, `kpartx`, or `fdisk`) works, but it does not necessarily generate the `udev` events that are required to update other layers. After you partition the device-mapper device, you should check the multipath map to make sure the device-mapper devices were mapped. If they are missing, you can remap the multipath devices or reboot the server to pick up all of the new partitions in the multipath map.

The device-mapper device for a partition on a multipath device is not the same as an independent device. When you create an LVM logical volume using the whole device, you must specify a device that contains no partitions. If you specify a multipath partition as the target device for the LVM logical volume, LVM recognizes that the underlying physical device is partitioned and the create fails. If you need to subdivide a SAN device, you can carve LUNs on the SAN device and present each LUN as a separate multipath device to the server.

7.4.3 Configuring the Device Drivers in `initrd` for Multipathing

The server 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 `force_drivers` in the file `/etc/dracut.conf.d/01-dist.conf`.

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 `qla23xx`, this entry would look like:

```
force_drivers+="cciss qla23xx"
```

After changing `/etc/dracut.conf.d/01-dist.conf`, you must re-create the `initrd` on your system with the `dracut -f` command, then reboot in order for the changes to take effect.

In SUSE Linux Enterprise Server 11 SP3 and later, four SCSI hardware handlers were added in the SCSI layer that can be used with DM-Multipath:

```
scsi_dh_alua
scsi_dh_rdac
scsi_dh_hp_sw
scsi_dh_emc
```

Add the modules to the `initrd` image, then specify them in the `/etc/multipath.conf` file as hardware handler types `alua`, `rdac`, `hp_sw`, and `emc`. For example, add one of these lines for a device definition:

```
hardware_handler "1 alua"

hardware_handler "1 rdac"

hardware_handler "1 hp_sw"

hardware_handler "1 emc"
```

To include the modules in the `initrd` image:

1. Add the device handler modules to the `force_drivers` variable in `/etc/dracut.conf.d/01-dist.conf`:

```
force_drivers+="alua rdac hp_sw emc"
```

2. Create a new `initrd`:

```
dracut /boot/initrd-<flavour>-scsi-dh \  
KERNELVERSION
```

3. Update the boot configuration file (`/etc/default/grub`, `/boot/efi/SuSE/elilo.conf`) with the newly built `initrd`.
4. Restart the server.

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 7.4.4.1, "Using YaST to Add multipathd"](#)
- [Section 7.4.4.2, "Using the Command Line to Add multipathd"](#)

7.4.4.1 Using YaST to Add multipathd

1. In YaST, click *System > Services Manager*.
2. Select `multipathd`. If the service is listed as `inactive`, click *Enable/Disable* to change this.
3. Confirm your changes.
The changes do not take affect until the server is restarted.

7.4.4.2 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

```
systemctl enable multipathd.service
```

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

```
systemctl enable multipathd.service multipath.service
```

If the multipath services 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

```
systemctl start multipath.service multipathd.service
```

7.6 Creating or Modifying the `/etc/multipath.conf` File

The `/etc/multipath.conf` file does not exist unless you create it. Default multipath device settings are applied automatically when the **`multipathd`** daemon runs unless you create the multipath configuration file and personalize the settings. 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.

Whenever you create or modify the `/etc/multipath.conf` file, the changes are not automatically applied when you save the file. This allows you time to perform a dry run to verify your changes before they are committed. When you are satisfied with the revised settings, you can update the multipath maps for the running multipathd daemon to use, or the changes will be applied the next time that the multipathd daemon is restarted, such as on a system restart.

- [Section 7.6.1, “Creating the /etc/multipath.conf File”](#)
- [Section 7.6.2, “Sections in the /etc/multipath.conf File”](#)
- [Section 7.6.3, “Verifying the Multipath Setup in the /etc/multipath.conf File”](#)
- [Section 7.6.4, “Applying the /etc/multipath.conf File Changes to Update the Multipath Maps”](#)

7.6.1 Creating the /etc/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 7.2.11.2, “Tested Storage Arrays for Multipathing Support”](#)), the default entry for that device can be used; no further configuration of the `/etc/multipath.conf` file is required.
5. Save the file.

7.6.2 Sections in the `/etc/multipath.conf` File

The `/etc/multipath.conf` file is organized in the following sections. See [/usr/share/doc/packages/multipath-tools/multipath.conf.annotated](#) for a template with extensive comments for each of the attributes and their options.

defaults

General default settings for multipath I/O. These values are used if no values are given in the appropriate device or multipath sections. For information, see [Section 7.7, “Configuring Default Policies for Polling, Queueing, and Failback”](#).

blacklist

Lists the device names to discard as not multipath candidates. Devices can be identified by their device node name (`devnode`), their WWID (`wwid`), or their vendor or product strings (`device`). For information, see [Section 7.8, “Blacklisting Non-Multipath Devices”](#).

You typically ignore non-multipathed devices, such as `cciss`, `fd`, `hd`, `md`, `dm`, `sr`, `scd`, `st`, `ram`, `raw`, `loop`.

Values.

For an example, see [Section 7.8, “Blacklisting Non-Multipath Devices”](#).

blacklist_exceptions

Lists the device names of devices to be treated as multipath candidates even if they are on the blacklist. Devices can be identified by their device node name (`devnode`), their WWID (`wwid`), or their vendor or product strings (`device`). You must specify the excepted devices by using the same keyword that you used in the blacklist. For example, if you used the `devnode` keyword for devices in the blacklist, you use the `devnode` keyword to exclude some of the devices in the blacklist exceptions. It is not possible to blacklist devices by using the `devnode` keyword and to exclude some devices of them by using the `wwid` keyword.

Values.

For examples, see [Section 7.8, “Blacklisting Non-Multipath Devices”](#) and the [/usr/share/doc/packages/multipath-tools/multipath.conf.annotated](#) file.

multipaths

Specifies settings for individual multipath devices. Except for settings that do not support individual settings, these values overwrite what is specified in the `defaults` and `devices` sections of the configuration file.

devices

Specifies settings for individual storage controllers. These values overwrite values specified in the [defaults](#) section of the configuration file. If you use a storage array that is not supported by default, you can create a [devices](#) subsection to specify the default settings for it. These values can be overwritten by settings for individual multipath devices if the keyword allows it.

For information, see the following:

- [Section 7.9, “Configuring User-Friendly Names or Alias Names”](#)
- [Section 7.10, “Configuring Default Settings for zSeries Devices”](#)

7.6.3 Verifying the Multipath Setup in the `/etc/multipath.conf` File

Whenever you create or modify the `/etc/multipath.conf` file, the changes are not automatically applied when you save the file. You can perform a “dry run” of the setup to verify the multipath setup before you update the multipath maps.

At the server command prompt, enter

```
multipath -v2 -d
```

This command scans the devices, then displays what the setup would look like if you commit the changes. It is assumed that the `multipathd` daemon is already running with the old (or default) multipath settings when you modify the `/etc/multipath.conf` file and perform the dry run. If the changes are acceptable, continue with [Section 7.6.4, “Applying the `/etc/multipath.conf` File Changes to Update the Multipath Maps”](#).

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 11 SP2 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 = XI0tech
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 --whitelisted --device=/dev/%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] XI0tech,Magnitude 3D
200d0b2da28005400 1:0:0:1 sde 8:64  1  [undef][undef] XI0tech,Magnitude 3D
200d0b2da28004d00 1:0:0:0 sdd 8:48  1  [undef][undef] XI0tech,Magnitude 3D
200d0b2da28001400 0:0:0:2 sdc 8:32  1  [undef][undef] XI0tech,Magnitude 3D
200d0b2da28005400 0:0:0:1 sdb 8:16  1  [undef][undef] XI0tech,Magnitude 3D
200d0b2da28004d00 0:0:0:0 sda 8:0   1  [undef][undef] XI0tech,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 >

```

7.6.4 Applying the `/etc/multipath.conf` File Changes to Update the Multipath Maps

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 and update the multipath maps:

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 `dracut -f` to re-create the `initrd` image on your system, then reboot in order for the changes to take effect.

7.7 Configuring Default Policies for Polling, Queueing, and Failback

The goal of multipath I/O is to provide connectivity fault tolerance between the storage system and the server. The desired default behavior depends on whether the server is a standalone server or a node in a high-availability cluster.

When you configure multipath I/O for a stand-alone server, the `no_path_retry` setting protects the server operating system from receiving I/O errors as long as possible. It queues messages until a multipath failover occurs and provides a healthy connection.

When you configure multipath I/O for a node in a high-availability cluster, you want multipath to report the I/O failure in order to trigger the resource failover instead of waiting for a multipath failover to be resolved. In cluster environments, you must modify the `no_path_retry` setting

so that the cluster node receives an I/O error in relation to the cluster verification process (recommended to be 50% of the heartbeat tolerance) if the connection is lost to the storage system. In addition, you want the multipath I/O fail back to be set to manual in order to avoid a ping-pong of resources because of path failures.

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

The following are the compiled in default settings. They will be used unless you overwrite these values by creating and configuring a personalized `/etc/multipath.conf` file.

```
defaults {
    verbosity 2
# udev_dir is deprecated in SLES 11 SP3
# udev_dir          /dev
    polling_interval    5
# path_selector default value is service-time in SLES 11 SP3
# path_selector      "round-robin 0"
    path selector       "service-time 0"
    path_grouping_policy failover
# getuid_callout is deprecated in SLES 11 SP3 and replaced with uid_attribute
# getuid_callout      "/lib/udev/scsi_id --whitelisted --device=/dev/%n"
# uid_attribute is new in SLES 11 SP3
    uid_attribute        "ID_SERIAL"
    prio                 "const"
    prio_args            ""
    features              "0"
    path_checker          "directio"
    alias_prefix          "mpath"
    rr_min_io_rq         1
    max_fds               "max"
    rr_weight             "uniform"
    queue_without_daemon "yes"
    flush_on_last_del     "no"
    user_friendly_names   "no"
    fast_io_fail_tmo     5
}
```

```
bindings_file      "/etc/multipath/bindings"
wwids_file         "/etc/multipath/wwids"
log_checker_err    "always"
retain_attached_hw_handler "no"
detect_prio        "no"
failback           "manual"
no_path_retry      "fail"
}
```

For information about setting the polling, queuing, and failback policies, see the following parameters in *Section 7.11, “Configuring Path Failover Policies and Priorities”*:

- *polling_interval*
- *no_path_retry*
- *failback*

If you modify the settings in the `defaults` section, the changes are not applied until you update the multipath maps, or until the `multipathd` daemon is restarted, such as at system restart.

7.8 Blacklisting Non-Multipath Devices

The `/etc/multipath.conf` file should contain a **`blacklist`** section where all non-multipath devices are listed. You can blacklist devices by WWID (`wwid` keyword), device name (`devnode` keyword), or device type (`device` section). You can also use the `blacklist_exceptions` section to enable multipath for some devices that are blacklisted by the regular expressions used in the `blacklist` section.

You typically ignore non-multipathed devices, such as `cciss`, `fd`, `hd`, `md`, `dm`, `sr`, `scd`, `st`, `ram`, `raw`, and `loop`. For example, local IDE hard drives and USB drives do not normally have multiple paths. If you want **`multipath`** to ignore single-path devices, put them in the **`blacklist`** section.



Note: Compatibility

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

With SUSE Linux Enterprise Server 12 the glibc-provided regular expressions are used. To match an arbitrary string, you must now use `".*"` rather than just `"*"`.

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, you can use the following regular expression to blacklist only partitions from the `cciss` driver and not the entire array:

```
blacklist {
    devnode "^cciss!c[0-9]d[0-9]*[p[0-9]*]"
}
```

You can blacklist by specific device types by adding a `device` section in the blacklist, and using the `vendor` and `product` keywords.

```
blacklist {
    device {
        vendor "DELL"
        product ".*"
    }
}
```

You can use a `blacklist_exceptions` section to enable multipath for some devices that were blacklisted by the regular expressions used in the `blacklist` section. You add exceptions by WWID (`wwid` keyword), device name (`devnode` keyword), or device type (`device` section). You must specify the exceptions in the same way that you blacklisted the corresponding devices. That is, `wwid` exceptions apply to a `wwid` blacklist, `devnode` exceptions apply to a `devnode` blacklist, and device type exceptions apply to a device type blacklist.

For example, you can enable multipath for a desired device type when you have different device types from the same vendor. Blacklist all of the vendor's device types in the `blacklist` section, and then enable multipath for the desired device type by adding a `device` section in a `blacklist_exceptions` section.

```
blacklist {
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st|sda)[0-9]*"
    device {
        vendor "DELL"
        product ".*"
    }
}

blacklist_exceptions {
    device {
        vendor "DELL"
        product "MD3220i"
    }
}
```

You can also use the `blacklist_exceptions` to enable multipath only for specific devices. For example:

```
blacklist {
    wwid ".*"
}

blacklist_exceptions {
```



```
wwid "3600d023000000000000e13955cc3751234"
wwid "3600d023000000000000e13955cc3751235"
}
```

After you modify the `/etc/multipath.conf` file, you must run `dracut -f` to re-create the `initrd` on your system, then restart the server 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.

7.9 Configuring User-Friendly Names or Alias Names

A multipath device can be identified by its WWID, by a user-friendly name, or by an alias that you assign for it. Before you begin, review the requirements in [Section 7.2.3, “Using WWID, User-Friendly, and Alias Names for Multipath Devices”](#).

Important

Because device node names in the form of `/dev/sdn` and `/dev/dm-n` can change on reboot, referring to multipath devices by their WWID is preferred. You can also use a user-friendly name or alias that is mapped to the WWID in order to identify the device uniquely across reboots.

[Table 7.4, “Comparison of Multipath Device Name Types”](#) describes the types of device names that can be used for a device in the `/etc/multipath.conf` file. For an example of `multipath.conf` settings, see the `/usr/share/doc/packages/multipath-tools/multipath.conf.synthetic` file.

TABLE 7.4: COMPARISON OF MULTIPATH DEVICE NAME TYPES

Name Types	Description
WWID (default)	The serial 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

Name Types	Description
	the <code>/dev/disk/by-id</code> directory. For example, a device with the WWID of <code>3600508e0000000009e6baa6f609e7908</code> is listed as <code>/dev/disk/by-id/scsi-3600508e0000000009e6baa6f609e7908</code> .
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	<p>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.</p> <p>If you are using <code>user_friendly_names</code>, do not set the alias to <code>mpathN</code> format. This may conflict with an automatically assigned user friendly name, and give you incorrect device node names.</p>

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>` in the `/dev/mapper` directory. 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 should not use aliases for the system root device. Because the device name would differ, using an alias causes you to lose the ability to seamlessly switch off multipathing via the kernel command line.

- **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.

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.
 - a. Move the /var/lib/multipath/bindings file to /etc/multipath/bindings.
 - b. 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, not recommended) Enable user-friendly names:
 - a. Uncomment the defaults section and its ending bracket.
 - b. 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         blue1
    }
    multipath {
        wwid          36006048000028350131253594d303041
        alias         blue2
    }
    multipath {
        wwid          36006048000028350131253594d303145
        alias         yellow1
    }
    multipath {
        wwid          36006048000028350131253594d303334
        alias         yellow2
    }
}

```

6. Save your changes, then close the file.

The changes are not applied until you update the multipath maps, or until the multipathd daemon is restarted, such as at system restart.

7.10 Configuring Default Settings for zSeries Devices

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.

If you modify the `/etc/multipath.conf` file, the changes are not applied until you update the multipath maps, or until the `multipathd` daemon is restarted, such as at system restart.

7.11 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 7.11.1, "Configuring the Path Failover Policies"](#)
- [Section 7.11.2, "Configuring Failover Priorities"](#)
- [Section 7.11.3, "Using a Script to Set Path Priorities"](#)
- [Section 7.11.4, "Configuring ALUA \(`mpath_prio_alua`\)"](#)
- [Section 7.11.5, "Reporting Target Path Groups"](#)

7.11.1 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
failover	(Default) One path per priority group.
multibus	All paths in one priority group.
group_by_serial	One priority group per detected serial number.
group_by_prio	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.
group_by_node_name	One priority group per target node name. Target node names are fetched in the <code>/sys/class/fc_transport/target*/node_name</code> location.

7.11.2 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.

If you modify the `/etc/multipath.conf` file, the changes are not automatically applied when you save the file. For information, see [Section 7.6.3, “Verifying the Multipath Setup in the `/etc/multipath.conf` File”](#) and [Section 7.6.4, “Applying the `/etc/multipath.conf` File Changes to Update the Multipath Maps”](#).

- [Section 7.11.2.1, “Understanding Priority Groups and Attributes”](#)
- [Section 7.11.2.2, “Configuring for Round-Robin Load Balancing”](#)
- [Section 7.11.2.3, “Configuring for Single Path Failover”](#)
- [Section 7.11.2.4, “Grouping I/O Paths for Round-Robin Load Balancing”](#)

7.11.2.1 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.

Multipath Tools 0.4.9 for SLES 11 SP2 uses the prio setting in the defaults{} or devices{} section of the /etc/multipath.conf file. It silently ignores the keyword prio when it is specified for an individual multipath definition in the multipaths{} section. Multipath Tools 0.4.8 for SLES 11 SP1 and earlier allows the prio setting in the individual multipath definition in the multipaths{} section to override the prio settings in the defaults{} or devices{} section.

The syntax for the prio keyword in the /etc/multipath.conf file is changed in multipath-tools-0.4.9. The prio line specifies the prioritizer. If the prioritizer requires an argument, you specify the argument by using the prio_args keyword on a second line. Previously, the prioritizer and its arguments were included on the prio line.

PRIO Settings for the Defaults or Devices Sections

prio

Specifies the prioritizer program to call to obtain a path priority value. Weights are summed for each path group to determine the next path group to use in case of failure.

Use the prio_args keyword to specify arguments if the specified prioritizer requires arguments.

Values.

If no prio keyword is specified, all paths are equal. The default setting is “const” with a prio_args setting with no value.


```
prio      "const"
prio_args ""
```

Example prioritizer programs include:

Prioritizer Program	Description
alua	Generates path priorities based on the SCSI-3 ALUA settings.
const	Generates the same priority for all paths.
emc	Generates the path priority for EMC arrays.
hdc	Generates the path priority for Hitachi HDS Modular storage arrays.
hp_sw	Generates the path priority for Compaq/HP controller in active/standby mode.
ontap	Generates the path priority for NetApp arrays.
random	Generates a random priority for each path.
rdac	Generates the path priority for LSI/Engenio RDAC controller.
weightedpath	<p>Generates the path priority based on the weighted values you specify in the arguments for <code>prio_args</code>, such as:</p> <pre><hbtl devname> <regex1> <prio1> <regex2> <prio2>...</pre> <p>The <code>hbtl regex</code> argument format uses the SCSI <code>H:B:T:L</code> notation (such as <code>1:0:..</code> and <code>*:0:0:.</code>) with a weight value, where H, B, T, L are the host, bus, target, and LUN IDs for a device. For example:</p> <pre>prio "weightedpath" prio_args "hbtl 1:..... 2 4:..... 4"</pre>

Prioritizer Program	Description
	<p>The devname regex argument format uses a device node name with a weight value for each device. For example:</p> <pre>prio "weightedpath" prio_args "devname sda 50 sde 10 sdc 50 sdf 10"</pre>

prio_args

Specifies the arguments for the specified prioritizer program that requires arguments. Most prio programs do not need arguments.

Values.

There is no default. The value depends on the prio setting and whether the prioritizer requires arguments.

```
prio      "const"
prio_args ""
```

Multipath Attributes

Multipath attributes are used to control the behavior of multipath I/O for devices. You can specify attributes as defaults for all multipath devices. You can also specify attributes that apply only to a given multipath device by creating an entry for that device in the multipaths section of the multipath configuration file.

user_friendly_names

Specifies whether to use world-wide IDs (WWIDs) or to use the /var/lib/multipath/bindings file to assign a persistent and unique alias to the multipath devices in the form of /dev/mapper/mpathN.

This option can be used in the devices section and the multipaths section.

Values.

Value	Description
no	(Default) Use the WWIDs shown in the <code>/dev/disk/by-id/</code> location.
yes	Autogenerate user-friendly names as aliases for the multipath devices instead of the actual ID.

failback

Specifies whether to monitor the failed path recovery, and indicates the timing for group failback after failed paths return to service.

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.

Values.

Value	Description
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.
immediate	When a path recovers, enable the path immediately.
n	When the path recovers, wait <code>n</code> seconds before enabling the path. Specify an integer value greater than 0.

We recommend failback setting of “manual” for multipath in cluster environments in order to prevent multipath failover ping-pong.

```
failback "manual"
```



Important

Ensure that you verify the failback setting with your storage system vendor. Different storage systems can require different settings.

getuid_callout

The default program and arguments to call to obtain a unique path identifier. Specify the location with an absolute Linux path.

This attribute is deprecated in SLES 11 SP3. It is replaced by the [uid_attribute](#).

Values.

The default location and arguments are:

```
/lib/udev/scsi_id -g -u -s
```

Example:

```
getuid_callout "/lib/udev/scsi_id -g -u -d /dev/%n"
```

```
getuid_callout "/lib/udev/scsi_id --whitelisted --device=/dev/%n"
```

no_path_retry

Specifies the behaviors to use on path failure.

Values.

Value	Description
n	<p>Specifies the number of retries until multipath stops the queuing and fails the path. Specify an integer value greater than 0.</p> <p>In a cluster, you can specify a value of “0” to prevent queuing and allow resources to fail over.</p>
fail	Specifies immediate failure (no queuing).
queue	Never stop queuing (queue forever until the path comes alive).

We recommend a retry setting of “fail” or “0” in the `/etc/multipath.conf` 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.

```
no_path_retry "fail"
no_path_retry "0"
```

! Important

Ensure that you verify the retry settings with your storage system vendor. Different storage systems can require different settings.

path_checker

Determines the state of the path.

Values.

Value	Description
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. Logs failure messages in the <code>systemd</code> journal (see <code>##### 11 ##journalctl### systemd ###</code>).
readsector0	(Default in <code>multipath-tools</code> version 0.4.7 and earlier; deprecated and replaced by <code>directio</code> .) Reads the first sector of the device. Logs failure messages in the <code>systemd</code> journal (see <code>##### 11 ##journalctl### systemd ###</code>).
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 the <code>systemd</code> log journal with messages.
<code>custom_vendor_value</code>	Some SAN vendors provide custom <code>path_checker</code> options: <ul style="list-style-type: none">• <code>cciss_tur</code>: Checks the path state for HP Smart Storage Arrays.• <code>emc_clariion</code>: Queries the EMC Clariion EVPD page 0xC0 to determine the path state.

Value	Description
	<ul style="list-style-type: none"> • 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.

path_grouping_policy

Specifies the path grouping policy for a multipath device hosted by a given controller.

Values.

Value	Description
failover	(Default) One path is assigned per priority group so that only one path at a time is used.
multibus	All valid paths are in one priority group. Traffic is load-balanced across all active paths in the group.
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.
group_by_serial	Paths are grouped by the SCSI target serial number (controller node WWN).
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> .

path_selector

Specifies the path-selector algorithm to use for load balancing.

Values.

Value	Description
round-robin 0	(Default in SLES 11 SP2 and earlier) The load-balancing algorithm used to balance traffic across all active paths in a priority group.
queue-length 0	A dynamic load balancer that balances the number of in-flight I/O on paths similar to the least-pending option.
service-time 0	(Default in SLES 11 SP3 and later) A service-time oriented load balancer that balances I/O on paths according to the latency.

pg_timeout

Specifies path group timeout handling.

Values.

NONE (internal default)

polling_interval

Specifies the time in seconds between the end of one path checking cycle and the beginning of the next path checking cycle.

Values.

Specify an integer value greater than 0. The default value is 5. Ensure that you verify the `polling_interval` setting with your storage system vendor. Different storage systems can require different settings.

prio_callout

Specifies the program and arguments to use to determine the layout of the multipath map. Multipath `prio_callout` programs are located in shared libraries in `/lib/libmultipath/lib*`. By using shared libraries, the callout programs are loaded into memory on daemon startup.

When queried by the `multipath` command, the specified `mpath_prio_*` callout program returns the priority for a given path in relation to the entire multipath layout.

When it is used with the `path_grouping_policy` of `group_by_prio`, all paths with the same priority are grouped into one multipath group. The group with the highest aggregate priority becomes the active group.

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.

The `mpath_prio_*` program can also be a custom script created by a vendor or administrator for a specified setup.

- A `%n` in the command line expands to the device name in the `/dev` directory.
- A `%b` in the command line expands to the device number in `major:minor` format in the `/dev` directory.
- A `%d` in the command line expands to the device ID in the `/dev/disk/by-id` directory.

If devices are hot-pluggable, use the `%d` flag instead of `%n`. This addresses the short time that elapses between the time when devices are available and when `udev` creates the device nodes.

Values.

Value	Description
(No value)	If no <code>prio_callout</code> attribute is used, all paths are equal. This is the default.
<code>/bin/true</code>	Specify this value when the <code>group_by_prio</code> is not being used.

The **prioritizer** programs generate path priorities when queried by the **multipath** command. The program names must begin with `mpath_prio_` and are named by the device type or balancing method used. Current prioritizer programs include the following:

Prioritizer Program	Description
<code>mpath_prio_alua %n</code>	Generates path priorities based on the SCSI-3 ALUA settings.
<code>mpath_prio_balance_units</code>	Generates the same priority for all paths.
<code>mpath_prio_emc %n</code>	Generates the path priority for EMC arrays.
<code>mpath_prio_hds_modular %b</code>	Generates the path priority for Hitachi HDS Modular storage arrays.

Prioritizer Program	Description
<code>mpath_prio_hp_sw %n</code>	Generates the path priority for Compaq/HP controller in active/standby mode.
<code>mpath_prio_netapp %n</code>	Generates the path priority for NetApp arrays.
<code>mpath_prio_random %n</code>	Generates a random priority for each path.
<code>mpath_prio_rdac %n</code>	Generates the path priority for LSI/Engenio RDAC controller.
<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.
<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 7.11.3, "Using a Script to Set Path Priorities" .

rr_min_io

Specifies the number of I/O transactions to route to a path before switching to the next path in the same path group, as determined by the specified algorithm in the `path_selector` setting.

The `rr_min_io` attribute is used only for kernels 2.6.31 and earlier. It is obsoleted in SLES 11 SP2 and replaced by the `rr_min_io_rq` attribute.

Values.

Specify an integer value greater than 0. The default value is 1000.

```
rr_min_io "1000"
```

rr_min_io_rq

Specifies the number of I/O requests to route to a path before switching to the next path in the current path group, using request-based device-mapper-multipath.

This attribute is available for systems running SLES 11 SP2 and later. It replaces the rr_min_io attribute.

Values.

Specify an integer value greater than 0. The default value is 1.

```
rr_min_io_rq "1"
```

rr_weight

Specifies the weighting method to use for paths.

Values.

Value	Description
uniform	(Default) All paths have the same round-robin weights.
priorities	Each path's weight is determined by the path's priority times the rr_min_io_rq setting (or the rr_min_io setting for kernels 2.6.31 and earlier).

uid_attribute

A udev attribute that provides a unique path identifier. The default value is ID_SERIAL.

7.11.2.2 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.

7.11.2.3 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.

7.11.2.4 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.

7.11.3 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 3600a0b8000fd6320000000045317563
2
```

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

```
0x10000000c95eb4 0x200200a0b8122c6e 2:0:0:3 sde 3600a0b8000fd6320000000245317593
2
```

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

```
0x10000000c95eb4 0x200300a0b8122c6e 2:0:1:1 sdj 3600a0b8000fd6320000000045317563
51
```

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

```
0x10000000c95eb4 0x200300a0b8122c6e 2:0:1:3 sdl 3600a0b8000fd6320000000245317593
51
```

To continue the example mentioned in [prio_callout](#), 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.

7.11.4 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 7.11.4.1, "Syntax"](#)
- [Section 7.11.4.2, "Prerequisite"](#)

- [Section 7.11.4.3, “Options”](#)
- [Section 7.11.4.4, “Return Values”](#)

7.11.4.1 Syntax

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

7.11.4.2 Prerequisite

SCSI devices.

7.11.4.3 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>.

7.11.4.4 Return Values

On success, returns a value of 0 and the priority value for the group. *Table 7.6, “ALUA Priorities for Device Mapper Multipath”* shows the priority values returned by the **mpath_prio_alua** command.

TABLE 7.6: 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**.

7.11.5 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)` .

7.12 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. In addition, the YaST partitioner in the YaST installer supports enabling multipath during the install.

- *Section 7.12.1, “Enabling Multipath I/O at Install Time”*
- *Section 7.12.2, “Enabling Multipath I/O for an Existing Root Device”*
- *Section 7.12.3, “Disabling Multipath I/O on the Root Device”*

7.12.1 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 7.12.1.1, “Enabling Multipath I/O at Install Time on an Active/Active Multipath Storage LUN”*
- *Section 7.12.1.2, “Enabling Multipath I/O at Install Time on an Active/Passive Multipath Storage LUN”*

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

1. During the install on the YaST 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.

YaST starts to rescan the disks and shows available multipath devices (such as `/dev/disk/by-id/dm-uuid-mpath-3600a0b80000f4593000012ae4ab0ae65`). This is the device that should be used for all further processing.

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

7.12.1.2 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 YaST 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.

YaST starts to rescan the disks and shows available multipath devices (such as `/dev/disk/by-id/dm-uuid-mpath-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, YaST 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.

- a. Mount the root device to `/mnt` by entering


```
mount /dev/disk/by-id/<UUID>_part2 /mnt
```

For example, enter

```
mount /dev/disk/by-id/dm-uuid-  
mpath-3600a0b80000f4593000012ae4ab0ae65_part2 /mnt
```

- b. Mount the boot device to /mnt/boot by entering

```
mount /dev/disk/by-id/<UUID>_part1 /mnt/boot
```

For example, enter

```
mount /dev/disk/by-id/dm-uuid-  
mpath-3600a0b80000f4593000012ae4ab0ae65_part2 /mnt/boot
```

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

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

- d. 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](#) and continue with [Step 9](#).
- **Passive path:** The configuration must be changed and the boot loader must be reinstalled. Continue with [Step 8](#).

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

- a. 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
```

- b. At the console, run **`multipath -ll`**, then check the output to find the active path. Passive paths are flagged as **`ghost`**.
- c. In the `/mnt/boot/grub/device.map` file, change the `hd0` entry to an active path, save the changes, and close the file.
- d. 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
```

- e. Reinstall the boot loader by entering

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

- f. 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.

7.12.2 Enabling Multipath I/O for an Existing Root Device

1. Install Linux with only a single path active, preferably one where the by-id symbolic links are listed in the partitioner.
2. Mount the devices by using the /dev/disk/by-id path used during the install.
3. Add dm-multipath to /etc/dracut.conf.d/01-dist.conf by adding the following line:

```
force_drivers+="dm-multipath"
```

4. For System Z, before running **dracut**, 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 /etc/fstab.
5. Run **dracut -f** to update the initrd image.
6. For System Z, after running **dracut**, run **zipl**.
7. Reboot the server.

7.12.3 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.

7.13 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. It assumes you have enabled user-friendly-names in the `/etc/multipath.conf` file as described in [Section 7.9, “Configuring User-Friendly Names or Alias Names”](#).

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 `md` service by entering

```
systemctl stop md.service
```

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

```
systemctl start multipath.service multipathd.service
```

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 2011-01-06 11:42 dm-uuid-  
mpath-36006016088d014007e0d0d2213ecdf11 -> ../../dm-1
```

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

```
systemctl start md.service
```

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 `dracut` 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

```
dracut -f --add-drivers 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

7.14 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.



Warning

In EMC PowerPath environments, do not use the `rescan-scsi-bus.sh` utility provided with the operating system or the HBA vendor scripts for scanning the SCSI buses. To avoid potential file system corruption, EMC requires that you follow the procedure provided in the vendor documentation for EMC PowerPath for Linux.

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

Option	Description
<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. <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 `systemd journal` (see `##### 11 ##journalctl### systemd ###`). At a terminal console prompt, enter

```
journalctl -r
```

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

```
# journalctl -r
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](#) through [Step 3](#) 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.

7.15 Scanning for New Partitioned Devices without Rebooting

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



Warning

In EMC PowerPath environments, do not use the `rescan-scsi-bus.sh` utility provided with the operating system or the HBA vendor scripts for scanning the SCSI buses. To avoid potential file system corruption, EMC requires that you follow the procedure provided in the vendor documentation for EMC PowerPath for Linux.

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 7.14](#), *"Scanning for New Devices without Rebooting"*.

3. Verify that the device is seen (such as if the link has a new time stamp) by entering

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

You can also verify the devices in /dev/disk/by-id by entering

```
ls -l /dev/disk/by-id/
```

4. Verify the new device appears in the log by entering

```
journalctl -r
```

5. Use a text editor to add a new alias definition for the device in the /etc/multipath.conf file, such as data_vol3.

For example, if the UUID is 36006016088d014006e98a7a94a85db11, make the following changes:

```
defaults {
    user_friendly_names    yes
}
multipaths {
    multipath {
        wwid      36006016088d014006e98a7a94a85db11
```

```
alias data_vol3
}
}
```

6. Create a partition table for the device by entering

```
fdisk /dev/disk/by-id/dm-uuid-mpath-<UUID>
```

Replace UUID with the device WWID, such as 36006016088d014006e98a7a94a85db11.

7. Trigger udev by entering

```
echo 'add' > /sys/block/<dm_device>/uevent
```

For example, to generate the device-mapper devices for the partitions on dm-8, enter

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

8. Create a file system and label for the new partition by entering the following commands:

```
mke2fs -j /dev/disk/by-id/dm-uuid-mpath-<UUID_partN>
tune2fs -L data_vol3 /dev/disk/by-id/dm-uuid-<UUID_partN>
```

Replace UUID_part1 with the actual UUID and partition number, such as 36006016088d014006e98a7a94a85db11_part1.

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

```
sudo systemctl restart multipathd.service
```

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](#) 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=data_vol3
```

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

```
md /data_vol3
```

```
mount /data_vol3
```

7.16 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

7.17 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 **dracut -f** 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.

7.18 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
```

7.19 Troubleshooting MPIO

This section describes some known issues and possible solutions for MPIO.

- *Section 7.19.1, "PRIO Settings for Individual Devices Fail After Upgrading to Multipath 0.4.9"*
- *Section 7.19.2, "PRIO Settings with Arguments Fail After Upgrading to multipath-tools-0.4.9"*
- *Section 7.19.3, "Technical Information Documents"*

7.19.1 PRIO Settings for Individual Devices Fail After Upgrading to Multipath 0.4.9

Multipath Tools 0.4.9 for SLES 11 SP2 uses the `prio` setting in the `defaults{}` or `devices{}` section of the `/etc/multipath.conf` file. It silently ignores the keyword `prio` when it is specified for an individual `multipath` definition in the `multipaths{}` section.

Multipath Tools 0.4.8 for SLES 11 SP1 and earlier allows the `prio` setting in the individual `multipath` definition in the `multipaths{}` section to override the `prio` settings in the `defaults{}` or `devices{}` section.

7.19.2 PRIO Settings with Arguments Fail After Upgrading to multipath-tools-0.4.9

When you upgrade from `multipath-tools-0.4.8` to `multipath-tools-0.4.9`, the `prio` settings in the `/etc/multipath.conf` file are broken for prioritizers that require an argument. In `multipath-tools-0.4.9`, the `prio` keyword is used to specify the prioritizer, and the `prio_args` keyword is used to specify the argument for prioritizers that require an argument. Previously, both the prioritizer and its argument were specified on the same `prio` line.

For example, in `multipath-tools-0.4.8`, the following line was used to specify a prioritizer and its arguments on the same line.

```
prio "weightedpath hbt1 [1,3]:...+:.+ 260 [0,2]:...+:.+ 20"
```

After upgrading to `multipath-tools-0.4.9`, the command causes an error. The message is similar to the following:

```
<Month day hh:mm:ss> | Prioritizer 'weightedpath hbt1 [1,3]:...+:.+ 260  
[0,2]:...+:.+ 20' not found in /lib64/multipath
```

To resolve this problem, use a text editor to modify the `prio` line in the `/etc/multipath.conf` file. Create two lines with the prioritizer specified on the `prio` line, and the prioritizer argument specified on the `prio_args` line below it:

```
prio "weightedpath"  
prio_args "hbt1 [1,3]:...+:.+ 260 [0,2]:...+:.+ 20"
```

7.19.3 Technical Information Documents

For information about troubleshooting multipath I/O issues on SUSE Linux Enterprise Server, see the following Technical Information Documents (TIDs) in the SUSE Knowledgebase:

- *Troubleshooting SLES Multipathing (MPIO) Problems (TID 3231766)* (<http://www.suse.com/support/kb/doc.php?id=3231766>) ↗
- *DM MPIO Device Blacklisting Not Honored in multipath.conf (TID 3029706)* (<http://www.suse.com/support/kb/doc.php?id=3029706>) ↗
- *Troubleshooting SCSI (LUN) Scanning Issues (TID 3955167)* (<http://www.suse.com/support/kb/doc.php?id=3955167>) ↗
- *Using LVM on Multipath (DM MPIO) Devices* (<http://www.suse.com/support/kb/doc.php?id=7007498>) ↗

7.20 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*
- *Chapter 10, Managing Software RAIDs 6 and 10 with mdadm*

8 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”](#)
- [Section 8.2, “Soft RAID Configuration with YaST”](#)
- [Section 8.3, “Troubleshooting Software RAIDs”](#)
- [Section 8.4, “For More Information”](#)

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”](#)
- [Section 8.1.2, “RAID 1”](#)

- [Section 8.1.3, “RAID 2 and RAID 3”](#)
- [Section 8.1.4, “RAID 4”](#)
- [Section 8.1.5, “RAID 5”](#)
- [Section 8.1.6, “Nested RAID Levels”](#)

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 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.

On the next page of the wizard, choose among RAID levels 0, 1, and 5, then click *Next*. The following dialog (see [Figure 8.1, “RAID Partitions”](#)) 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 “--”.

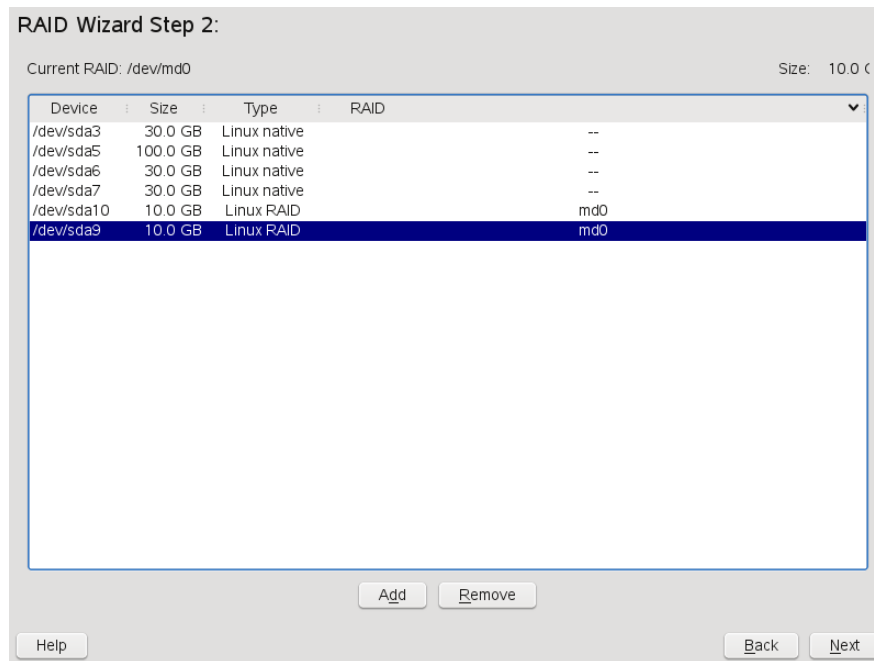


FIGURE 8.1: RAID PARTITIONS

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”](#)).

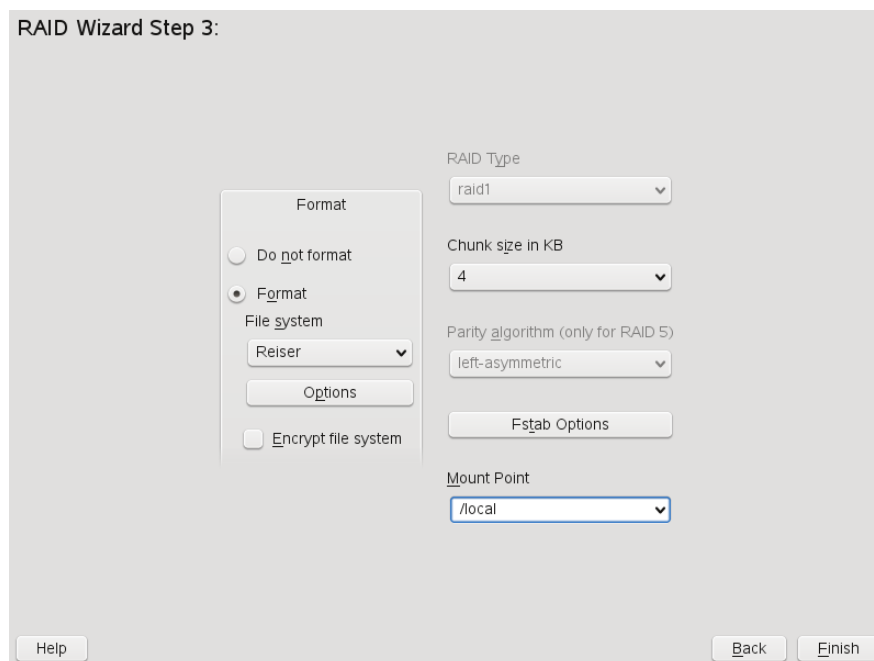


FIGURE 8.2: FILE SYSTEM SETTINGS

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.info/?l=linux-raid) (<http://marc.info/?l=linux-raid>) ↗.

9 Configuring Software RAID1 for the Root Partition

In SUSE Linux Enterprise Server, the Device Mapper RAID tool has been integrated into the YaST Partitioner. You can use the partitioner at install time to create a software RAID1 for the system device that contains your root (`/`) partition. The `/boot` partition must be created on a separate device than the MD RAID1.

- *Section 9.1, “Prerequisites for Using a Software RAID1 Device for the Root Partition”*
- *Section 9.2, “Enabling iSCSI Initiator Support at Install Time”*
- *Section 9.3, “Enabling Multipath I/O Support at Install Time”*
- *Section 9.4, “Creating a Software RAID1 Device for the Root (`/`) Partition”*

9.1 Prerequisites for Using a Software RAID1 Device for the Root Partition

Ensure that your configuration meets the following requirements:

- You need two hard drives to create the RAID1 mirror device. The hard drives should be similarly sized. The RAID assumes the size of the smaller 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.
- You need a third device to use for the `/boot` partition. The boot device should be a local device.
- 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, you must 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 RAID1 device.

1. Proceed with the YaST install of SUSE Linux Enterprise Server 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 *Configure 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 that you want to use for the root (/) partition, you must enable multipath support before you create the software RAID1 device.

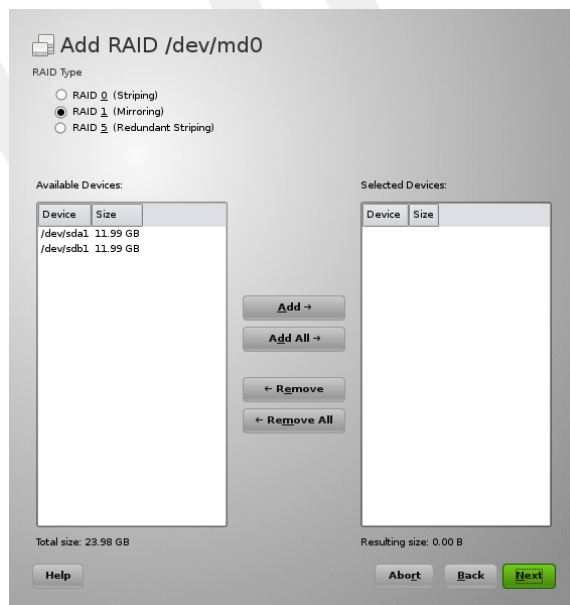
1. Proceed with the YaST install of SUSE Linux Enterprise Server until you reach the Installation Settings page.
2. Click *Partitioning* to open the Preparing Hard Disk page, click *Custom Partitioning (for experts)*, then click *Next*.
3. On the Expert Partitioner page, expand *Hard Disks* in the *System View* panel to view the default proposal.
4. On the *Hard Disks* page, select *Configure 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 RAID1 Device for the Root (/) Partition

1. Proceed with the YaST install of SUSE Linux Enterprise Server 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 */boot* partition.
 - a. On the Expert Partitioner page under *Hard Disks*, select the device you want to use for the */boot* partition, then click *Add* on the *Hard Disk Partitions* tab.
 - b. Under *New Partition Type*, select *Primary Partition*, then click *Next*.
 - c. Under *New Partition Size*, specify the size to use, then click *Next*.
 - d. Under *Format Options*, select *Format partition*, then select your preferred file system type (such as Ext2 or Ext3) from the drop-down list.
 - e. Under *Mount Options*, select *Mount partition*, then select */boot* from the drop-down list.
 - f. Click *Finish*.
5. Create a swap partition.
 - a. 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.
 - b. Under *New Partition Type*, select *Primary Partition*, then click *Next*.
 - c. Under *New Partition Size*, specify the size to use, then click *Next*.
 - d. Under *Format Options*, select *Format partition*, then select *Swap* from the drop-down list.
 - e. Under *Mount Options*, select *Mount partition*, then select *swap* from the drop-down list.

- f. Click *Finish*.
6. Set up the *0x*FD Linux RAID format for each of the devices you want to use for the software RAID1.
- a. On the Expert Partitioner page under *Hard Disks*, select the device you want to use in the RAID1, then click *Add* on the *Hard Disk Partitions* tab.
 - b. Under *New Partition Type*, select *Primary Partition*, then click *Next*.
 - c. Under *New Partition Size*, specify to use the maximum size, then click *Next*.
 - d. Under *Format Options*, select *Do not format partition*, then select *0x*FD Linux RAID from the drop-down list.
 - e. Under *Mount Options*, select *Do not mount partition*.
 - f. Click *Finish*.
 - g. Repeat *Step 6.a* to *Step 6.f* for each device that you plan to use in the software RAID1.
7. Create the RAID device.
- a. In the *System View* panel, select *RAID*, then click *Add RAID* on the RAID page. The devices that you prepared in *Step 6* are listed in *Available Devices*.

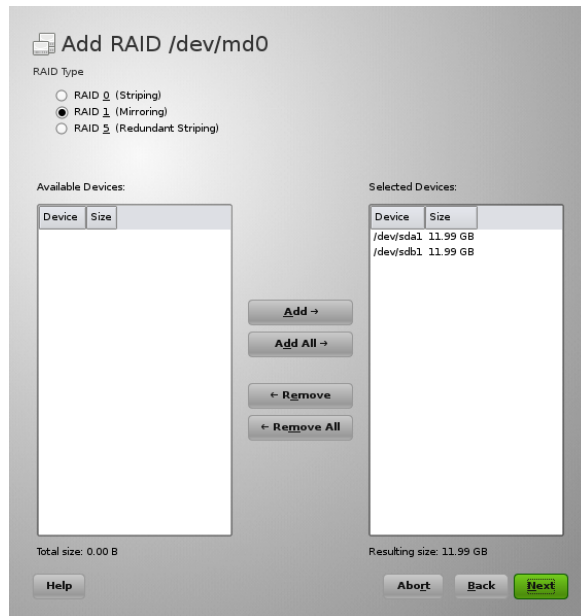


- b. Under *RAID Type*, select *RAID 1 (Mirroring)*.

- c. 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 RAID1.

To continue the example, two devices are selected for RAID1.



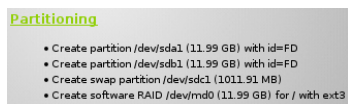
- d. Click *Next*.
- e. Under *RAID Options*, select the chunk size from the drop-down list.
The default chunk size for a RAID1 (Mirroring) is 4 KB.
Available chunk sizes are 4 KB, 8 KB, 16 KB, 32 KB, 64 KB, 128 KB, 256 KB, 512 KB, 1 MB, 2 MB, or 4 MB.
- f. Under *Formatting Options*, select *Format partition*, then select the file system type (such as Ext3) from the *File system* drop-down list.
- g. Under *Mounting Options*, select *Mount partition*, then select */* from the *Mount Point* drop-down list.
- h. Click *Finish*.
The software RAID device is managed by Device Mapper, and creates a device under the /dev/md0 path.



8. On the Expert Partitioner page, click *Accept*.

The new proposal appears under *Partitioning* on the Installation Settings page.

For example, the setup for the



9. 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.

10 Managing Software RAID 6 and 10 with mdadm

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"*
- *Section 10.2, "Creating Nested RAID 10 Devices with mdadm"*
- *Section 10.3, "Creating a Complex RAID 10"*
- *Section 10.4, "Creating a Degraded RAID Array"*

10.1 Creating a RAID 6

- *Section 10.1.1, "Understanding RAID 6"*
- *Section 10.1.2, "Creating a RAID 6"*

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

Feature	RAID 5	RAID 6
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"*
- *Section 10.2.2, "Creating Nested RAID 10 (1+0) with mdadm"*
- *Section 10.2.3, "Creating Nested RAID 10 (0+1) with mdadm"*

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 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.</p> <p>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.</p>

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](#).

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)
<u>/dev/sdb1</u> <u>/dev/sdc1</u>	<u>/dev/md0</u>	<u>/dev/md2</u>
<u>/dev/sdd1</u> <u>/dev/sde1</u>	<u>/dev/md1</u>	

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](#):

```
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](#).

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)
<u>/dev/sdb1</u> <u>/dev/sdc1</u>	<u>/dev/md0</u>	<u>/dev/md2</u>

Raw Devices	RAID 0 (stripe)	RAID 0 + 1 (mirrored stripes)
<u>/dev/sdd1</u> <u>/dev/sde1</u>	<u>/dev/md1</u>	

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*:

```
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

- *Section 10.3.1, "Understanding the Complex RAID10"*
- *Section 10.3.2, "Creating a Complex RAID 10 with mdadm"*
- *Section 10.3.3, "Creating a Complex RAID10 with the YaST Partitioner"*

10.3.1 Understanding the Complex 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 10.3.1.1, "Comparing the Complex RAID10 and Nested RAID 10 (1+0)"*
- *Section 10.3.1.2, "Number of Replicas in the Complex RAID10"*
- *Section 10.3.1.3, "Number of Devices in the Complex RAID10"*
- *Section 10.3.1.4, "Near Layout"*
- *Section 10.3.1.5, "Far Layout"*
- *Section 10.3.1.6, "Offset Layout"*

10.3.1.1 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	Complex RAID10	Nested RAID 10 (1 + 0)
Number of devices	Allows an even or odd number of component devices	Requires an even number of component devices

Feature	Complex RAID10	Nested RAID 10 (1 + 0)
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. The far layout provides sequential read throughput that scales by number of drives, rather than number of RAID 1 pairs.	Striping occurs consecutively across component devices
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.

10.3.1.2 Number of Replicas in the Complex RAID10

When configuring an complex 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.

10.3.1.3 Number of Devices in the Complex 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.

10.3.1.4 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

10.3.1.5 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
```

10.3.1.6 Offset Layout

The offset layout duplicates stripes so that the multiple copies of a given chunk are laid out on consecutive drives and at consecutive offsets. Effectively, each stripe is duplicated and the copies are offset by one device. This should give similar read characteristics to a far layout if a suitably large chunk size is used, but without as much seeking for writes.

Offset layout with an even number of disks and two replicas:

sda1	sdb1	sdcl	sde1
0	1	2	3
3	0	1	2
4	5	6	7
7	4	5	6
8	9	10	11
11	8	9	10

Offset layout with an odd number of disks and two replicas:

sda1	sdb1	sdcl	sde1	sdf1
0	1	2	3	4
4	0	1	2	3
5	6	7	8	9
9	5	6	7	8
10	11	12	13	14
14	10	11	12	13

10.3.2 Creating a Complex RAID 10 with mdadm

The RAID10 option for **mdadm** creates a RAID 10 device without nesting. For information about RAID10, see [Section 10.3.1, "Understanding the Complex RAID10"](#).

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)
<u>/dev/sdf1</u>	<u>/dev/md3</u>
<u>/dev/sdg1</u>	
<u>/dev/sdh1</u>	
<u>/dev/sdi1</u>	

1. In YaST, create a 0x^{FD} 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:

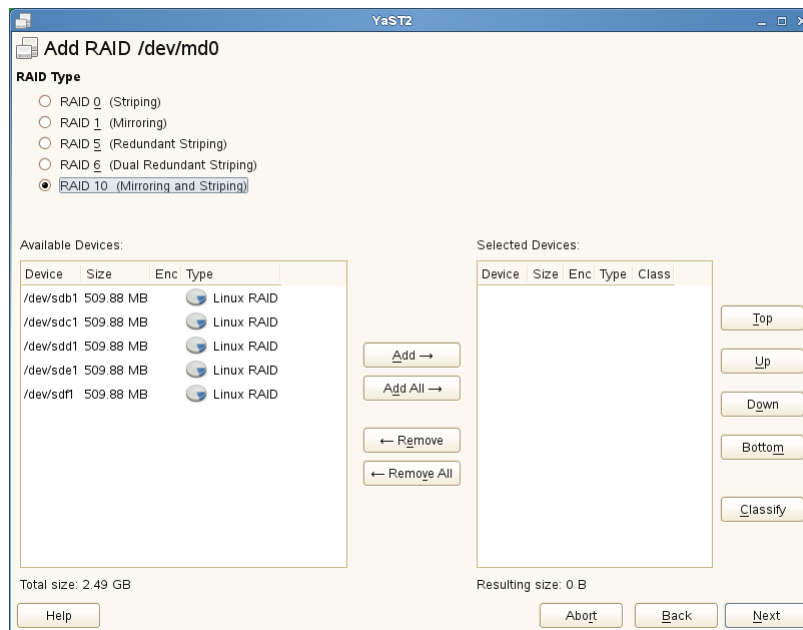
```
DEVICE /dev/md3
```

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.

10.3.3 Creating a Complex RAID10 with the YaST Partitioner

1. Launch YaST as the root user, then open the Partitioner.
2. Select *Hard Disks* to view the available disks, such as sdab, sdc, sdd, and sde.
3. For each disk that you will use in the software RAID, create a RAID partition on the device. Each partition should be the same size. For a RAID 10 device, you need
 - a. Under *Hard Disks*, select the device, then select the *Partitions* tab in the right panel.
 - b. Click *Add* to open the *Add Partition* wizard.
 - c. Under *New Partition Type*, select *Primary Partition*, then click *Next*.
 - d. For *New Partition Size*, specify the desired size of the RAID partition on this disk, then click *Next*.
 - e. Under *Formatting Options*, select *Do not format partition*, then select *0xFD Linux RAID* from the *File system ID* drop-down list.
 - f. Under *Mounting Options*, select *Do not mount partition*, then click *Finish*.
 - g. Repeat these steps until you have defined a RAID partition on the disks you want to use in the RAID 10 device.
4. Create a RAID 10 device:
 - a. Select *RAID*, then select *Add RAID* in the right panel to open the *Add RAID* wizard.
 - b. Under *RAID Type*, select *RAID 10 (Mirroring and Striping)*.



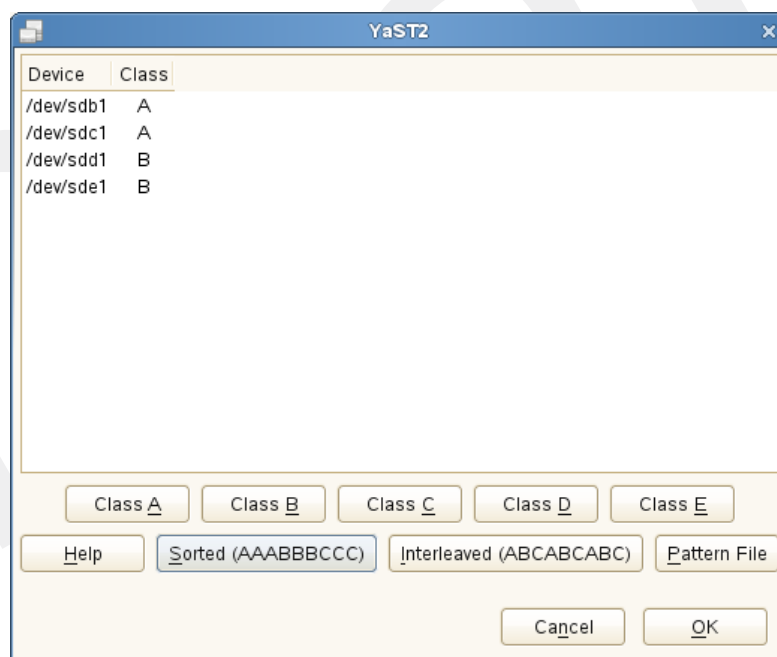
- c. In the *Available Devices* list, select the desired Linux RAID partitions, then click *Add* to move them to the *Selected Devices* list.
- d. (Optional) Click *Classify*, specify the preferred order of the disks in the RAID array. For RAID types where the order of added disks matters, you can specify the order in which the devices will be used to ensure that one half of the array resides on one disk subsystem and the other half of the array resides on a different disk subsystem. For example, if one disk subsystem fails, the system keeps running from the second disk subsystem.
 - i. Select each disk in turn and click one of the *Class X* buttons, where *X* is the letter you want to assign to the disk. Available classes are A, B, C, D and E but for many cases fewer classes are needed (e.g. only A and B). Assign all available RAID disks this way.
You can press the Ctrl or Shift key to select multiple devices. You can also right-click a selected device and choose the appropriate class from the context menu.
 - ii. Specify the order the devices by selecting one of the sorting options:

 Sorted: Sorts all devices of class A before all devices of class B and so on. For example: AABBCC .

Interleaved: Sorts devices by the first device of class A, then first device of class B, then all the following classes with assigned devices. Then the second device of class A, the second device of class B, and so on follows. All devices without a class are sorted to the end of devices list. For example, ABCABC .

Pattern File: Select an existing file that contains multiple lines, where each is a regular expression and a class name ("sda.* A"). All devices that match the regular expression are assigned to the specified class for that line. The regular expression is matched against the kernel name (/dev/sda1), the udev path name (/dev/disk/by-path/pci-0000:00:1f.2-scsi-0:0:0:0-part1) and then the udev ID (/dev/disk/by-id/ata-ST3500418AS_9VMN8X8L-part1). The first match made determines the class if a device's name matches more then one regular expression.

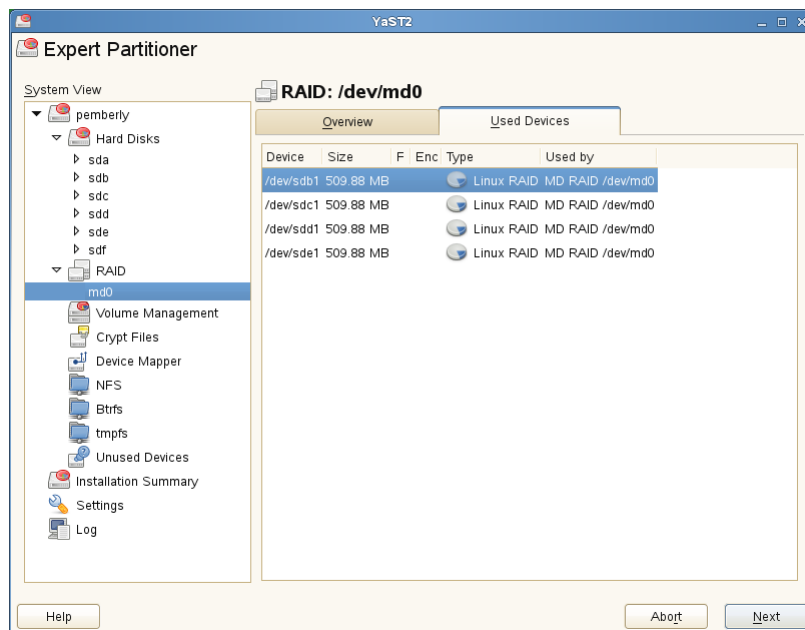
- iii. At the bottom of the dialog box, click *OK* to confirm the order.



- e. Click *Next*.
- f. Under *RAID Options*, specify the *Chunk Size* and *Parity Algorithm*, then click *Next*.
For a RAID 10, the parity options are n (near), f (far), and o (offset). The number indicates the number of replicas of each data block are required. Two is the default. For information, see [Section 10.3.1, "Understanding the Complex RAID10"](#).

g. Add a file system and mount options to the RAID device, then click *Finish*.

5. Select *RAID*, select the newly created RAID device, then click *Used Devices* to view its partitions.



6. Click *Next*.

7. Verify the changes to be made, then click *Finish* to create the RAID.

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 Type	Allowable Number of Slots Missing	
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
```

11 Resizing Software RAID Arrays with mdadm

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”*
- *Section 11.2, “Increasing the Size of a Software RAID”*
- *Section 11.3, “Decreasing the Size of a Software RAID”*

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”*
- *Section 11.1.2, “Overview of Tasks”*

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, 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 Increasing Size	Order If Decreasing 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

Tasks	Description	Order If Increasing Size	Order If Decreasing Size
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”*.

- *Section 11.2.1, “Increasing the Size of Component Partitions”*
- *Section 11.2.2, “Increasing the Size of the RAID Array”*
- *Section 11.2.3, “Increasing the Size of the File System”*

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
<u>/dev/md0</u>	<u>/dev/sda1</u> <u>/dev/sdb1</u> <u>/dev/sdc1</u>

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* 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* through *Step 5* 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"*.

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"*), 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"](#).
- 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"](#)), 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.

11.2.3.1 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.

11.2.3.2 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"*.

- *Section 11.3.1, "Decreasing the Size of the File System"*
- *Section 11.3.2, "Decreasing the Size of the RAID Array"*
- *Section 11.3.3, "Decreasing the Size of Component Partitions"*

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.

- [Section 11.3.1.1, "Ext2 or Ext3"](#)
- [Section 11.3.1.2, "ReiserFS"](#)

11.3.1.1 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.

5. Continue with *Section 11.3.2, “Decreasing the Size of the RAID Array”*.

11.3.1.2 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 removable device.

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 print sizes in human-readable format, such as 1K, 234M, or 2G.

6. Continue with [Section 11.3.2, “Decreasing the Size of the RAID Array”](#).

11.3.2 Decreasing the Size of the RAID Array

After you have resized the file system, the RAID array configuration continues to use the original array size until you force it to reduce the available space. Use the **mdadm --grow** mode to force the RAID to use a smaller segment size. To do this, you must use the **-z** option to specify the amount of space in kilobytes to use from each device in the RAID. This size must be a multiple of the chunk size, and it must leave about 128KB of space for the RAID superblock to be written to the device.

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. Decrease the size of the array's device size 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.)

For example, the following command sets the segment size for each RAID device to about 40 GB where the chunk size is 64 KB. It includes 128 KB for the RAID superblock.

```
mdadm --grow /dev/md2 -z 41943168
```

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 11.3.3, “Decreasing the Size of Component Partitions”](#).
- If your array was not resized as you expected, you must reboot, then try this procedure again.

11.3.3 Decreasing the Size of Component Partitions

After you decrease the segment size that is used on each device in the RAID, the remaining space in each component partition is not used by the RAID. You can leave partitions at their current size to allow for the RAID to grow at a future time, or you can reclaim this now unused space.

To reclaim the space, you decrease the component partitions one at a time. For each component partition, you remove it from the RAID, reduce its partition size, return the partition to the RAID, then wait until the RAID stabilizes. To allow for metadata, you should specify a slightly larger size than the size you specified for the RAID in [Section 11.3.2, “Decreasing the Size of the RAID Array”](#).

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
<u>/dev/md0</u>	<u>/dev/sda1</u> <u>/dev/sdb1</u> <u>/dev/sdc1</u>

To decrease 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. Decrease the size of the partition that you removed in [Step 3](#) to a size that is slightly larger than the size you set for the segment size. The size should be a multiple of the chunk size and allow 128 KB for the RAID superblock. Use a disk partitioner such as [fdisk](#), [cfdisk](#), or [parted](#) to decrease the size of the partition.

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](#) through [Step 5](#) 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. (Optional) Expand the size of the RAID and file system to use the maximum amount of space in the now smaller component partitions:
 - a. Expand the size of the RAID to use the maximum amount of space that is now available in the reduced-size component partitions:

```
mdadm --grow /dev/md0 -z max
```

- b. Expand the size of the file system to use all of the available space in the newly resized RAID. For information, see [Section 11.2.3, "Increasing the Size of the File System"](#).

12 Storage Enclosure LED Utilities for MD Software RAIDs

Storage enclosure LED Monitoring utility ([`ledmon\(8\)`](#)) and LED Control ([`ledctl\(8\)`](#)) utility are Linux user space applications that use a broad range of interfaces and protocols to control storage enclosure LEDs. The primary usage is to visualize the status of Linux MD software RAID devices created with the `mdadm` utility. The `ledmon` daemon monitors the status of the drive array and updates the status of the drive LEDs. The `ledctl` utility allows you to set LED patterns for specified devices.

- *Section 12.1, "Supported LED Management Protocols"*
- *Section 12.2, "Supported Storage Enclosure Systems"*
- *Section 12.3, "Storage Enclosure LED Monitor Service (`ledmon(8)`)"*
- *Section 12.4, "Storage Enclosure LED Control Application (`ledctl(8)`)"*
- *Section 12.5, "Enclosure LED Utilities Configuration File (`ledctl.conf(5)`)"*
- *Section 12.6, "Additional Information"*

12.1 Supported LED Management Protocols

These LED utilities use the SGPIO (Serial General Purpose Input/Output) specification (Small Form Factor (SFF) 8485) and the SCSI Enclosure Services (SES) 2 protocol to control LEDs. They implement the International Blinking Pattern Interpretation (IBPI) patterns of the SFF-8489 specification for SGPIO. The IBPI defines how the SGPIO standards are interpreted as states for drives and slots on a backplane and how the backplane should visualize the states with LEDs.

Some storage enclosures do not adhere strictly to the SFF-8489 specification. An enclosure processor might accept an IBPI pattern but not blink the LEDs according to the SFF-8489 specification, or the processor might support only a limited number of the IBPI patterns.

LED management (AHCI) and SAF-TE protocols are not supported by the [`ledmon`](#) and [`ledctl`](#) utilities.

12.2 Supported Storage Enclosure Systems

The **ledmon** and **ledctl** applications have been verified to work with Intel storage controllers such as the Intel AHCI controller and Intel SAS controller. Beginning in SUSE Linux Enterprise Server 11 SP3, they also support PCIe-SSD (solid state disk) enclosure LEDs to control the storage enclosure status (OK, Fail, Rebuilding) LEDs of PCIe-SSD devices that are part of an MD software RAID volume. The applications might also work with the IBPI-compliant storage controllers of other vendors (especially SAS/SCSI controllers); however, other vendors' controllers have not been tested.

12.3 Storage Enclosure LED Monitor Service (ledmon(8))

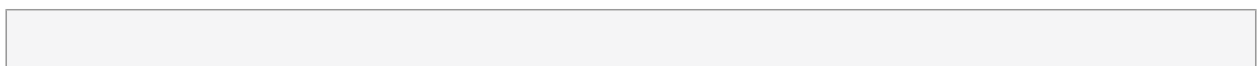
The **ledmon** application is a daemon process that constantly monitors the state of MD software RAID devices or the state of block devices in a storage enclosure or drive bay. Only a single instance of the daemon should be running at a time. The **ledmon** application is part of Intel Enclosure LED Utilities.

The state is visualized on LEDs associated with each slot in a storage array enclosure or a drive bay. The application monitors all software RAID devices and visualizes their state. It does not provide a way to monitor only selected software RAID volumes.

The **ledmon** application supports two types of LED systems: A two-LED system (Activity LED and Status LED) and a three-LED system (Activity LED, Locate LED, and Fail LED). This tool has the highest priority when accessing the LEDs.

- [Section 12.3.1, "Syntax"](#)
- [Section 12.3.2, "Options"](#)
- [Section 12.3.3, "Files"](#)
- [Section 12.3.4, "Known Issues"](#)

12.3.1 Syntax



```
ledmon [options]
```

Issue the command as the root user or as user with root privileges.

12.3.2 Options

-c,

--config=path

Sets a path to local configuration file. If this option is specified, the global configuration file and user configuration file have no effect.

-l,

--log=path

Sets a path to local log file. If this user-defined file is specified, the global log file /var/log/ledmon.log is not used.

-t,

--interval=seconds

Sets the time interval between scans of sysfs. The value is given in seconds. The minimum is 5 seconds. The maximum is not specified.

<--quiet|--error|--warning|--info|--debug|--all>

Specifies the verbose level. The level options are specified in the order of no information to the most information. Use the **--quiet** option for no logging. Use the **--all** option to log everything. If you specify more than one verbose option, the last option in the command applies.

-h,

--help

Prints the command information to the console, then exits.

-v,

--version

Displays version of ledmon and information about the license, then exits.

12.3.3 Files

`/var/log/ledmon.log`

Global log file, used by ledmon application. To force logging to a user-defined file, use the `-l` option.

`~/.ledctl`

User configuration file, shared between ledmon and all ledctl application instances.

`/etc/ledcfg.conf`

Global configuration file, shared between ledmon and all ledctl application instances.

12.3.4 Known Issues

The ledmon daemon does not recognize the PFA (Predicted Failure Analysis) state from the SFF-8489 specification. Thus, the PFA pattern is not visualized.

12.4 Storage Enclosure LED Control Application (ledctl(8))

The Enclosure LED Application (ledctl(8)) is a user space application that controls LEDs associated with each slot in a storage enclosure or a drive bay. The ledctl application is a part of Intel Enclosure LED Utilities.

When you issue the command, the LEDs of the specified devices are set to a specified pattern and all other LEDs are turned off. User must have root privileges to use this application. Because the ledmon application has the highest priority when accessing LEDs, some patterns set by ledctl might have no effect if ledmon is running (except the Locate pattern).

The ledctl application supports two types of LED systems: A two-LED system (Activity LED and Status LED) and a three-LED system (Activity LED, Fail LED, and Locate LED).

- *Section 12.4.1, "Syntax"*
- *Section 12.4.2, "Pattern Names"*
- *Section 12.4.3, "Pattern Translation"*
- *Section 12.4.4, "List of Devices"*

- [Section 12.4.5, “Options”](#)
- [Section 12.4.6, “Files”](#)
- [Section 12.4.7, “Examples”](#)

12.4.1 Syntax

```
ledctl [options] pattern_name=list_of_devices
```

Issue the command as the root user or as user with root privileges.

12.4.2 Pattern Names

The **ledctl** application accepts the following names for *pattern_name* argument, according to the SFF-8489 specification.

locate

Turns on the Locate LED associated with the specified devices or empty slots. This state is used to identify a slot or drive.

locate_off

Turns off the Locate LED associated with the specified devices or empty slots.

normal

Turns off the Status LED, Failure LED, and Locate LED associated with the specified devices.

off

Turns off only the Status LED and Failure LED associated with the specified devices.

ica,

degraded

Visualizes the In a Critical Array pattern.

rebuild,

rebuild_p

Visualizes the Rebuild pattern. This supports both of the rebuild states for compatibility and legacy reasons.

ifa,

failed_array

Visualizes the In a Failed Array pattern.

hotspare

Visualizes the Hotspare pattern.

pfa

Visualizes the Predicted Failure Analysis pattern.

failure,

disk_failed

Visualizes the Failure pattern.

ses_abort

SES-2 R/R ABORT

ses_rebuild

SES-2 REBUILD/REMAP

ses_ifa

SES-2 IN FAILED ARRAY

ses_ica

SES-2 IN CRITICAL ARRAY

ses_cons_check

SES-2 CONS CHECK

ses_hotspare

SES-2 HOTSPARE

ses_rsvd_dev

SES-2 RSVD DEVICE

ses_ok

SES-2 OK

ses_ident

SES-2 IDENT

ses_rm

SES-2 REMOVE

ses_insert
SES-2 INSERT

ses_missing
SES-2 MISSING

ses_dnr
SES-2 DO NOT REMOVE

ses_active
SES-2 ACTIVE

ses_enable_bb
SES-2 ENABLE BYP B

ses_enable_ba
SES-2 ENABLE BYP A

ses_devooff
SES-2 DEVICE OFF

ses_fault
SES-2 FAULT

12.4.3 Pattern Translation

When a non-SES-2 pattern is sent to a device in an enclosure, the pattern is automatically translated to the SCSI Enclosure Services (SES) 2 pattern as shown in *Table 12.1, “Translation between Non-SES-2 Patterns and SES-2 Patterns”*.

TABLE 12.1: TRANSLATION BETWEEN NON-SES-2 PATTERNS AND SES-2 PATTERNS

Non-SES-2 Pattern	SES-2 Pattern
locate	ses_ident
locate_off	ses_ident
normal	ses_ok
off	ses_ok
ica	ses_ica

Non-SES-2 Pattern	SES-2 Pattern
degraded	ses_ica
rebuild	ses_rebuild
rebuild_p	ses_rebuild
ifa	ses_ifa
failed_array	ses_ifa
hotspare	ses_hotspare
pfa	ses_rsvd_dev
failure	ses_fault
disk_failed	ses_fault

12.4.4 List of Devices

When you issue the **ledctl** command, the LEDs of the specified devices are set to the specified pattern and all other LEDs are turned off. The list of devices can be provided in one of two formats:

- A list of devices separated by a comma and no spaces
- A list in curly braces with devices separated by a space

If you specify multiple patterns in the same command, the device list for each pattern can use the same or different format. For examples that show the two list formats, see [Section 12.4.7, “Examples”](#).

A device is a path to file in the `/dev` directory or in the `/sys/block` directory. The path can identify a block device, an MD software RAID device, or a container device. For a software RAID device or a container device, the reported LED state is set for all of the associated block devices. The LEDs of devices listed in `list_of_devices` are set to the given pattern `pattern_name` and all other LEDs are turned off.

12.4.5 Options

-c,

--config=path

Sets a path to local configuration file. If this option is specified, the global configuration file and user configuration file have no effect.

-l,

--log=path

Sets a path to local log file. If this user-defined file is specified, the global log file /var/log/ledmon.log is not used.

--quiet

Turns off all messages sent to stdout or stderr out. The messages are still logged to local file and the syslog facility.

-h,

--help

Prints the command information to the console, then exits.

-v,

--version

Displays version of ledctl and information about the license, then exits.

12.4.6 Files

/var/log/ledctl.log

Global log file, used by all instances of the ledctl application. To force logging to a user-defined file, use the -l option.

~/.ledctl

User configuration file, shared between ledmon and all ledctl application instances.

/etc/ledcfg.conf

Global configuration file, shared between ledmon and all ledctl application instances.

12.4.7 Examples

To locate a single block device:

```
ledctl locate=/dev/sda
```

To turn off the Locate LED off for a single block device:

```
ledctl locate_off=/dev/sda
```

To locate disks of an MD software RAID device and to set a rebuild pattern for two of its block devices at the same time:

```
ledctl locate=/dev/md127 rebuild={ /sys/block/sd[a-b] }
```

To turn off the Status LED and Failure LED for the specified devices:

```
ledctl off={ /dev/sda /dev/sdb }
```

To locate three block devices:

```
ledctl locate=/dev/sda,/dev/sdb,/dev/sdc
```

```
ledctl locate={ /dev/sda /dev/sdb /dev/sdc }
```

12.5 Enclosure LED Utilities Configuration File (ledctl.conf(5))

The `ledctl.conf` file is the configuration file for the Intel Enclosure LED Utilities. The utilities do not use a configuration file at the moment. The name and location of file have been reserved for feature improvements.

12.6 Additional Information

See the following resources for details about the LED patterns and monitoring tools:

- LEDMON open source project on Sourceforge.net (<http://ledmon.sourceforge.net/>) 
- SGPIO specification SFF-8485 (<ftp://ftp.seagate.com/sff/SFF-8485.PDF>) 
- IBPI specification SFF-8489 (<ftp://ftp.seagate.com/sff/SFF-8489.PDF>) 

Draft

13 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 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 13.1, “How iSNS Works”*
- *Section 13.2, “Installing iSNS Server for Linux”*
- *Section 13.3, “Configuring iSNS Discovery Domains”*
- *Section 13.4, “Starting iSNS”*
- *Section 13.5, “Stopping iSNS”*
- *Section 13.6, “For More Information”*

13.1 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.

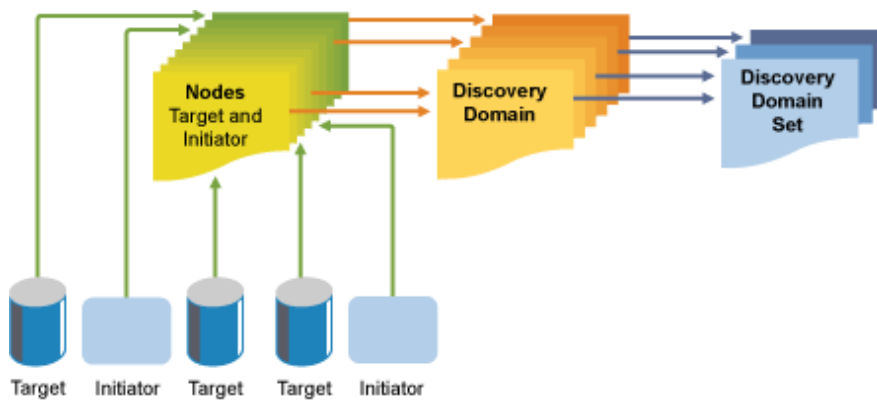


FIGURE 13.1: iSNS DISCOVERY DOMAINS AND DISCOVERY DOMAIN SETS

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.

13.2 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.



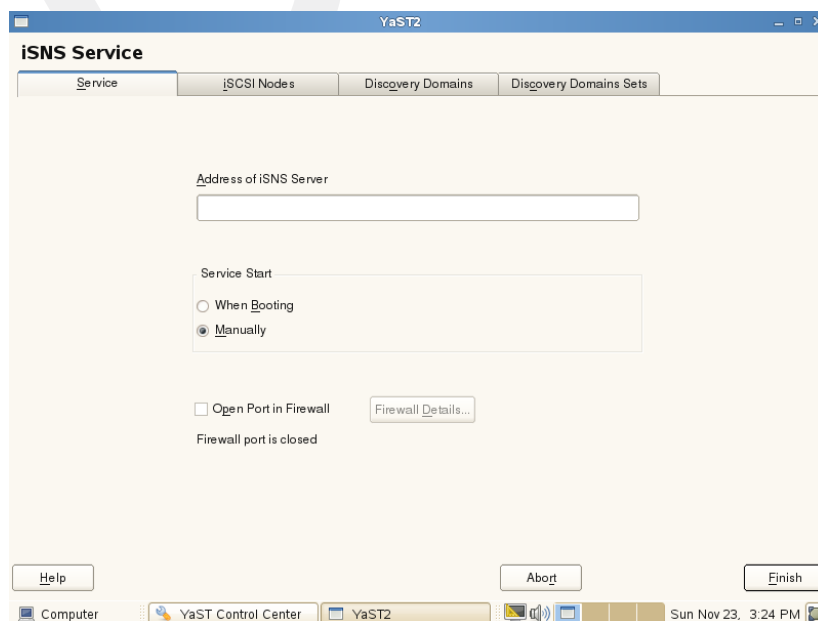
Note

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 instructions to provide the SUSE Linux Enterprise Server installation disks.

When the installation is complete, the iSNS Service configuration dialog box 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 `sudo systemctl start isns.service` 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 13.3, "Configuring iSNS Discovery Domains"*.

13.3 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 13.4, "Starting iSNS"*.

- *Section 13.3.1, "Creating iSNS Discovery Domains"*
- *Section 13.3.2, "Creating iSNS Discovery Domain Sets"*
- *Section 13.3.3, "Adding iSCSI Nodes to a Discovery Domain"*
- *Section 13.3.4, "Adding Discovery Domains to a Discovery Domain Set"*

13.3.1 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.

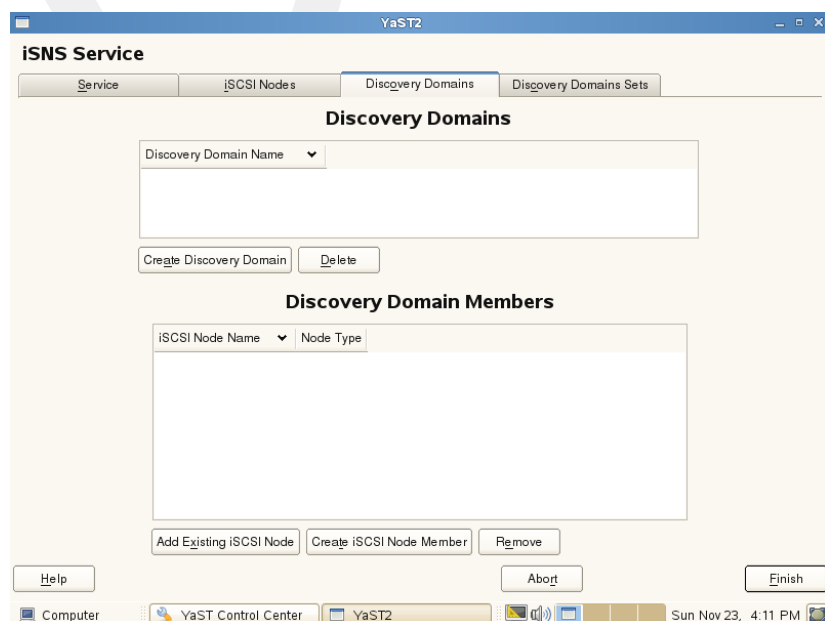
To create a new discovery domain:

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 13.3.2, “Creating iSNS Discovery Domain Sets”](#).

13.3.2 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*.

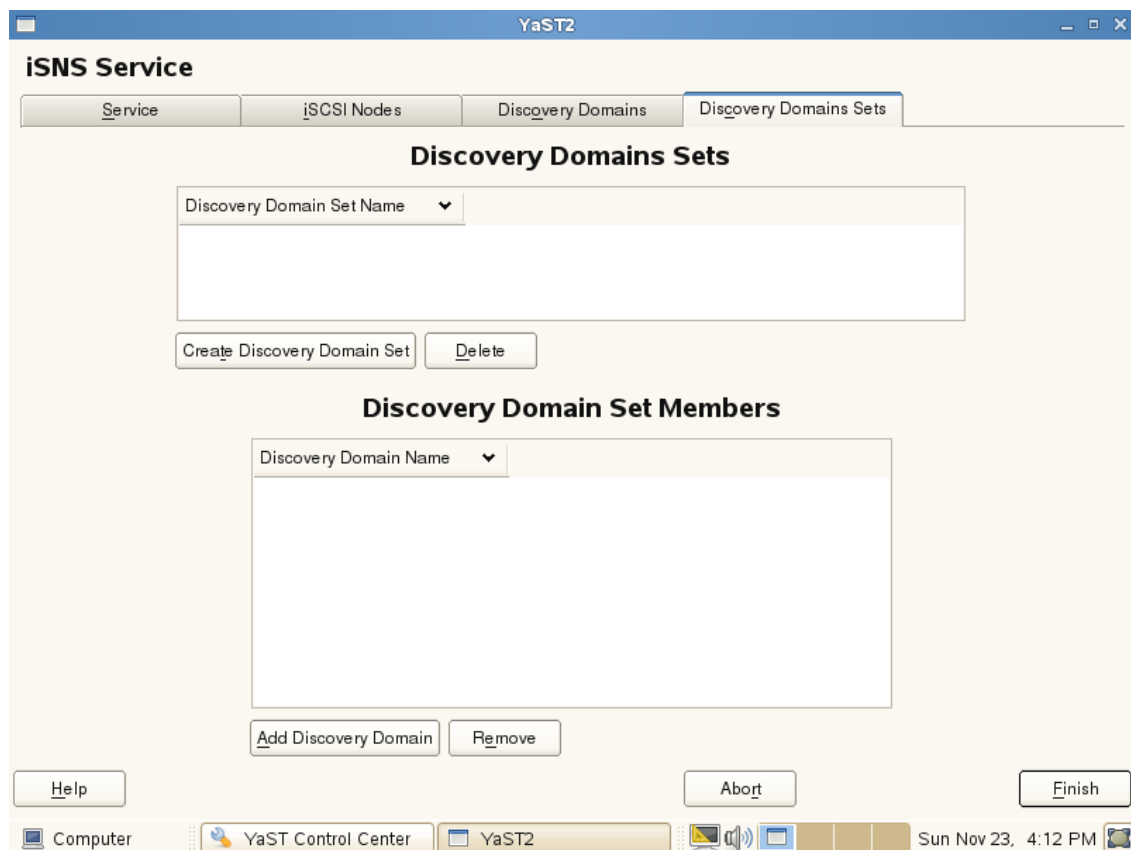
2. Click the *Discovery Domains Sets* tab.

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.

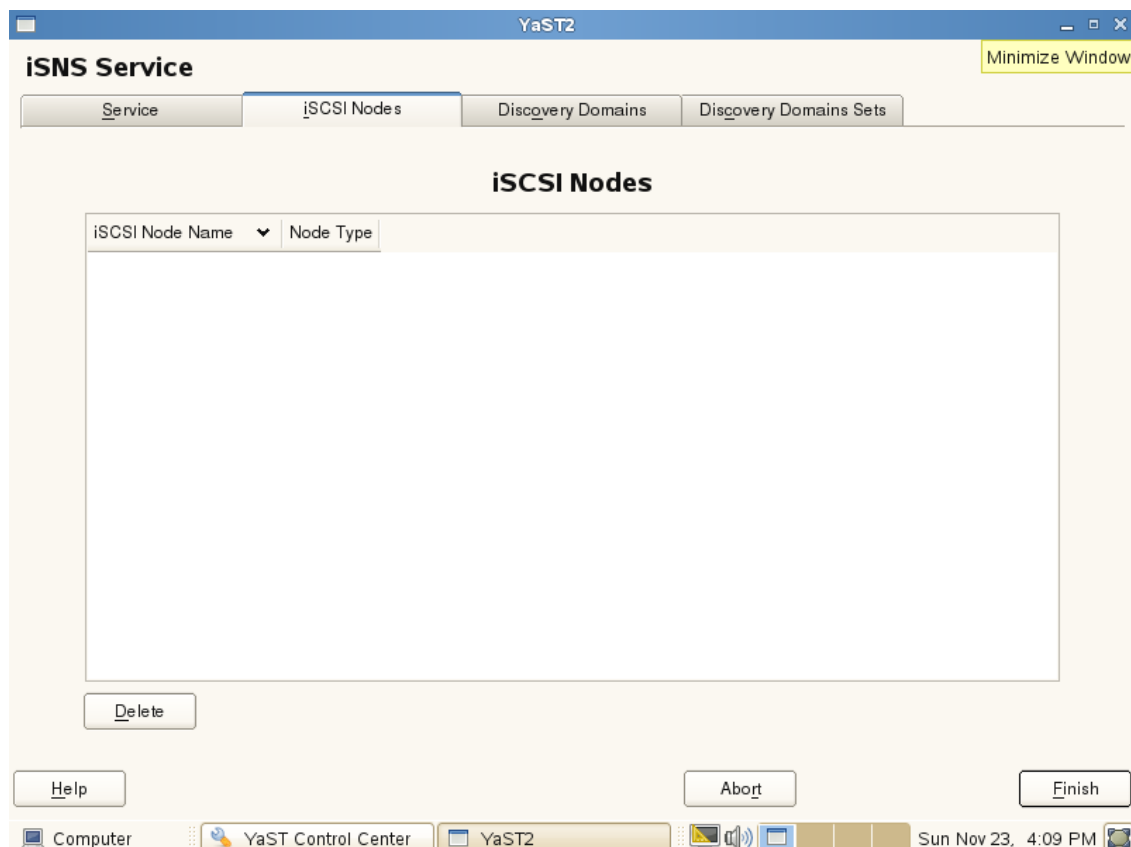
Adding an discovery domain to a set allows a not yet registered iSNS discovery domain to be added as a member of the discovery domain set.



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 13.3.3, "Adding iSCSI Nodes to a Discovery Domain"](#).

13.3.3 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 `sudo systemctl restart open-iscsi.service` command to restart an initiator or the `sudo systemctl restart iscsitarget.service` 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* 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 13.3.4, "Adding Discovery Domains to a Discovery Domain Set"*.

13.3.4 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.

13.4 Starting iSNS

iSNS must be started at the server where you install it. Enter the following command at a terminal console:

```
sudo systemctl start isns.service
```

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 `sudo systemctl start isns.service` to start the service each time the server is restarted.

13.5 Stopping iSNS

iSNS must be stopped at the server where it is running. Enter the following command at a terminal console:

```
sudo systemctl stop isns.service
```

13.6 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>).

14 Mass Storage over IP Networks: iSCSI

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.

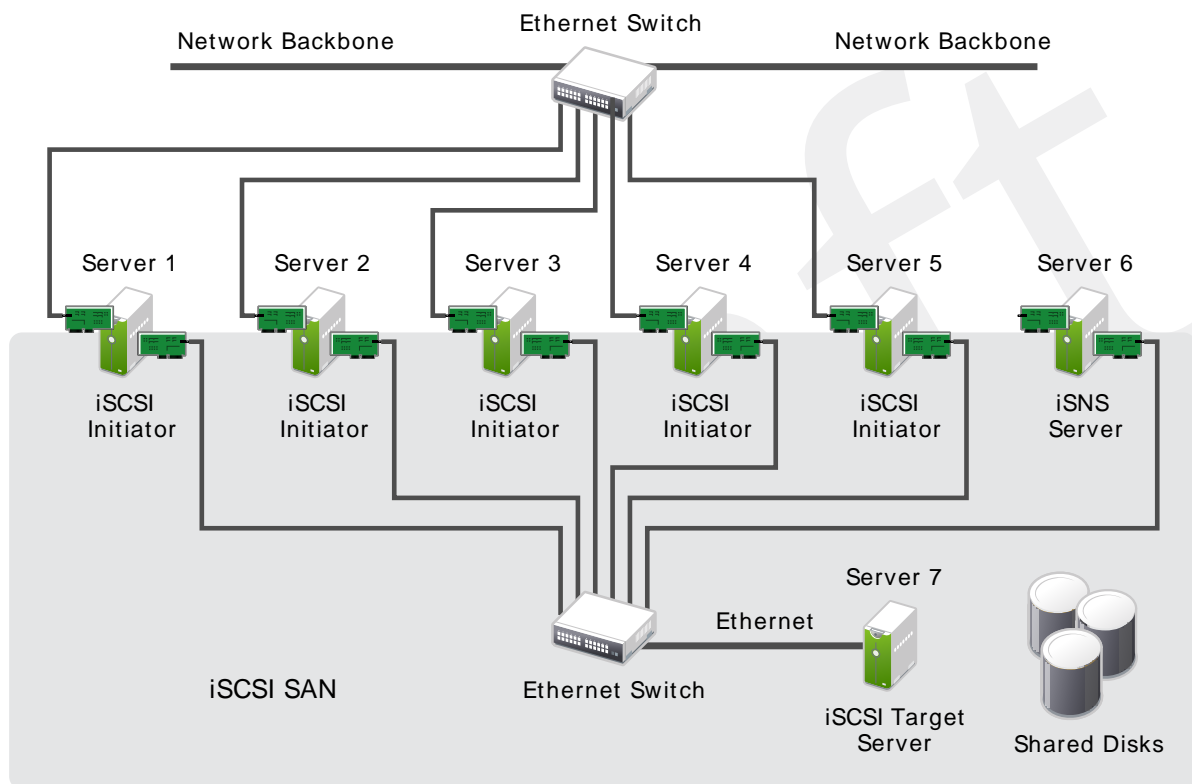


FIGURE 14.1: iSCSI SAN WITH AN iSNS SERVER

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](#).

- [Section 14.1, "Installing iSCSI Target and Initiator"](#)
- [Section 14.2, "Setting Up an iSCSI Target"](#)
- [Section 14.3, "Configuring iSCSI Initiator"](#)
- [Section 14.4, "Using iSCSI Disks when Installing"](#)
- [Section 14.5, "Troubleshooting iSCSI"](#)
- [Section 14.6, "Additional Information"](#)

14.1 Installing iSCSI Target and Initiator

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 14.1.1, "Installing iSCSI Target Software"](#)
- [Section 14.1.2, "Installing the iSCSI Initiator Software"](#)

14.1.1 Installing iSCSI Target Software

Install the iSCSI target software on the server where you want to create iSCSI target devices.

1. Launch YaST 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 14.2, "Setting Up an iSCSI Target"*.

14.1.2 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. Launch YaST 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 14.3, "Configuring iSCSI Initiator"*.

14.2 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 14.2.1, “Preparing the Storage Space”](#)
- [Section 14.2.2, “Creating iSCSI Targets with YaST”](#)
- [Section 14.2.3, “Configuring an iSCSI Target Manually”](#)
- [Section 14.2.4, “Configuring Online Targets with ietadm”](#)

14.2.1 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. iSCSI LIO targets can use unformatted partitions with Linux, Linux LVM, or Linux RAID file system IDs.

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 14.2.1.1, “Partitioning Devices”](#)
- [Section 14.2.1.2, “Partitioning Devices in a Virtual Environment”](#)

14.2.1.1 Partitioning Devices

1. Launch YaST as the root user.
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.
 - a. Select *Primary Partition*, then click *Next*.
 - b. Specify the amount of space to use, then click *Next*.
 - c. Select *Do not format*, then specify the file system ID type.

iSCSI targets can use unformatted partitions with Linux, Linux LVM, or Linux RAID file system IDs.

- d. Select *Do not mount*.
- e. Click *Finish*.
5. Repeat [Step 4](#) for each area that you want to use later as an iSCSI LUN.
6. Click *Accept* to keep your changes, then close YaST.

14.2.1.2 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 14.2.1.1, "Partitioning Devices"](#).

14.2.2 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 or DNS names; you must use the IP address. For more about iSNS, read [Chapter 13, iSNS for Linux](#).

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 box 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.

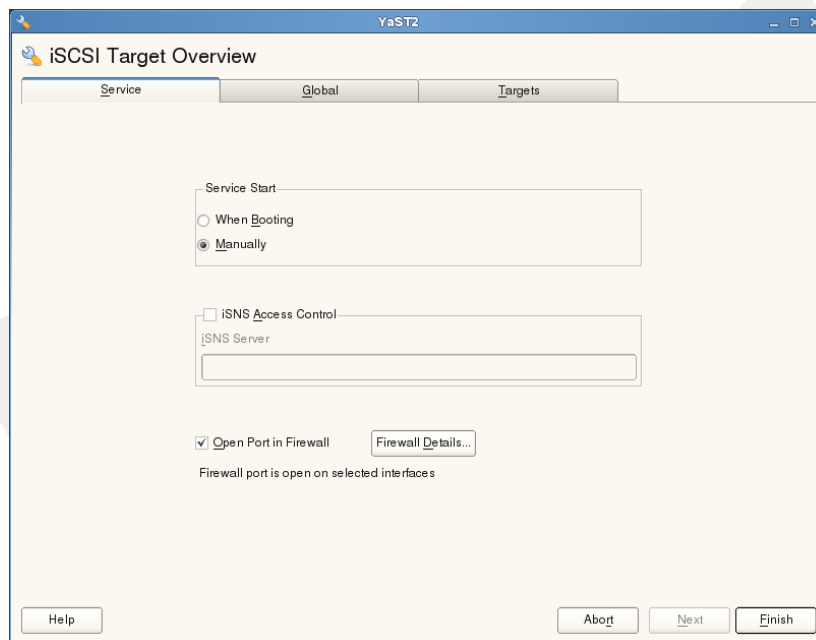
The next menu configures the access restrictions of the target. The configuration is very similar to the configuration of the discovery authentication. In this case, at least an incoming authentication should be setup.

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. Launch YaST as the root user.
2. Select *Network Services* *iSCSI Target*.

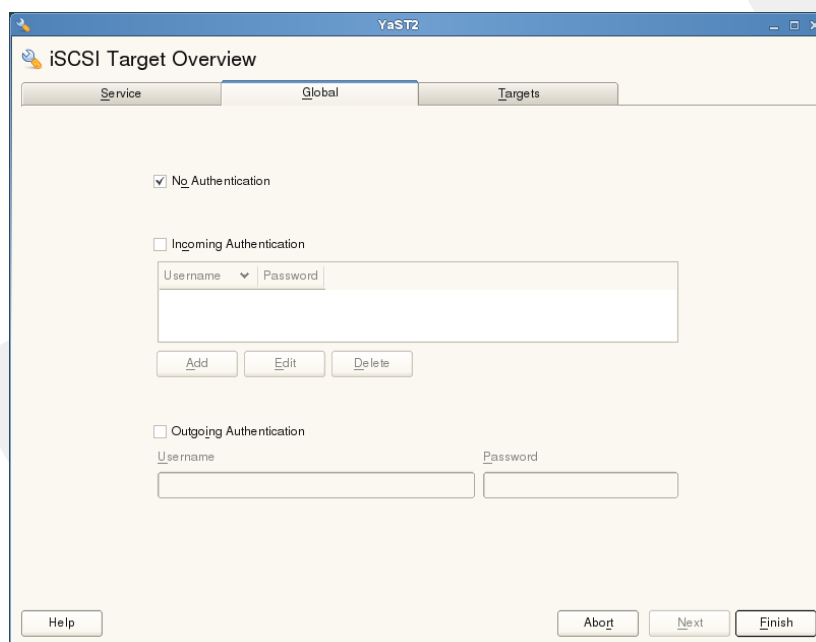
YaST opens to the iSCSI Target Overview page with the *Service* tab selected.



3. In the *Service Start* area, select one of the following:
 - **When booting:** Automatically start the initiator service on subsequent server reboots.
 - **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.

- a. Select the *Open Port in Firewall* check box.
 - b. 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.
 - a. Select the *Targets* tab.
 - b. If you have not already done so, select and delete the example iSCSI target from the list, then confirm the deletion by clicking *Continue*.
 - c. 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.

- d. 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*.
- e. Click *Next*
- f. Repeat [Step 7.c](#) to [Step 7.e](#) for each iSCSI target device you want to create.
- g. (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.
- h. Click *Finish* to create the devices, then click *Yes* to restart the iSCSI software stack.

14.2.3 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 13, iSNS for Linux](#). The access control for the iSNS discovery is not supported. 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*), 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 **sudo systemctl restart open-iscsi.service**. 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
```

14.2.4 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 iqn.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=iqn.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:iqn.2006-03.com.example.iserv:system
      sid:281474980708864 initiator:iqn.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
DataPDUInOrder=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`.

14.3 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 14.2, “Setting Up an iSCSI Target”](#). 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 14.3.1, “Using YaST for the iSCSI Initiator Configuration”](#)
- [Section 14.3.2, “Setting Up the iSCSI Initiator Manually”](#)
- [Section 14.3.3, “The iSCSI Client Databases”](#)

14.3.1 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.

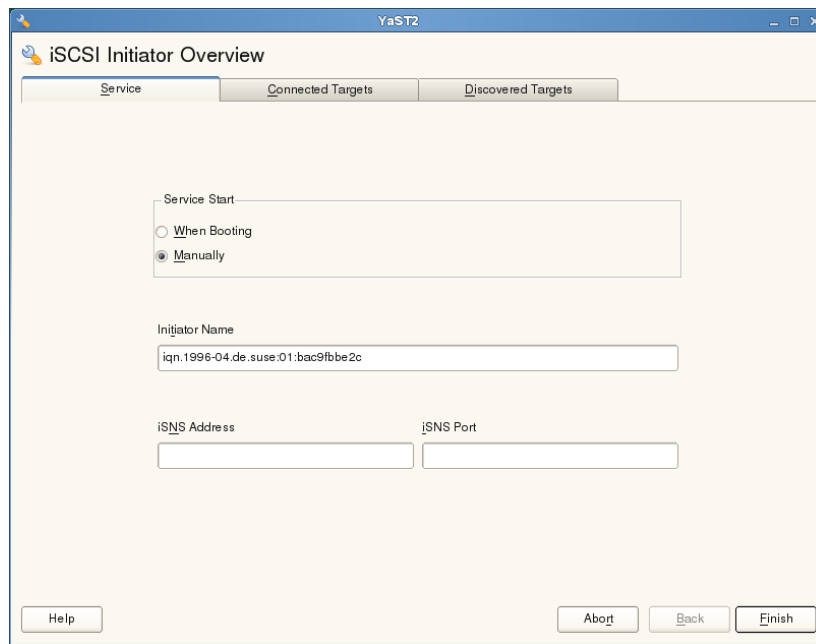
On this page, you can select a target device, then toggle the start-up setting for each iSCSI target device:

- **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 14.3.1.1, "Configuring the iSCSI Initiator"*
- *Section 14.3.1.2, "Discovering iSCSI Targets by Using iSNS"*
- *Section 14.3.1.3, "Discovering iSCSI Targets Manually"*
- *Section 14.3.1.4, "Setting the Start-up Preference for iSCSI Target Devices"*

14.3.1.1 Configuring the iSCSI Initiator

1. Launch YaST 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 14.3.1.2, “Discovering iSCSI Targets by Using iSNS”](#).
- **Discovered Targets:** To discover iSCSI target devices manually, continue with [Section 14.3.1.3, “Discovering iSCSI Targets Manually”](#).

14.3.1.2 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 13, iSNS for Linux](#).

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.

14.3.1.3 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 box.
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
```

14.3.1.4 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.

14.3.2 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 following command:

```
iscsiadm -m discovery --type=st --portal=<targetip>
```

The output should look like the following:

```
10.44.171.99:3260,1 iqn.2006-02.com.example.iserv:systems
```

To discover the available targets on a iSNS server, use the following 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 14.3.3, “The iSCSI Client Databases”](#).

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 **ls SCSI** and can now be accessed by mount.

14.3.3 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** with one of these parameters gives an overview of the stored records:

```
iscsiadm -m discovery
10.44.171.99:3260,1 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_cmdsn = 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 -p ip:port --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 -p ip:port --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.

14.4 Using iSCSI Disks when Installing

Booting 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.

14.5 Troubleshooting iSCSI

- *Section 14.5.1, “Hotplug Doesn’t Work for Mounting iSCSI Targets”*
- *Section 14.5.2, “Data Packets Dropped for iSCSI Traffic”*
- *Section 14.5.3, “Using iSCSI Volumes with LVM”*
- *Section 14.5.4, “iSCSI Targets Are Mounted When the Configuration File Is Set to Manual”*

14.5.1 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.suse.com/support/php/search.do?cmd=displayKC&docType=kc&externalId=7004427>) ↗.

14.5.2 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.

14.5.3 Using iSCSI Volumes with LVM

Use the troubleshooting tips in this section when using LVM on iSCSI targets.

- *Section 14.5.3.1, “Check the iSCSI Initiator Discovery Occurs at Boot”*
- *Section 14.5.3.2, “Check that iSCSI Target Discovery Occurs at Boot”*

14.5.3.1 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.

14.5.3.2 Check that iSCSI Target Discovery Occurs at Boot

Remember that `udev` provides the default setup for devices. Ensure that all of the applications that create devices are started at boot time 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 settings for LVM2 and iSCSI with *Services Manager* or with the `systemctl` commands. The following services should be enabled:

```
lvm
open-iscsi
```

14.5.4 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.

14.6 Additional Information

The iSCSI protocol has been available for several years. There are many reviews comparing iSCSI with SAN solutions, benchmarking performance, and there also is documentation 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/cool-solutions/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**.

15 Mass Storage over IP Networks: iSCSI LIO Target Server

LIO (linux-iscsi.org) is the standard open-source multiprotocol SCSI target for Linux. LIO replaced the STGT (SCSI Target) framework as the standard unified storage target in Linux with Linux kernel version 2.6.38 and later. YaST supports the iSCSI LIO Target Server software in SUSE Linux Enterprise Server 11 SP3 and later.

This section describes how to use YaST to configure an iSCSI LIO Target Server and set up iSCSI LIO target devices. You can use any iSCSI initiator software to access the target devices.

- *Section 15.1, "Installing the iSCSI LIO Target Server Software"*
- *Section 15.2, "Starting the iSCSI LIO Target Service"*
- *Section 15.3, "Configuring Authentication for Discovery of iSCSI LIO Targets and Clients"*
- *Section 15.4, "Preparing the Storage Space"*
- *Section 15.5, "Setting Up an iSCSI LIO Target Group"*
- *Section 15.6, "Modifying an iSCSI LIO Target Group"*
- *Section 15.7, "Deleting an iSCSI LIO Target Group"*
- *Section 15.8, "Troubleshooting iSCSI LIO Target Server"*
- *Section 15.9, "iSCSI LIO Target Terminology"*

15.1 Installing the iSCSI LIO Target Server Software

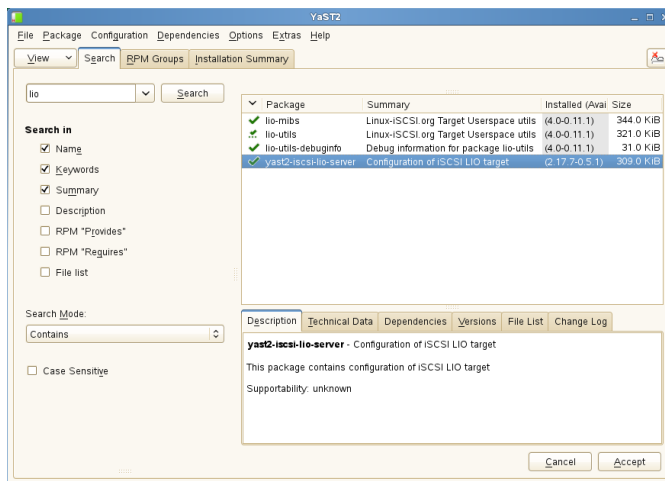
Use the YaST Software Management tool to install the iSCSI LIO Target Server software on the SUSE Linux Enterprise Server server where you want to create iSCSI LIO target devices.

1. Launch YaST as the root user.
2. Select *SoftwareSoftware Management*.

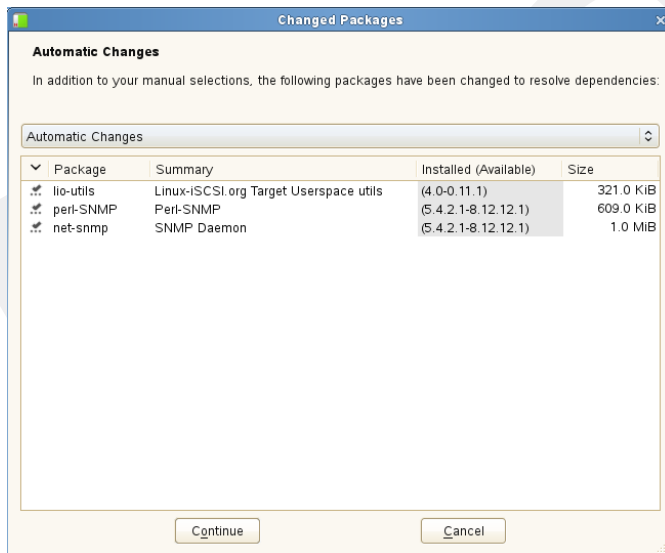
3. Select the *Search* tab, type lio, then click *Search*.
4. Select the iSCSI LIO Target Server packages:

iSCSI LIO Target Server Packages	Description
<u>yast2-iscsi-lio-server</u>	Provides a GUI interface in YaST for the configuration of iSCSI LIO target devices.
<u>lio-utils</u>	Provides APIs for configuring and controlling iSCSI LIO target devices that are used by <u>yast2-iscsi-lio-server</u> .
<u>lio-mibs</u>	<p>Provides SNMP (Simple Network Management Protocol) monitoring of iSCSI LIO target devices by using the dynamic load module (<u>dlmod</u>) functionality of the Net-SNMP agent. It supports SNMP v1, v2c, and v3. The configuration file is <u>/etc/snmp/snmpd.conf</u>.</p> <p>The <u>lio-mibs</u> software uses the <u>perl-SNMP</u> and <u>net-snmp</u> packages.</p> <p>For information about Net-SNMP, see the open source Net-SNMP Project (http://www.net-snmp.org/).</p>
<u>lio-utils-debuginfo</u>	Provides debug information for the <u>lio-utils</u> package. You can use this package when developing or debugging applications for <u>lio-utils</u> .

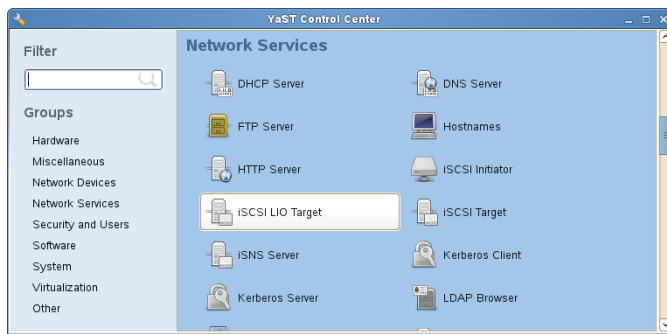
5. In the lower right corner of the dialog box, click *Accept* to install the selected packages.



- When you are prompted to approve the automatic changes, click *Continue* to accept the iSCSI LIO Target Server dependencies for the lio-utils, perl-SNMP, and net-snmp packages.



- Close and re-launch YaST, then click *Network Services* and verify that the *iSCSI LIO Target* option is available in the menu.



8. Continue with *Section 15.2, "Starting the iSCSI LIO Target Service"*.

15.2 Starting the iSCSI LIO Target Service

The iSCSI LIO Target service is by default configured to be started manually. You can configure the service to start automatically on system restart. If you use a firewall on the server and you want the iSCSI LIO targets to be available to other computers, you must open a port in the firewall for each adapter that you want to use for target access. TCP port 3260 is the port number for the iSCSI protocol, as defined by IANA (Internet Assigned Numbers Authority).

- *Section 15.2.1, "Configuring iSCSI LIO Startup Preferences"*
- *Section 15.2.2, "Manually Starting iSCSI LIO Target at the Command Line"*
- *Section 15.2.3, "Manually Starting iSCSI LIO Target in YaST"*

15.2.1 Configuring iSCSI LIO Startup Preferences

To configure the iSCSI LIO Target Server service settings:

1. Log in to the iSCSI LIO target server as the root user, then launch a terminal console.
2. Ensure that the /etc/init.d/target daemon is running. At the command prompt, enter

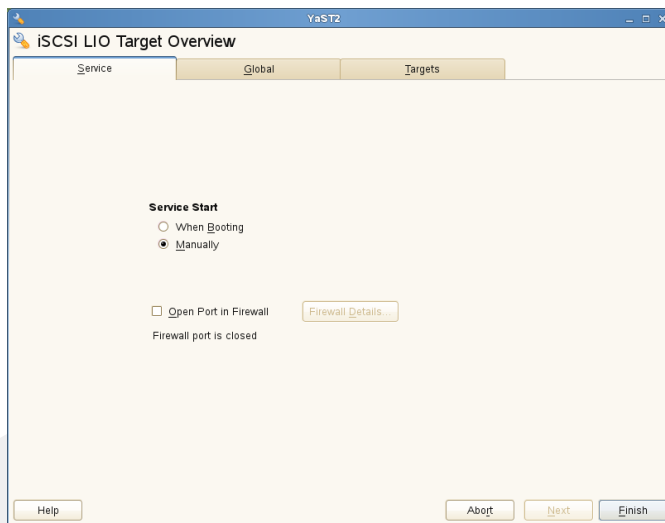
```
systemctl start target.service
```

The command returns a message to confirm that the daemon is started, or that the daemon is already running.

3. Launch YaST as the root user.
4. In the YaST Control Center, select *Network Services*, then select *iSCSI LIO Target*. You can also search for “lio”, then select *iSCSI LIO Target*.



5. In the iSCSI LIO Target Overview dialog box, select the *Service* tab.

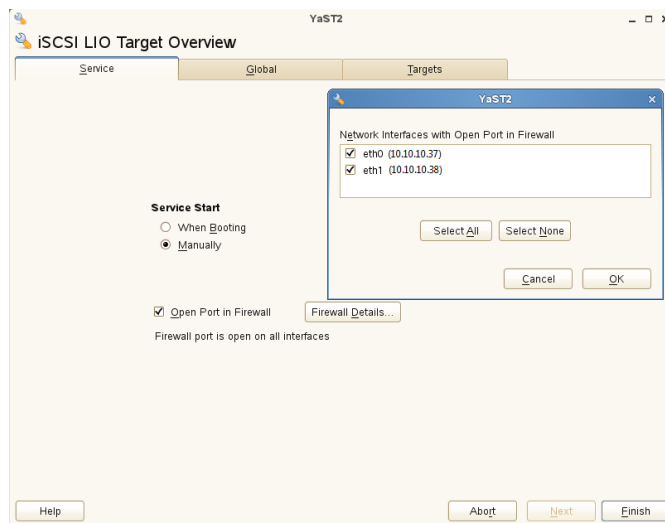


6. Under *Service Start*, specify how you want the iSCSI LIO target service to be started:
 - **When Booting:** The service starts automatically on server restart.
 - **Manually:** (Default) You must start the service manually after a server restart. The target devices are not available until you start the service.
7. If you use a firewall on the server and you want the iSCSI LIO targets to be available to other computers, open a port in the firewall for each adapter interface that you want to use for target access.

Firewall settings are disabled by default. They are not needed unless you deploy a firewall on the server. The default port number is 3260. If the port is closed for all of the network interfaces, the iSCSI LIO targets are not available to other computers.

 - a. On the *Services* tab, select the *Open Port in Firewall* check box to enable the firewall settings.

- b. Click *Firewall Details* to view or configure the network interfaces to use. All available network interfaces are listed, and all are selected by default.



- c. For each interface, specify whether to open or close a port for it:
- **Open:** Select the interface's check box to open the port. You can also click *Select All* to open a port on all of the interfaces.
 - **Close:** Deselect the interface's check box to close the port. You can also click *Select None* to close the port on all of the interfaces.
- d. Click *OK* to save and apply your changes.
- e. If you are prompted to confirm the settings, click *Yes* to continue, or click *No* to return to the dialog box and make the desired changes.
8. Click *Finish* to save and apply the iSCSI LIO Target service settings.

15.2.2 Manually Starting iSCSI LIO Target at the Command Line

1. Log in as the root user, then launch a terminal console.
2. At the command prompt, enter


```
systemctl start target.service
```

The command returns a message to confirm that the daemon is started, or that the daemon is already running.

15.2.3 Manually Starting iSCSI LIO Target in YaST

1. Launch YaST as the root user.
2. In the YaST Control Center, select *System*, then select *Services Manager*.
3. In the YaST Services Manager dialog box, select *target (TCM/ConfigFS and LIO-Target)* in the list of services.
4. If the service is listed as inactive, click *Start/Stop* to change this.
5. Click *OK*.

15.3 Configuring Authentication for Discovery of iSCSI LIO Targets and Clients

The iSCSI LIO Target Server software supports the PPP-CHAP (Point-to-Point Protocol Challenge Handshake Authentication Protocol), a three-way authentication method defined in the *Internet Engineering Task Force (IETF) RFC 1994* (<http://www.ietf.org/rfc/rfc1994.txt>) . iSCSI LIO Target Server uses this authentication method for the discovery of iSCSI LIO targets and clients, not for accessing files on the targets. If you do not want to restrict the access to the discovery, use *No Authentication*. The *No Authentication* option is enabled by default. If authentication for discovery is enabled, its settings apply to all iSCSI LIO target groups.

Important

We recommend that you use authentication for target and client discovery in production environments.

If authentication is needed for a more secure configuration, you can use incoming authentication, outgoing authentication, or both. *Incoming Authentication* requires an iSCSI initiator to prove that it has the permissions to run a discovery on the iSCSI LIO target. The initiator must provide the incoming user name and password. *Outgoing Authentication* requires the iSCSI LIO target to prove to the initiator that it is the expected target. The iSCSI LIO target must provide the outgoing user name and password to the iSCSI initiator. The user name and password pair can be different for incoming and outgoing discovery.

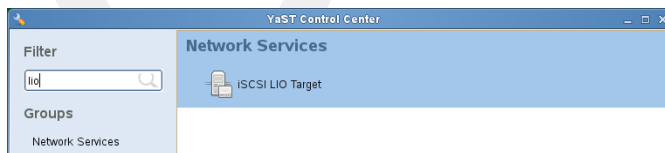
To configure authentication preferences for iSCSI LIO targets:

1. Log in to the iSCSI LIO target server as the root user, then launch a terminal console.
2. Ensure that the /etc/init.d/target daemon is running. At the command prompt, enter

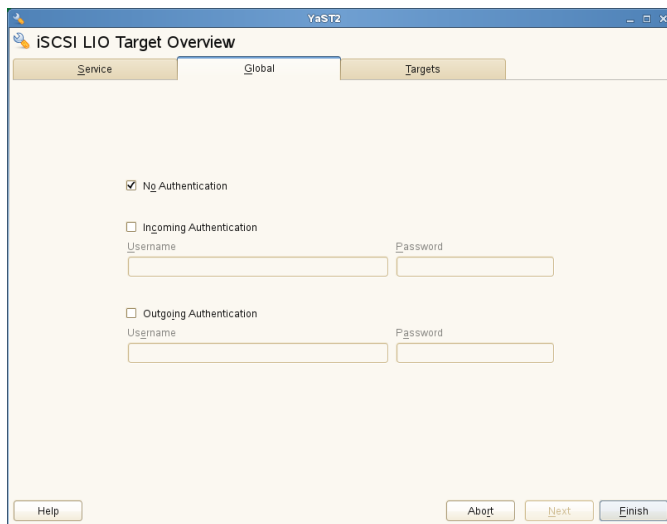
```
systemctl start target.service
```

The command returns a message to confirm that the daemon is started, or that the daemon is already running.

3. Launch YaST as the root user.
4. In the YaST Control Center, select *Network Services*, then select *iSCSI LIO Target*. You can also search for “lio”, then select *iSCSI LIO Target*.

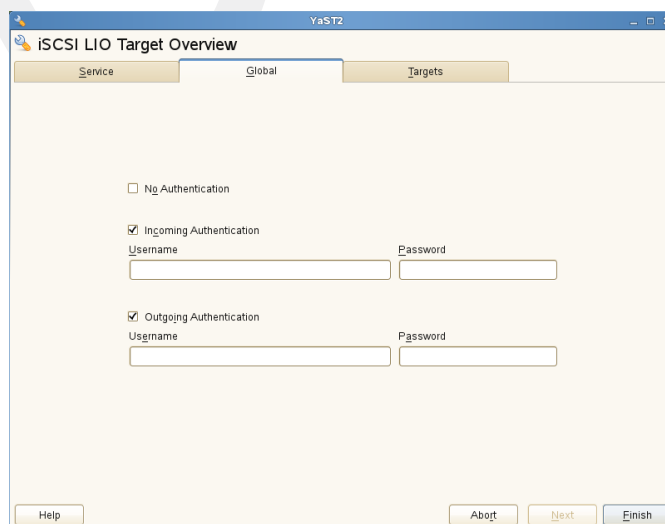


5. In the iSCSI LIO Target Overview dialog box, select the *Global* tab to configure the authentication settings. Authentication settings are disabled by default.



6. Specify whether to require authentication for iSCSI LIO targets:

- **Disable authentication:** (Default) Select the *No Authentication* check box to disable incoming and outgoing authentication for discovery on this server. All iSCSI LIO targets on this server can be discovered by any iSCSI initiator client on the same network. This server can discover any iSCSI initiator client on the same network that does not require authentication for discovery. Skip [Step 7](#) and continue with [Step 8](#).
- **Enable authentication:** Deselect the *No Authentication* check box. The check boxes for both *Incoming Authentication* and *Outgoing Authentication* are automatically selected. Continue with [Step 7](#).



7. Configure the authentication credentials needed for incoming discovery, outgoing discovery, or both. The user name and password pair can be different for incoming and outgoing discovery.
 - a. Configure incoming authentication by doing one of the following:
 - **Disable incoming authentication:** Deselect the *Incoming Authentication* check box. All iSCSI LIO targets on this server can be discovered by any iSCSI initiator client on the same network.
 - **Enable incoming authentication:** Select the *Incoming Authentication* check box, then specify an existing user name and password pair to use for incoming discovery of iSCSI LIO targets.
 - b. Configure outgoing authentication by doing one of the following:
 - **Disable outgoing authentication:** Deselect the *Outgoing Authentication* check box. This server can discover any iSCSI initiator client on the same network that does not require authentication for discovery.
 - **Enable outgoing authentication:** Select the *Outgoing Authentication* check box, then specify an existing user name and password pair to use for outgoing discovery of iSCSI initiator clients.
8. Click *Finish* to save and apply the settings.

15.4 Preparing the Storage Space

The iSCSI LIO 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 on the server. iSCSI LIO targets can use unformatted partitions with Linux, Linux LVM, or Linux RAID file system IDs.



Important

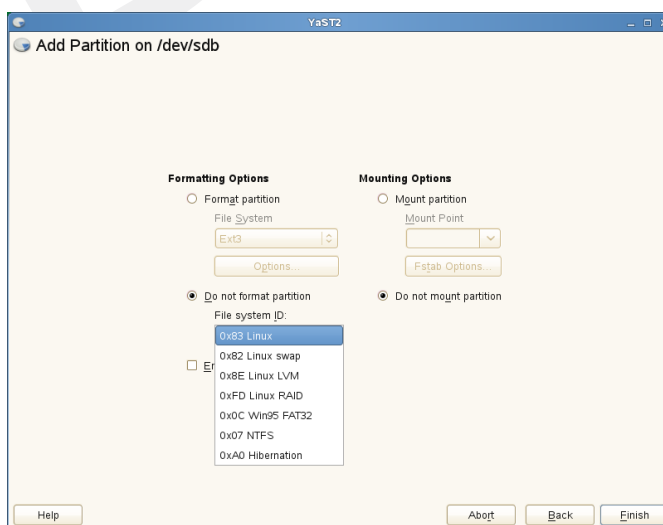
After you set up a device or partition for use as an iSCSI target, you never access it directly via its local path. Do not specify a mount point for it when you create it.

- Section 15.4.1, “Partitioning Devices”
- Section 15.4.2, “Partitioning Devices in a Virtual Environment”

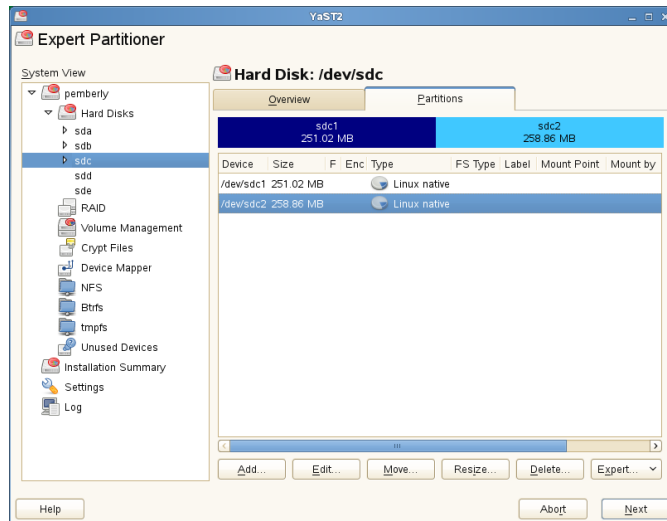
15.4.1 Partitioning Devices

1. Launch YaST as the root user.
2. In YaST, select *SystemPartitioner*.
3. Click *Yes* to continue through the warning about using the Partitioner.
4. At the bottom of the Partitions page, click *Add* to create a partition, but do not format it, and do not mount it.
 - a. On the Expert Partitioner page, select *Hard Disks*, then select the leaf node name (such as sdc) of the disk you want to configure.
 - b. Select *Primary Partition*, then click *Next*.
 - c. Specify the amount of space to use, then click *Next*.
 - d. Under *Formatting Options*, select *Do not format*, then select the file system ID type from the drop-down list.

iSCSI LIO targets can use unformatted partitions with Linux (0x83), Linux LVM (0x8E), or Linux RAID (0xFD) file system IDs.



- e. Under *Mounting Options*, select *Do not mount*.
 - f. Click *Finish*.
5. Repeat *Step 4* to create an unformatted partition for each area that you want to use later as an iSCSI LIO target.



6. Click *NextFinish* to keep your changes, then close YaST.

15.4.2 Partitioning Devices in a Virtual Environment

You can use a virtual machine guest server as a iSCSI LIO Target Server. This section describes how to assign partitions to a Xen virtual machine. You can also use other virtual environments that are supported by SUSE Linux Enterprise Server 11 SP2 or later.

In a Xen virtual environment, you must assign the storage space you want to use for the iSCSI LIO target 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 15.4.1, "Partitioning Devices"](#).

15.5 Setting Up an iSCSI LIO Target Group

You can use YaST to configure iSCSI LIO target devices. YaST uses APIs provided by the **lio-utils** software. iSCSI LIO targets can use unformatted partitions with Linux, Linux LVM, or Linux RAID file system IDs.

! Important

Before you begin, create the unformatted partitions that you want to use as iSCSI LIO targets as described in [Section 15.4, "Preparing the Storage Space"](#).

1. Log in to the iSCSI LIO target server as the `root` user, then launch a terminal console.
2. Ensure that the `/etc/init.d/target` daemon is running. At the command prompt, enter

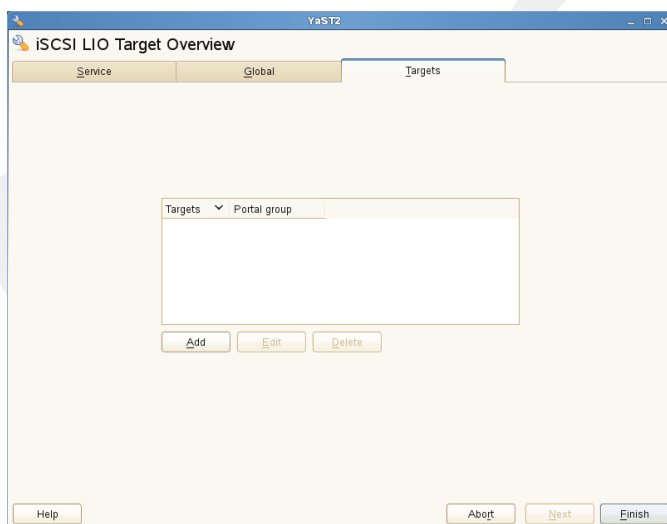
```
systemctl start target.service
```

The command returns a message to confirm that the daemon is started, or that the daemon is already running.

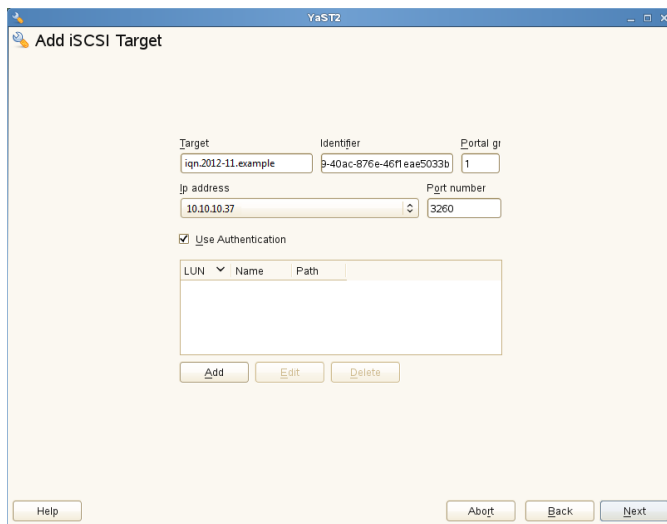
3. Launch YaST as the root user.
4. In the YaST Control Center, select *Network Services*, then select *iSCSI LIO Target*. You can also search for “lio”, then select *iSCSI LIO Target*.



5. In the iSCSI LIO Target Overview dialog box, select the *Targets* tab to configure the targets.



6. Click *Add*, then define a new iSCSI LIO target group and devices:
The iSCSI LIO Target software automatically completes the *Target*, *Identifier*, *Portal Group*, *IP Address*, and *Port Number* fields. *Use Authentication* is selected by default.

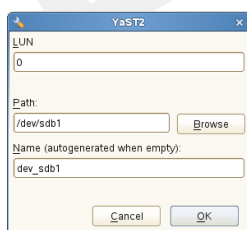


- a. If you have multiple network interfaces, use the IP address drop-down list to select the IP address of the network interface to use for this target group.
- b. Select *Use Authentication* if you want to require client authentication for this target group.

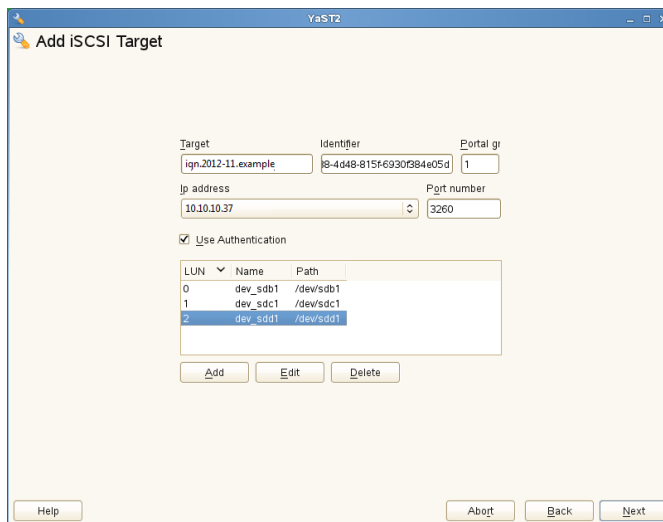
! Important

Requiring authentication is recommended in a production environment.

- c. Click *Add*, browse to select the device or partition, specify a name, then click *OK*. The LUN number is automatically generated, beginning with 0. A name is automatically generated if you leave the field empty.



- d. (Optional) Repeat *Step 6.a* through *Step 6.c* to add more targets to this target group.



Add iSCSI Target

Target: Identifier: Portal gr:

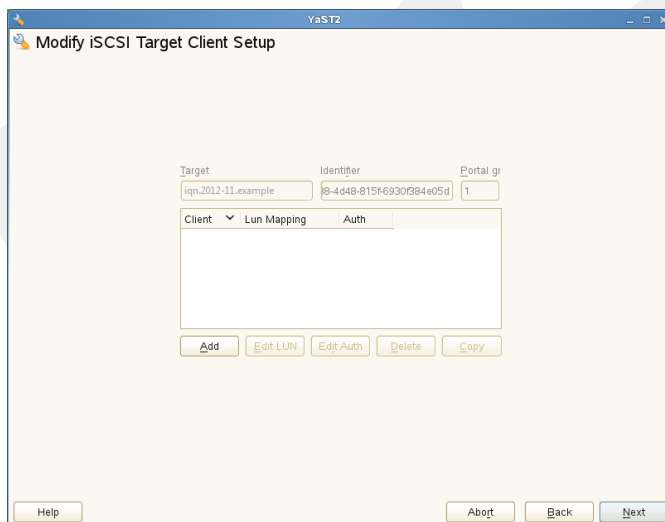
Ip address: Port number:

☒ Use Authentication

LUN	Name	Path
0	dev_sdb1	/dev/sdb1
1	dev_sdc1	/dev/sdc1
2	dev_sdd1	/dev/sdd1

e. After all desired targets have been added to the group, click *Next*.

- On the Modify iSCSI Target Client Setup page, configure information for the clients that are permitted to access LUNs in the target group:



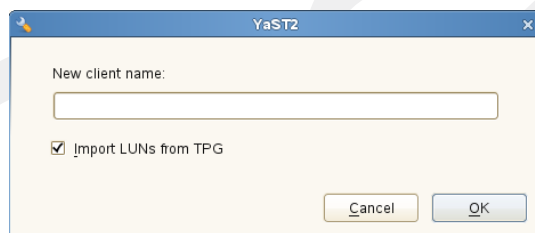
Modify iSCSI Target Client Setup

Target: Identifier: Portal gr:

Client

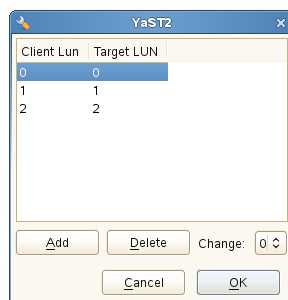
After you specify at least one client for the target group, the *Edit LUN*, *Edit Auth*, *Delete*, and *Copy* buttons are enabled. You can use *Add* or *Copy* to add more clients for the target group.

- **Add:** Add a new client entry for the selected iSCSI LIO target group.
 - **Edit LUN:** Configure which LUNs in the iSCSI LIO target group to map to a selected client. You can map each of the allocated targets to a preferred client LUN.
 - **Edit Auth:** Configure the preferred authentication method for a selected client. You can specify no authentication, or you can configure incoming authentication, outgoing authentication, or both.
 - **Delete:** Remove a selected client entry from the list of clients allocated to the target group.
 - **Copy:** Add a new client entry with the same LUN mappings and authentication settings as a selected client entry. This allows you to easily allocate the same shared LUNs, in turn, to each node in a cluster.
- a. Click *Add*, specify the client name, select or deselect the *Import LUNs from TPG* check box, then click *OK* to save the settings.



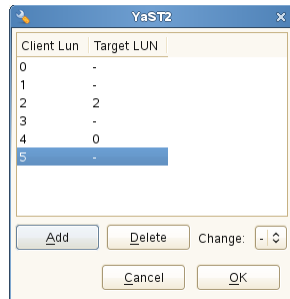
- b. Select a client entry, click *Edit LUN*, modify the LUN mappings to specify which LUNs in the iSCSI LIO target group to allocate to the selected client, then click *OK* to save the changes.

If the iSCSI LIO target group consists of multiple LUNs, you can allocate one or multiple LUNs to the selected client. By default, each of the available LUNs in the group are assigned to a Client LUN.

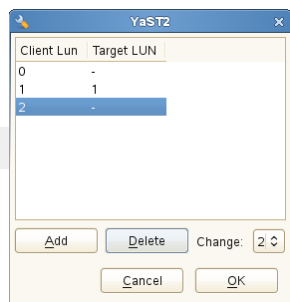


To modify the LUN allocation, perform one or more of the following actions:

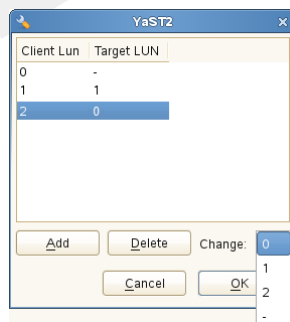
- **Add:** Click *Add* to create a new *Client LUN* entry, then use the *Change* drop-down list to map a Target LUN to it.



- **Delete:** Select the *Client LUN* entry, then click *Delete* to remove a Target LUN mapping.



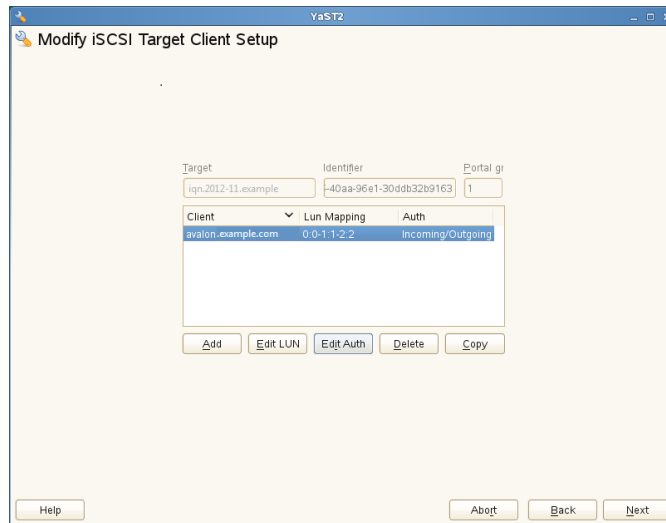
- **Change:** Select the *Client LUN* entry, then use the *Change* drop-down list to select which Target LUN to map to it.



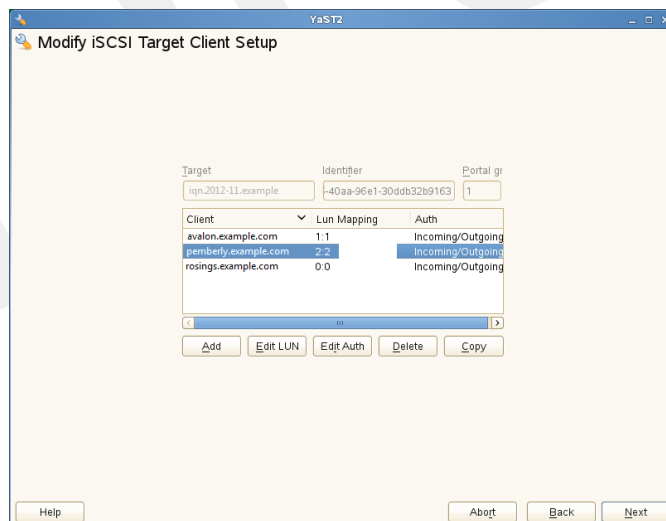
Typical allocation plans include the following:

- A single server is listed as a client. All of the LUNs in the target group are allocated to it.

You can use this grouping strategy to logically group the iSCSI SAN storage for a given server.



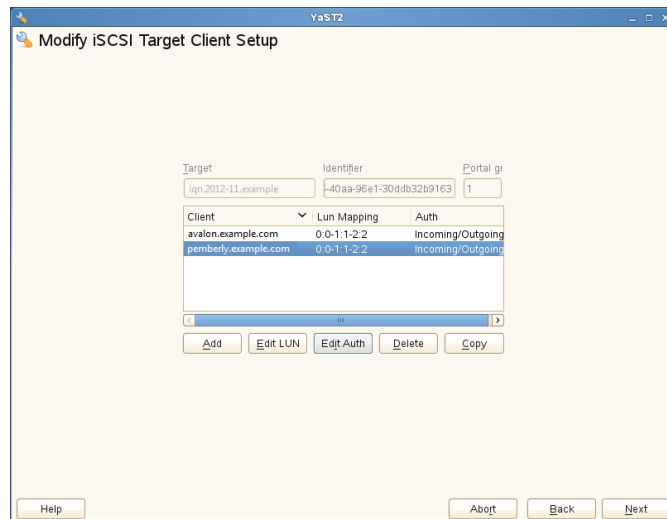
- Multiple independent servers are listed as clients. One or multiple target LUNs are allocated to each server. Each LUN is allocated to only one server. You can use this grouping strategy to logically group the iSCSI SAN storage for a given department or service category in the data center.



- Each node of a cluster is listed as a client. All of the shared target LUNs are allocated to each node. All nodes are attached to the devices, but for most file systems, the cluster software locks a device for access and mounts it on only

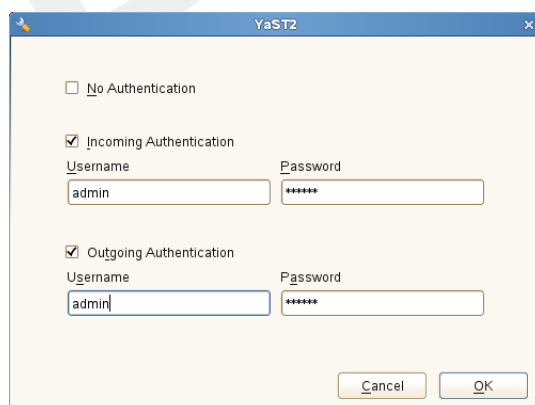
one node at a time. Shared file systems (such as OCFS2) make it possible for multiple nodes to concurrently mount the same file structure and to open the same files with read and write access.

You can use this grouping strategy to logically group the iSCSI SAN storage for a given server cluster.



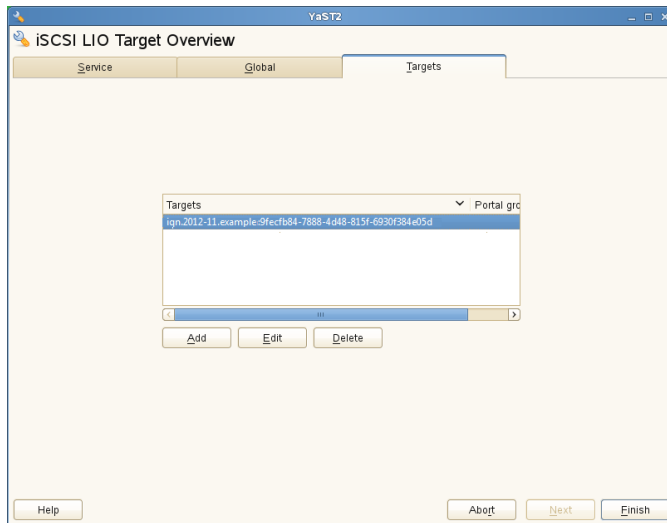
- c. Select a client entry, click *Edit Auth*, specify the authentication settings for the client, then click *OK* to save the settings.

You can require *No Authentication*, or you can configure *Incoming Authentication*, *Outgoing Authentication*, or both. You can specify only one user name and password pair for each client. The credentials can be different for incoming and outgoing authentication for a client. The credentials can be different for each client.



- d. Repeat [Step 7.a](#) through [Step 7.c](#) for each iSCSI client that can access this target group.
- e. After the client assignments are configured, click *Next*.

8. Click *Finish* to save and apply the settings.



15.6 Modifying an iSCSI LIO Target Group

You can modify an existing iSCSI LIO target group as follows:

- Add or remove target LUN devices from a target group
- Add or remove clients for a target group
- Modify the client LUN-to-target LUN mappings for a client of a target group
- Modify the user name and password credentials for a client authentication (incoming, outgoing, or both)

To view or modify the settings for an iSCSI LIO target group:

1. Log in to the iSCSI LIO target server as the root user, then launch a terminal console.
2. Ensure that the /etc/init.d/target daemon is running. At the command prompt, enter

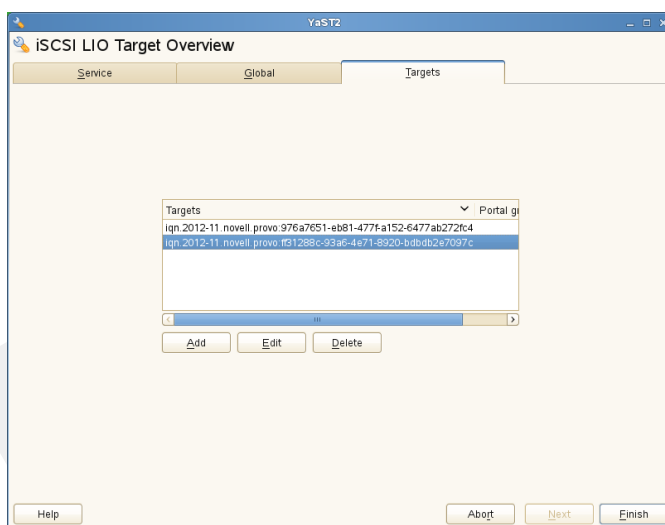
```
systemctl start target.service
```

The command returns a message to confirm that the daemon is started, or that the daemon is already running.

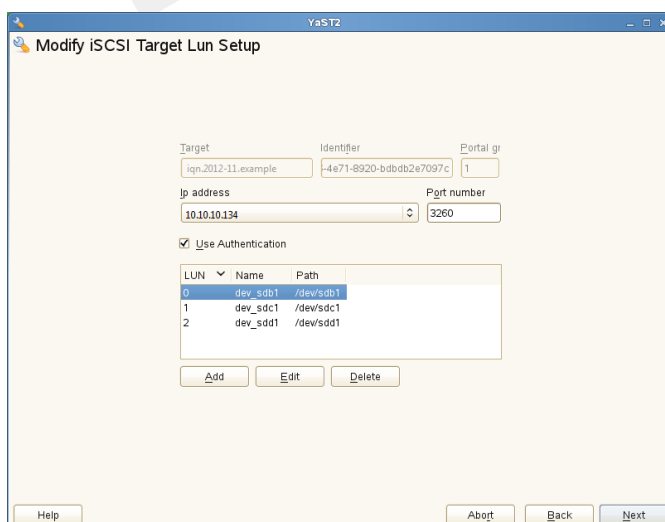
3. Launch YaST as the root user.
4. In the YaST Control Center, select *Network Services*, then select *iSCSI LIO Target*.
You can also search for “lio”, then select *iSCSI LIO Target*.



5. In the iSCSI LIO Target Overview dialog box, select the *Targets* tab to view a list of target groups.



6. Select the iSCSI LIO target group to be modified, then click *Edit*.



7. On the Modify iSCSI Target LUN Setup page, add LUNs to the target group, edit the LUN assignments, or remove target LUNs from the group. After all desired changes have been made to the group, click *Next*.
For option information, see *Step 6* in *Section 15.5, “Setting Up an iSCSI LIO Target Group”*.
8. On the Modify iSCSI Target Client Setup page, configure information for the clients that are permitted to access LUNs in the target group. After all desired changes have been made to the group, click *Next*.
For option information, see *Step 7* in *Section 15.5, “Setting Up an iSCSI LIO Target Group”*.
9. Click *Finish* to save and apply the settings.

15.7 Deleting an iSCSI LIO Target Group

Deleting an iSCSI LIO target group removes the definition of the group, and the related setup for clients, including LUN mappings and authentication credentials. It does not destroy the data on the partitions. To give clients access again, you can allocate the target LUNs to a different or new target group, and configure the client access for them.

1. Log in to the iSCSI LIO target server as the root user, then launch a terminal console.
2. Ensure that the /etc/init.d/target daemon is running. At the command prompt, enter

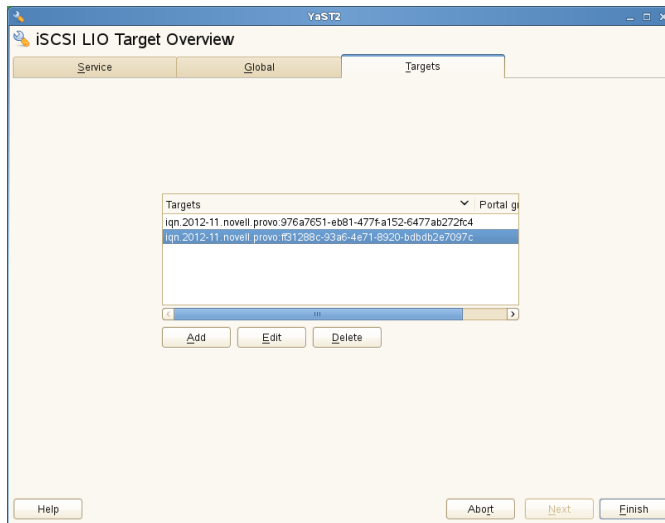
```
systemctl start target.service
```

The command returns a message to confirm that the daemon is started, or that the daemon is already running.

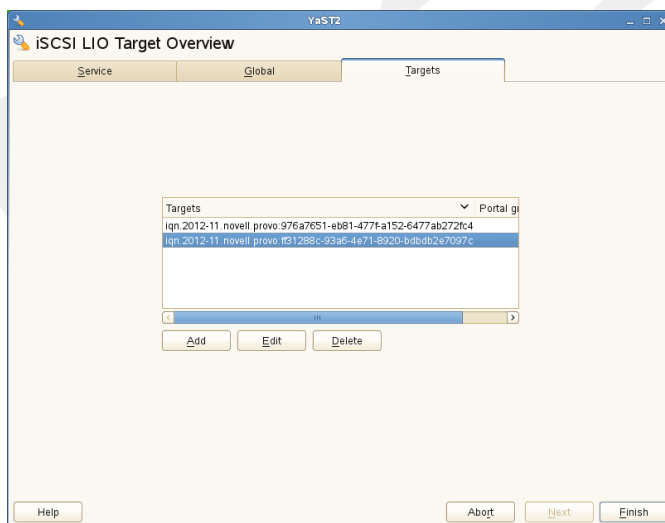
3. Launch YaST as the root user.
4. In the YaST Control Center, select *Network Services*, then select *iSCSI LIO Target*.
You can also search for “lio”, then select *iSCSI LIO Target*.



5. In the iSCSI LIO Target Overview dialog box, select the *Targets* tab to view a list of configured target groups.



6. Select the iSCSI LIO target group to be deleted, then click *Delete*.



7. When you are prompted, click *Continue* to confirm the deletion, or click *Cancel* to cancel it.
8. Click *Finish* to save and apply the settings.

15.8 Troubleshooting iSCSI LIO Target Server

This section describes some known issues and possible solutions for iSCSI LIO Target Server.

- [Section 15.8.1, “Portal Error When Setting Up Target LUNs”](#)
- [Section 15.8.2, “iSCSI LIO Targets Are Not Visible from Other Computers”](#)

15.8.1 Portal Error When Setting Up Target LUNs

When adding or editing an iSCSI LIO target group, you get an error:

```
Problem setting network portal <ip_address>:3260
```

The `/var/log/YaST2/y2log` log file contains the following error:

```
find: `/sys/kernel/config/target/iscsi': No such file or directory
```

This problem occurs if the iSCSI LIO Target Server software is not currently running. To resolve this issue, exit YaST, manually start iSCSI LIO at the command line, then try again.

1. Open a terminal console as the root user.
2. At the command prompt, enter

```
systemctl start target.service
```

You can also enter the following to check if `configfs`, `iscsi_target_mod`, and `target_core_mod` are loaded. A sample response is shown.

```
lsmod | grep iscsi
```

```
iscsi_target_mod      295015  0
target_core_mod       346745  4
iscsi_target_mod,target_core_pscsi,target_core_iblock,target_core_file
configfs              35817  3 iscsi_target_mod,target_core_mod
scsi_mod              231620  16
```

```
iscsi_target_mod,target_core_pscsi,target_core_mod,sg,sr_mod,mptctl,sd_mod,  
scsi_dh_rdac,scsi_dh_emc,scsi_dh_alua,scsi_dh_hp_sw,scsi_dh,libata,mptspi,  
mptscsih,scsi_transport_spi
```

15.8.2 iSCSI LIO Targets Are Not Visible from Other Computers

If you use a firewall on the target server, you must open the iSCSI port that you are using to allow other computers to see the iSCSI LIO targets. For information, see *Step 7* in *Section 15.2.1*, “Configuring iSCSI LIO Startup Preferences”.

15.9 iSCSI LIO Target Terminology

backstore

A physical storage object that provides the actual storage underlying an iSCSI endpoint.

CDB (command descriptor block)

The standard format for SCSI commands. CDBs are commonly 6, 10, or 12 bytes long, though they can be 16 bytes or of variable length.

CHAP (Challenge Handshake Authentication Protocol)

A point-to-point protocol (PPP) authentication method used to confirm the identity of one computer to another. After the Link Control Protocol (LCP) connects the two computers, and the CHAP method is negotiated, the authenticator sends a random Challenge to the peer. The peer issues a cryptographically hashed Response that depends upon the Challenge and a secret key. The authenticator verifies the hashed Response against its own calculation of the expected hash value, and either acknowledges the authentication or terminates the connection. CHAP is defined in the [Internet Engineering Task Force \(IETF\)](http://www.ietf.org) (<http://www.ietf.org>) [RFC 1994](#).

CID (connection identifier)

A 16-bit number, generated by the initiator, that uniquely identifies a connection between two iSCSI devices. This number is presented during the login phase.

endpoint

The combination of an iSCSI Target Name with an iSCSI TPG (IQN + Tag).

EUI (extended unique identifier)

A 64-bit number that uniquely identifies every device in the world. The format consists of 24 bits that are unique to a given company, and 40 bits assigned by the company to each device it builds.

initiator

The originating end of a SCSI session. Typically a controlling device such as a computer.

IPS (Internet Protocol storage)

The class of protocols or devices that use the IP protocol to move data in a storage network. FCIP (Fibre Channel over Internet Protocol), iFCP (Internet Fibre Channel Protocol), and iSCSI (Internet SCSI) are all examples of IPS protocols.

IQN (iSCSI qualified name)

A name format for iSCSI that uniquely identifies every device in the world (for example: iqn.5886.com.acme.tapedrive.sn-a12345678).

ISID (initiator session identifier)

A 48-bit number, generated by the initiator, that uniquely identifies a session between the initiator and the Target. This value is created during the login process, and is sent to the target with a Login PDU.

MCS (multiple connections per session)

A part of the iSCSI specification that allows multiple TCP/IP connections between an initiator and a target.

MPIO (multipath I/O)

A method by which data can take multiple redundant paths between a server and storage.

network portal

The combination of an iSCSI Endpoint with an IP address plus a TCP (Transmission Control Protocol) port. TCP port 3260 is the port number for the iSCSI protocol, as defined by IANA (Internet Assigned Numbers Authority).

SAM (SCSI architectural model)

A document that describes the behavior of SCSI in general terms, allowing for different types of devices communicating over various media.

target

The receiving end of a SCSI session, typically a device such as a disk drive, tape drive, or scanner.

target group (TG)

A list of SCSI target ports that are all treated the same when creating views. Creating a view can help facilitate LUN (logical unit number) mapping. Each view entry specifies a target group, host group, and a LUN.

target port

The combination of an iSCSI endpoint with one or more LUNs.

target port group (TPG)

A list of IP addresses and TCP port numbers that determines which interfaces a specific iSCSI target will listen to.

target session identifier (TSID)

A 16-bit number, generated by the target, that uniquely identifies a session between the initiator and the target. This value is created during the login process, and is sent to the initiator with a Login Response PDU (protocol data units).

16 Fibre Channel Storage over Ethernet Networks: FCoE

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.

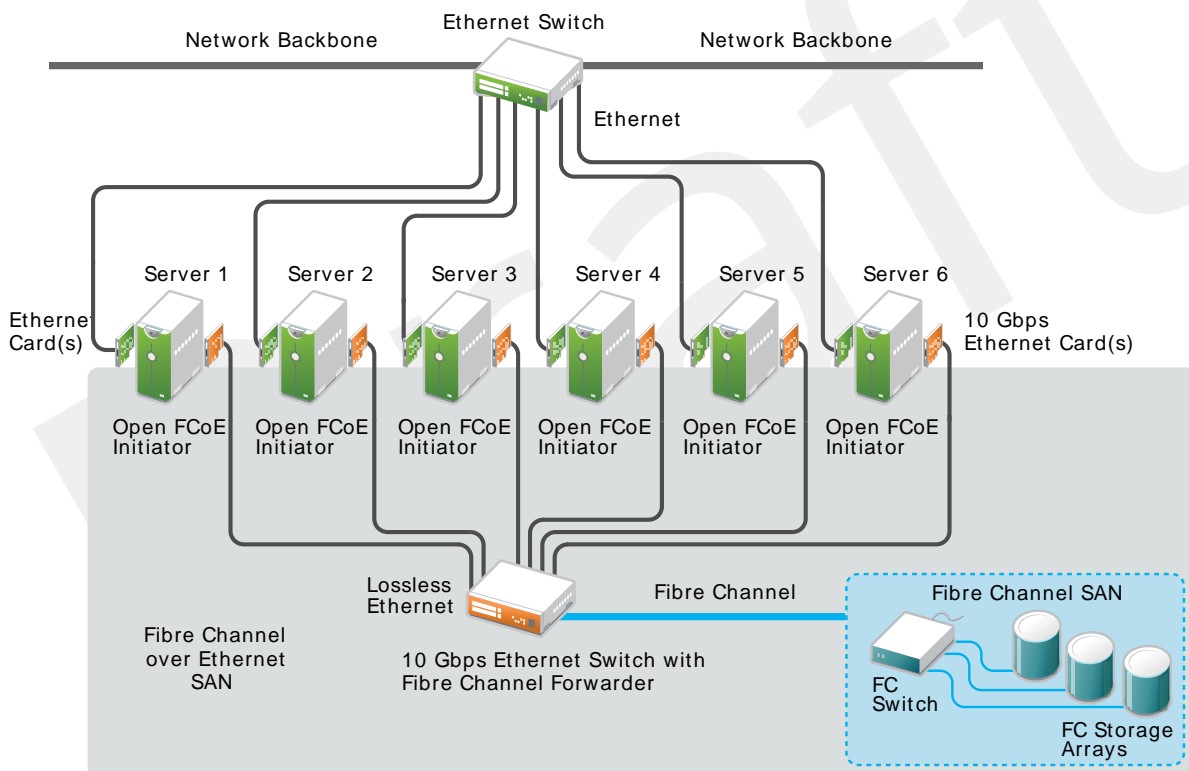


FIGURE 16.1: OPEN FIBRE CHANNEL OVER ETHERNET SAN

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 16.1, "Installing FCoE and the YaST FCoE Client"*
- *Section 16.2, "Configuring FCoE Interfaces during the Installation"*
- *Section 16.3, "Managing FCoE Services with YaST"*
- *Section 16.4, "Configuring FCoE with Commands"*
- *Section 16.5, "Managing FCoE Instances with the FCoE Administration Tool"*
- *Section 16.6, "Setting Up Partitions for an FCoE Initiator Disk"*
- *Section 16.7, "Creating a File System on an FCoE Initiator Disk"*
- *Section 16.8, "Additional Information"*

16.1 Installing FCoE and the YaST FCoE Client

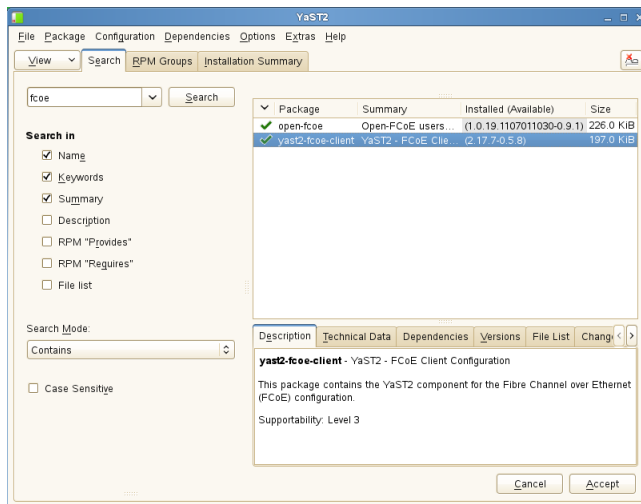
You can set up FCoE disks in your storage infrastructure by enabling FCoE at the switch for the connections to a server. If FCoE disks are available when the SUSE Linux Enterprise Server operating system is installed, the FCoE Initiator software is automatically installed at that time. If the FCoE Initiator software and YaST 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 YaST, 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.

16.2 Configuring FCoE Interfaces during the Installation

The YaST 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 YaST 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
```

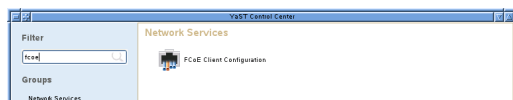
When the FCoE disks are detected, the YaST installation offers the option to configure FCoE instances at that time. On the Disk Activation page, select *Configure FCoE Interfaces* to access the FCoE configuration. For information about configuring the FCoE interfaces, see [Section 16.3](#), “Managing FCoE Services with YaST”.



16.3 Managing FCoE Services with YaST

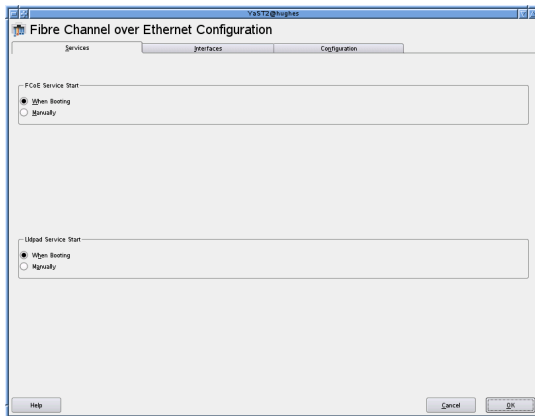
You can use the YaST FCoE Client Configuration option to create, configure, and remove FCoE interfaces for the FCoE disks in your Fibre Channel storage infrastructure. To use this option, the FCoE Initiator service (the `fcoemon` daemon) and the Link Layer Discovery Protocol agent daemon (`lldpad`) must 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 YaST.
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.



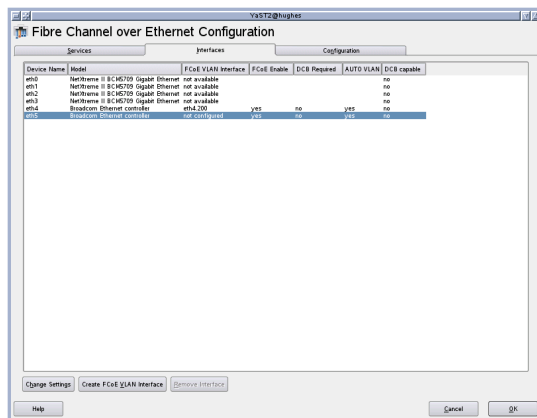
- **FCoE Service Start:** Specifies whether to start the Fibre Channel over Ethernet service **fcoemon** daemon at the server boot time or manually. The daemon controls the FCoE interfaces and establishes a connection with the **lldpad** daemon. The values are *When Booting* (default) or *Manually*.
- **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.

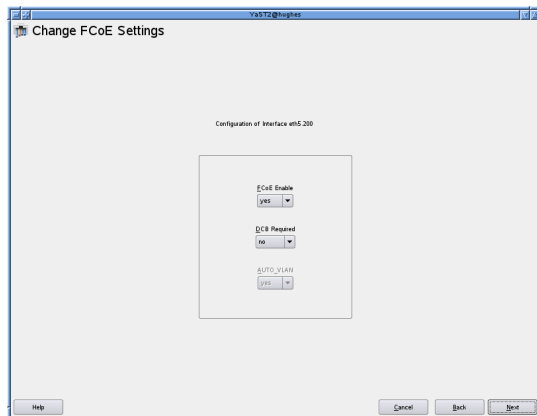
The *FCoE Interfaces* table displays the following information about each adapter:



- **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.

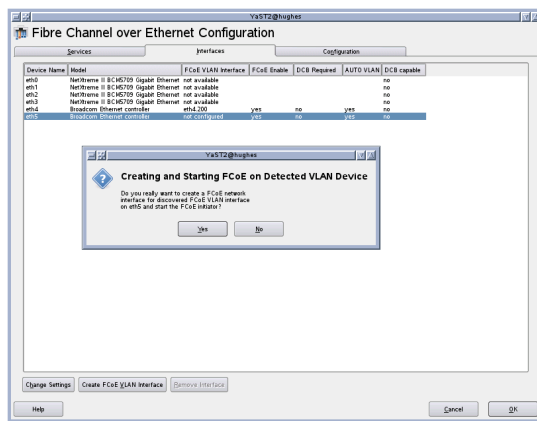


- **FCoE Enable:** Enable or disable the creation of FCoE instances for the adapter. Values are *yes* or *no*.
- **DCB Required:** Specifies whether Data Center Bridging is required for the adapter. Values are *yes* (default) or *no*. DCB is usually required.
- **Auto VLAN:** Specifies whether the `fcoemon` 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.

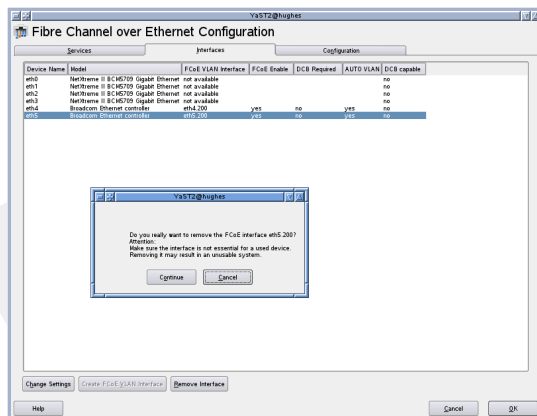
Create FCoE VLAN Interfaces.

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`.

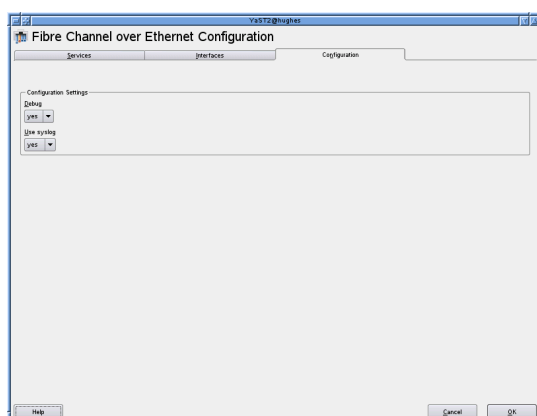


Remove FCoE Interface.

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*.



- 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*.

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.

16.4 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).

```
rclldpad 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.

```
systemctl enable lldpad.service fcoe.service
```

16.5 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 \(http://open-fcoe.org/\)](http://open-fcoe.org/).

Syntax

Fiber 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)

16.6 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.
```

16.7 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.
```

16.8 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/quick-start>) .

17 LVM Volume Snapshots

A Logical Volume Manager (LVM) logical volume 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 17.1, "Understanding Volume Snapshots"*
- *Section 17.2, "Creating Linux Snapshots with LVM"*
- *Section 17.3, "Monitoring a Snapshot"*
- *Section 17.4, "Deleting Linux Snapshots"*
- *Section 17.5, "Using Snapshots for Virtual Machines on a Virtual Host"*
- *Section 17.6, "Merging a Snapshot with the Source Logical Volume to Revert Changes or Roll Back to a Previous State"*

17.1 Understanding Volume Snapshots

A file system snapshot contains metadata about and data blocks from a source logical volume that have changed since the snapshot was taken. When you access data via the snapshot, you see a point-in-time copy of the source logical volume. There is no need to restore data from backup media or to overwrite the changed data.



Important

During the snapshot's lifetime, the snapshot must be mounted before its source logical volume can be mounted.

LVM volume 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 source logical 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 source block. As more blocks change from their original value on the source logical volume, the snapshot size grows.

When you are sizing the snapshot, consider how much data is expected to change on the source logical 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 source logical 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 source logical volume will change at least one time before you delete the snapshot, then the snapshot volume should be at least as large as the source logical 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. In LVM2, snapshots are read/write by default. When you write data directly to the snapshot, that block is marked in the exception table as used, and never gets copied from the source logical volume. You can mount the snapshot volume, and test application changes by writing data directly to the snapshot volume. You can easily discard the changes by dismounting the snapshot, removing the snapshot, and then re-mounting the source logical volume.

In a virtual guest environment, you can use the snapshot function for LVM logical volumes you create on the server's disks, as you would on a physical server.

In a virtual host environment, you can use the snapshot function to back up the virtual machine's storage back-end, or to test changes to a virtual machine image, such as for patches or upgrades, without modifying the source logical volume. The virtual machine must be using an LVM logical volume as its storage back-end, as opposed to using a virtual disk file. You can mount the LVM logical volume and use it to store the virtual machine image as a file-backed disk, or you can assign the LVM logical volume as a physical disk to write to it as a block device.

Beginning in SLES 11 SP3, an LVM logical volume snapshots can be thinly provisioned. Thin provisioning is assumed if you to create a snapshot without a specified size. The snapshot is created as a thin volume that uses space as needed from a thin pool. A thin snapshot volume has the same characteristics as any other thin volume. You can independently activate the volume, extend the volume, rename the volume, remove the volume, and even snapshot the volume.

! Important

To use thinly provisioned snapshots in a cluster, the source logical volume and its snapshots must be managed in a single cluster resource. This allows the volume and its snapshots to always be mounted exclusively on the same node.

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 source logical volume. When the snapshot is full, it is disabled, which prevents you from remounting the source logical volume. If you create multiple snapshots for a source logical volume, remove the snapshots in a last created, first deleted order.

17.2 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 <size>] -n snap_volume source_volume_path
```

If no size is specified, the snapshot is created as a thin snapshot.

For example:

```
lvcreate -s -L 1G -n linux01-snap /dev/lvm/linux01
```

The snapshot is created as the /dev/lvm/linux01-snap volume.

17.3 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
```

17.4 Deleting Linux Snapshots

- Open a terminal console, log in as the root user, then enter

```
lvremove snap_volume_path
```

For example:

```
lvremove /dev/lvmvg/linux01-snap
```

17.5 Using Snapshots for Virtual Machines on a Virtual Host

Using an LVM logical volume for a virtual machine's back-end storage allows flexibility in administering the underlying device, such as making it easier to move storage objects, create snapshots, and back up data. You can mount the LVM logical volume and use it to store the virtual machine image as a file-backed disk, or you can assign the LVM logical volume as a physical disk to write to it as a block device. You can create a virtual disk image on the LVM logical volume, then snapshot the LVM.

You can leverage the read/write capability of the snapshot to create different instances of a virtual machine, where the changes are made to the snapshot for a particular virtual machine instance. You can create a virtual disk image on an LVM logical volume, snapshot the source logical volume, and modify the snapshot for a particular virtual machine instance. You can create another snapshot of the source logical volume, and modify it for a different virtual machine instance. The majority of the data for the different virtual machine instances resides with the image on the source logical volume.

You can also leverage the read/write capability of the snapshot to preserve the virtual disk image while testing patches or upgrades in the guest environment. You create a snapshot of the LVM volume that contains the image, and then run the virtual machine on the snapshot location. The source logical volume is unchanged, and all changes for that machine are written to the snapshot. To return to the source logical volume of the virtual machine image, you power off the virtual machine, then remove the snapshot from the source logical volume. To start over, you re-create the snapshot, mount the snapshot, and restart the virtual machine on the snapshot image.

Important

The following procedure uses a file-backed virtual disk image and the Xen hypervisor. You can adapt the procedure in this section for other hypervisors that run on the SUSE Linux platform, such as KVM.

To run a file-backed virtual machine image from the snapshot volume:

1. Ensure that the source logical volume that contains the file-backed virtual machine image is mounted, such as at mount point `/var/lib/xen/images/<image_name>`.
2. Create a snapshot of the LVM logical volume with enough space to store the differences that you expect.

```
lvcreate -s -L 20G -n myvm-snap /dev/lvmvg/myvm
```

If no size is specified, the snapshot is created as a thin snapshot.

3. Create a mount point where you will mount the snapshot volume.

```
mkdir -p /mnt/xen/vm/myvm-snap
```

4. Mount the snapshot volume at the mount point you created.

```
mount -t auto /dev/lvmvg/myvm-snap /mnt/xen/vm/myvm-snap
```

5. In a text editor, copy the configuration file for the source virtual machine, modify the paths to point to the file-backed image file on the mounted snapshot volume, and save the file such as `/etc/xen/myvm-snap.cfg`.
6. Start the virtual machine using the mounted snapshot volume of the virtual machine.

```
xm create -c /etc/xen/myvm-snap.cfg
```

7. (Optional) Remove the snapshot, and use the unchanged virtual machine image on the source logical volume.

```
umount /mnt/xenvms/myvm-snap  
lvremove -f /dev/lvmvg/mylvm-snap
```

8. (Optional) Repeat this process as desired.

17.6 Merging a Snapshot with the Source Logical Volume to Revert Changes or Roll Back to a Previous State

Snapshots can be useful if you need to roll back or restore data on a volume to a previous state. For example, you might need to revert data changes that resulted from an administrator error or a failed or undesirable package installation or upgrade.

You can use the **`lvconvert --merge`** command to revert the changes made to an LVM logical volume. The merging begins as follows:

- If both the source logical volume and snapshot volume are not open, the merge begins immediately.
- If the source logical volume or snapshot volume are open, the merge starts the first time either the source logical volume or snapshot volume are activated and both are closed.
- If the source logical volume cannot be closed, such as the `root` file system, the merge is deferred until the next time the server reboots and the source logical volume is activated.
- If the source logical volume contains a virtual machine image, you must shut down the virtual machine, deactivate the source logical volume and snapshot volume (by dismounting them in that order), and then issue the merge command. Because the source logical volume is automatically remounted and the snapshot volume is deleted when the merge is complete, you should not restart the virtual machine until after the merge is complete. After the merge is complete, you use the resulting logical volume for the virtual machine.

After a merge begins, the merge continues automatically after server restarts until it is complete. A new snapshot cannot be created for the source logical volume while a merge is in progress.

While the merge is in progress, reads or writes to the source logical volume are transparently redirected to the snapshot that is being merged. This allows users to immediately view and access the data as it was when the snapshot was created. They do not need to wait for the merge to complete.

When the merge is complete, the source logical volume contains the same data as it did when the snapshot was taken, plus any data changes made after the merge began. The resulting logical volume has the source logical volume's name, minor number, and UUID. The source logical volume is automatically remounted, and the snapshot volume is removed.

1. Open a terminal console, log in as the root user, then enter

```
lvconvert --merge [-b] [-i <seconds>] [<snap_volume_path>[...<snapN>]]  
@<volume_tag>]
```

You can specify one or multiple snapshots on the command line. You can alternatively tag multiple source logical volumes with the same volume tag then specify @<volume_tag> on the command line. The snapshots for the tagged volumes are merged to their respective source logical volumes. For information about tagging logical volumes, see [Section 4.7, “Tagging LVM2 Storage Objects”](#).

The options include:

-b,

--background

Run the daemon in the background. This allows multiple specified snapshots to be merged concurrently in parallel.

-i,

--interval <seconds>

Report progress as a percentage at regular intervals. Specify the interval in seconds.

For more information about this command, see the **lvconvert(8)** man page.

For example:

```
lvconvert --merge /dev/lvmvg/linux01-snap
```

This command merges /dev/lvmvg/linux01-snap into its source logical volume.

```
lvconvert --merge @mytag
```

If lv011, lv012, and lv013 are all tagged with mytag, each snapshot volume is merged serially with its respective source logical volume; that is: lv011, then lv012, then lv013. If the --background option is specified, the snapshots for the respective tagged logical volume are merged concurrently in parallel.

2. (Optional) If both the source logical volume and snapshot volume are open and they can be closed, you can manually deactivate and activate the source logical volume to get the merge to start immediately.

```
umount <original_volume>
lvchange -an <original_volume>
lvchange -ay <original_volume>
mount <original_volume> <mount_point>
```

For example:

```
umount /dev/lvmvg/lvol01
lvchange -an /dev/lvmvg/lvol01
lvchange -ay /dev/lvmvg/lvol01
mount /dev/lvmvg/lvol01 /mnt/lvol01
```

3. (Optional) If both the source logical volume and snapshot volume are open and the source logical volume cannot be closed, such as the root file system, you can restart the server and mount the source logical volume to get the merge to start immediately after the restart.

18 Managing Access Control Lists over NFSv4

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 file system 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 file system, 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 will affect programs running on the client in that some particular combinations of Access Control Entries (ACEs) might not be possible.

It is not supported to mount NFS volumes locally on the exporting NFS server.

Additional Information

For information, see “Introduction to NFSv4 ACLs” on the Linux-nfs.org Web site (http://wiki.linux-nfs.org/wiki/index.php/ACLs#Introduction_to_NFSv4_ACLs) ↗.

19 Troubleshooting Storage Issues

This section describes how to work around known issues for devices, software RAIDs, multipath I/O, and volumes.

- [Section 19.1, “Is DM-MPIO Available for the Boot Partition?”](#)
- [Section 19.2, “Btrfs Error: No space is left on device”](#)
- [Section 19.3, “Issues for Multipath I/O”](#)
- [Section 19.4, “Issues for Software RAIDs”](#)
- [Section 19.5, “Issues for iSCSI”](#)
- [Section 19.6, “Issues for iSCSI LIO Target”](#)

19.1 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. For information, see [Section 7.12, “Configuring Multipath I/O for the Root Device”](#).

19.2 Btrfs Error: No space is left on device

The root (`/`) partition using the Btrfs file system stops accepting data. You receive the error “No space left on device”.

See the following sections for information about possible causes and prevention of this issue.

- [Section 19.2.1, “Disk Space Consumed by Snapper Snapshots”](#)
- [Section 19.2.2, “Disk Space Consumed by Log, Crash, and Cache Files”](#)

19.2.1 Disk Space Consumed by Snapper Snapshots

If Snapper is running for the Btrfs file system, the “No space left on device” problem is typically caused by having too much data stored as snapshots on your system.

You can remove some snapshots from Snapper, however, the snapshots are not deleted immediately and might not free up as much space as you need.

To delete files from Snapper:

1. Log in as the root user, then open a terminal console.
2. Gain back enough space for the system to come up.

- a. At the command prompt, enter

```
btrfs filesystem show
```

```
Label: none uuid: 40123456-cb2c-4678-8b3d-d014d1c78c78
Total devices 1 FS bytes used 20.00GB
devid 1 size 20.00GB used 20.00GB path /dev/sda3
```

- b. Enter

```
btrfs fi balance start </mountpoint> -usage=5
```

This command attempts to relocate data in empty or near-empty data chunks, allowing the space to be reclaimed and reassigned to metadata. This can take awhile (many hours for 1 TB) although the system is otherwise usable during this time.

3. List the snapshots in Snapper. Enter

```
snapper -c root list
```

4. Delete one or more snapshots from Snapper. Enter

```
snapper -c root delete #
```

Ensure that you delete the oldest snapshots first. The older a snapshot is, the more disk space it occupies.

To help prevent this problem, you can change the Snapper cleanup defaults to be more aggressive in the `/etc/snapper/configs/root` configuration file, or for other mount points. Snapper provides three algorithms to clean up old snapshots. The algorithms are executed in a daily cron-job. The cleanup frequency is defined in the Snapper configuration for the mount point. Lower the `TIMELINE_LIMIT` parameters for daily, monthly, and yearly to reduce how long and the number of snapshots to be retained. For information, see `##### 4 ##### Snapper ##### 4.4 ##### Snapper ###`

If you use Snapper with Btrfs on the file system disk, it is advisable to reserve twice the amount of disk space than the standard storage proposal. The YaST Partitioner automatically proposes twice the standard disk space in the Btrfs storage proposal for the root file system.

19.2.2 Disk Space Consumed by Log, Crash, and Cache Files

If the system disk is filling up with data, you can try deleting files from `/var/log`, `/var/crash`, and `/var/cache`.

The Btrfs `root` file system subvolumes `/var/log`, `/var/crash` and `/var/cache` can use all of the available disk space during normal operation, and cause a system malfunction. To help avoid this situation, SUSE Linux Enterprise Server offers Btrfs quota support for subvolumes. See the `btrfs(8)` manual page for more details.

19.3 Issues for Multipath I/O

See *Section 7.19, "Troubleshooting MPIO"*.

19.4 Issues for Software RAIDs

See *Section 8.3, "Troubleshooting Software RAIDs"*.

19.5 Issues for iSCSI

See *Section 14.5, "Troubleshooting iSCSI"*.

19.6 Issues for iSCSI LIO Target

See *Section 15.8, “Troubleshooting iSCSI LIO Target Server”*.

Draft

A 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”*
- *Section A.2, “GNU Free Documentation License”*

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, "November 4, 2013"*
- *Section B.2, "October 14, 2013"*
- *Section B.3, "October 4, 2013"*
- *Section B.4, "June 2013 (SLES 11 SP3)"*
- *Section B.5, "March 19, 2013"*
- *Section B.6, "March 11, 2013"*
- *Section B.7, "February 8, 2013"*
- *Section B.8, "January 8, 2013"*
- *Section B.9, "November 14, 2012"*
- *Section B.10, "October 29, 2012"*
- *Section B.11, "October 19, 2012"*
- *Section B.12, "September 28, 2012"*
- *Section B.13, "April 12, 2012"*
- *Section B.14, "February 27, 2012 (SLES 11 SP2)"*
- *Section B.15, "July 12, 2011"*
- *Section B.16, "June 14, 2011"*
- *Section B.17, "May 5, 2011"*
- *Section B.18, "January 2011"*
- *Section B.19, "September 16, 2010"*
- *Section B.20, "June 21, 2010"*
- *Section B.21, "May 2010 (SLES 11 SP1)"*

- *Section B.22, "February 23, 2010"*
- *Section B.23, "December 1, 2009"*
- *Section B.24, "October 20, 2009"*
- *Section B.25, "August 3, 2009"*
- *Section B.26, "June 22, 2009"*
- *Section B.27, "May 21, 2009"*

B.1 November 4, 2013

Updates were made to the following section. The changes are explained below.

B.1.1 Managing Multipath I/O for Devices

Updates were made to the following section. The changes are explained below.

Location	Change
<i>Section 7.4.2, "Partitioning Multipath Devices"</i>	Added issues related to partitioning multipath devices.

B.2 October 14, 2013

Updates were made to the following section. The changes are explained below.

B.2.1 iSNS for Linux

Location	Change
<i>Section 13.3.3, "Adding iSCSI Nodes to a Discovery Domain"</i>	Added the -p ip:port option to the following commands: <div></div>

Location	Change
	<pre>iscsiadm -m node -t iqn.2006-02.com.example.iserv:systems -p ip:port --op=update -- name=node.startup --value=automatic iscsiadm -m node -t iqn.2006-02.com.example.iserv:systems -p ip:port --op=delete</pre>

B.3 October 4, 2013

Updates were made to the following sections. The changes are explained below.

- *Section B.3.1, “Managing Multipath I/O for Devices”*
- *Section B.3.2, “Volume Snapshots”*
- *Section B.3.3, “What’s New for Storage in SLES 11”*

B.3.1 Managing Multipath I/O for Devices

Location	Change
<i>blacklist_exceptions</i>	Added a link to <u>blacklist_exceptions</u> examples in <i>Section 7.8, “Blacklisting Non-Multipath Devices”</i> .
<i>Section 7.8, “Blacklisting Non-Multipath Devices”</i>	Added examples for using the <u>blacklist_exceptions</u> section to enable multipath for devices blacklisted by the regular expressions used in the <u>blacklist</u> section.

B.3.2 Volume Snapshots

Location	Change
<i>Chapter 17, LVM Volume Snapshots</i>	For clarity, changed “original volume” to “source logical volume”.
<i>Section 17.1, “Understanding Volume Snapshots”</i>	In a Xen host environment, you can use the snapshot function to back up the virtual machine’s storage back-end or to test changes to a virtual machine image such as for patches or upgrades.
<i>Section 17.5, “Using Snapshots for Virtual Machines on a Virtual Host”</i>	This section is new.
<i>Section 17.6, “Merging a Snapshot with the Source Logical Volume to Revert Changes or Roll Back to a Previous State”</i>	If the source logical volume contains a virtual machine image, you must shut down the virtual machine, deactivate the source logical volume and snapshot volume (by dismounting them in that order), issue the merge command, and then activate the snapshot volume and source logical volume (by mounting them in that order). Because the source logical volume is automatically remounted and the snapshot volume is deleted when the merge is complete, you should not restart the virtual machine until after the merge is complete. After the merge is complete, you use the resulting logical volume for the virtual machine.

B.3.3 What's New for Storage in SLES 11

Location	Change
<i>Section 2.4, "What's New in SLES 11"</i>	Added information about how to prepare for a SLES 10 SP4 to SLES 11 upgrade if the system device is managed by EVMS.

B.4 June 2013 (SLES 11 SP3)

Updates were made to the following sections. The changes are explained below.

- *Section B.4.1, "LVM Configuration"*
- *Section B.4.2, "Mass Storage over IP Networks: iSCSI LIO Target Server"*
- *Section B.4.3, "Managing Multipath I/O for Devices"*
- *Section B.4.4, "Managing Software RAIDs 6 and 10 with mdadm"*
- *Section B.4.5, "Overview of File Systems in Linux"*
- *Section B.4.6, "Storage Enclosure LED Utilities for Software RAIDs"*
- *Section B.4.7, "Volume Snapshots"*
- *Section B.4.8, "Troubleshooting Storage Issues"*
- *Section B.4.9, "What's New for Storage in SLES 11"*

B.4.1 LVM Configuration


Location	Change
<i>Section 4.5, "Configuring Logical Volumes"</i>	Added information about thin provisioning of LVM logical volumes.

B.4.2 Mass Storage over IP Networks: iSCSI LIO Target Server

Chapter 15, Mass Storage over IP Networks: iSCSI LIO Target Server is new.

B.4.3 Managing Multipath I/O for Devices

Location	Change
<i>Section 7.2.6, "Using Multipath with NetApp Devices"</i>	This section is new.
<i>Section 7.2.11.1, "Storage Arrays That Are Automatically Detected for Multipathing"</i>	Updated the list of storage arrays that have a default definition in the <code>/usr/share/doc/packages/multipath-tools/multipath.conf.defaults</code> file.
<i>Section 7.3.4, "Linux multipath(8) Command"</i>	The <code>-r</code> option is new.
<i>Section 7.3.5, "Linux mpathpersist(8) Utility"</i>	This section is new.
<i>Section 7.4.3, "Configuring the Device Drivers in initrd for Multipathing"</i>	Added information about the SCSI hardware handlers for <code>alua</code> , <code>rdac</code> , <code>hp-sw</code> , and <code>emc</code> .
<i>Section 7.7, "Configuring Default Policies for Polling, Queueing, and Failback"</i>	Updated the default values list to annotate the deprecated attributes <code>udev_dir</code> and <code>getuid_callout</code> , the new attribute <code>uid_attribute</code> , and the changed default values for <code>path_selector</code> and <code>max_fds</code> .
<i>Section 7.11.2.1, "Understanding Priority Groups and Attributes"</i>	<p>The default for <code>path_selector</code> changed from <code>round-robin</code> to <code>service-time</code>.</p> <p>The <code>getuid_callout</code> attribute is deprecated and replaced by the <code>uid_attribute</code> parameter.</p> <p>Added <code>uid_attribute</code>.</p>

Location	Change
<p><i>Section 7.14, "Scanning for New Devices without Rebooting"</i></p> <p><i>Section 7.15, "Scanning for New Partitioned Devices without Rebooting"</i></p>	<p> Warning</p> <p>In EMC PowerPath environments, do not use the <code>rescan-scsi-bus.sh</code> utility provided with the operating system or the HBA vendor scripts for scanning the SCSI buses. To avoid potential file system corruption, EMC requires that you follow the procedure provided in the vendor documentation for EMC PowerPath for Linux.</p>

B.4.4 Managing Software RAID6 and 10 with mdadm

Location	Change
<p><i>Section 10.3.3, "Creating a Complex RAID10 with the YaST Partitioner"</i></p>	<p>This section is new.</p>

B.4.5 Overview of File Systems in Linux

Location	Change
<p><i>Section 1.2.1.4, "Snapshots for the Root File System"</i></p>	<p>The cleanup frequency is defined in the Snapper configuration for the mount point. Lower the <code>TIMELINE_LIMIT</code> parameters for daily, monthly, and yearly to reduce how long and the number of snapshots to be retained.</p>

Location	Change
<i>Section 1.2.1.9, "Btrfs Quota Support for Subvolumes"</i>	This section is new.

B.4.6 Storage Enclosure LED Utilities for Software RAID5

Chapter 12, Storage Enclosure LED Utilities for MD Software RAID5 is new.

B.4.7 Volume Snapshots

Location	Change
<i>Section 17.2, "Creating Linux Snapshots with LVM"</i>	Added information about thin provisioning for LVM logical volume snapshots.
<i>Section 17.6, "Merging a Snapshot with the Source Logical Volume to Revert Changes or Roll Back to a Previous State"</i>	This section is new.

B.4.8 Troubleshooting Storage Issues

Location	Change
<i>Section 19.2, "Btrfs Error: No space is left on device"</i>	This section is new.

B.4.9 What's New for Storage in SLES 11

Location	Change
<i>Section 2.1, "What's New in SLES 11 SP3"</i>	This section is new.

B.5 March 19, 2013

Updates were made to the following sections. The changes are explained below.

- *Section B.5.1, "Overview of File Systems in Linux"*

B.5.1 Overview of File Systems in Linux

Location	Change
<i>Section 1.2.3.4, "Ext3 File System Inode Size and Number of Inodes"</i>	This section was updated to discuss changes to the default settings for inode size and bytes-per-inode ratio in SLES 11.

B.6 March 11, 2013

Updates were made to the following sections. The changes are explained below.

- *Section B.6.1, "LVM Configuration"*
- *Section B.6.2, "Overview of File Systems in Linux"*

B.6.1 LVM Configuration

Location	Change
<i>Section 4.6, "Automatically Activating Non-Root LVM Volume Groups"</i>	This section is new.

B.6.2 Overview of File Systems in Linux

Location	Change
<i>Section 1.2.1, "Btrfs"</i>	Multiple device support that allows you to grow or shrink the file system. The feature is planned to be available in a future release of the YaST Partitioner.

B.7 February 8, 2013

Updates were made to the following sections. The changes are explained below.

- *Section B.7.1, "Configuring Software RAID1 for the Root Partition"*

B.7.1 Configuring Software RAID1 for the Root Partition

This section has been modified to focus on the software RAID1 type. Software RAID0 and RAID5 are not supported. They were previously included in error. Additional important changes are noted below.


Location	Change
<i>Section 9.1, "Prerequisites for Using a Software RAID1 Device for the Root Partition"</i>	You need a third device to use for the <u>/boot</u> partition. The devices should be a local device.
<i>Step 4 in Section 9.4, "Creating a Software RAID1 Device for the Root (/) Partition"</i>	Create the <u>/boot</u> partition. The devices should be a local device.
<i>Step 7.b in Section 9.4, "Creating a Software RAID1 Device for the Root (/) Partition"</i>	Under <i>RAID Type</i> , select <i>RAID 1 (Mirroring)</i> .

B.8 January 8, 2013

Updates were made to the following sections. The changes are explained below.

- [Section B.8.1, “LVM Configuration”](#)
- [Section B.8.2, “Managing Multipath I/O for Devices”](#)

B.8.1 LVM Configuration

Location	Change
Section 4.1, “Understanding the Logical Volume Manager”	 Important If you add multipath support after you have configured LVM, you must modify the <code>/etc/lvm/lvm.conf</code> file to scan only the multipath device names in the <code>/dev/disk/by-id</code> directory as described in Section 7.2.4, “Using LVM2 on Multipath Devices” , then reboot the server.

B.8.2 Managing Multipath I/O for Devices

Location	Change
Section 7.2.1.5, “High-Availability Solutions”	Ensure that the configuration settings in the <code>/etc/multipath.conf</code> file on each node are consistent across the cluster.
Section 7.2.3, “Using WWID, User-Friendly, and Alias Names for Multipathed Devices”	When using links to multipath-mapped devices in the <code>/dev/disk/by-id</code> directory, ensure that you specify the <code>dm-uuid*</code> name or alias name, and not a fixed path instance of the device.

Location	Change
<i>Section 7.2.4, "Using LVM2 on Multipath Devices"</i>	<p>To accept both raw disks and partitions for Device Mapper names, specify the path as follows, with no hyphen (-) before <u>mpath</u>:</p> <pre>filter = ["a /dev/disk/by-id/dm- uuid-.*mpath-.* ", "r .* "]</pre>
<i>Section 7.2.9, "Partitioning Multipath Devices"</i>	<p>Ensure that the configuration files for <u>lvm.conf</u> and <u>md.conf</u> point to the multipath-device names. This should happen automatically if <u>boot.multipath</u> is enabled and loads before <u>boot.lvm</u> and <u>boot.md</u>. Otherwise, the LVM and MD configuration files might contain fixed paths for multipath-devices, and you must correct those paths to use the multipath-device names.</p>
<i>Section 7.9, "Configuring User-Friendly Names or Alias Names"</i>	<p>Before you begin, review the requirements in <i>Section 7.2.3, "Using WWID, User-Friendly, and Alias Names for Multipathed Devices"</i>.</p>

B.9 November 14, 2012

Updates were made to the following section. The changes are explained below.

- *Section B.9.1, “Overview of File Systems in Linux”*

B.9.1 Overview of File Systems in Linux

Location	Change
<i>Section 1.4, “Large File Support in Linux”</i>	This section has been updated to be consistent with 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/) ↗ .
<i>Section 1.5, “Linux Kernel Storage Limitations”</i>	This section is new.

B.10 October 29, 2012

Corrections for front matter and typographical errata.

B.11 October 19, 2012

Updates were made to the following sections. The changes are explained below.

- *Section B.11.1, “Managing Multipath I/O for Devices”*
- *Section B.11.2, “Overview of File Systems in Linux”*

B.11.1 Managing Multipath I/O for Devices

Location	Change
<i>Section 7.6, “Creating or Modifying the <code>/etc/multipath.conf</code> File”</i>	Specific examples for configuring devices were moved to a higher organization level.

Location	Change
<i>Section 7.6.3, "Verifying the Multipath Setup in the /etc/multipath.conf File"</i>	It is assumed that the <code>multipathd</code> daemon is already running with the old (or default) multipath settings when you modify the <code>/etc/multipath.conf</code> file and perform the dry run.

B.11.2 Overview of File Systems in Linux

Location	Change
<i>Section 1.2.3.4, "Ext3 File System Inode Size and Number of Inodes"</i>	To allow space for extended attributes and ACLs for a file on Ext3 file systems, the default inode size for Ext3 was increased from 128 bytes on SLES 10 to 256 bytes on SLES11.

B.12 September 28, 2012

Updates were made to the following sections. The changes are explained below.

- *Section B.12.1, "Managing Multipath I/O for Devices"*
- *Section B.12.2, "Overview of File Systems in Linux"*

B.12.1 Managing Multipath I/O for Devices

Location	Change
<i>Section 7.2.2, "PRIO Settings in multipath-tools-0.4.9"</i>	In SLES 11 SP2, the <code>prio</code> keyword is used to specify the prioritizer, and the <code>prio_args</code> keyword is used to specify its argument if the prioritizer requires an argument.

Location	Change
	Multipath Tools 0.4.9 and later uses the <code>prio</code> setting in the <code>defaults{}</code> or <code>devices{}</code> section of the <code>/etc/multipath.conf</code> file. It silently ignores the keyword <code>prio</code> when it is specified for an individual <code>multipath</code> definition in the <code>multipaths{}</code> section.
<i>Section 7.11.2.1, "Understanding Priority Groups and Attributes"</i>	Added information about using <code>prio</code> and <code>prio_args</code> keywords.
<i>Section 7.11.2.1, "Understanding Priority Groups and Attributes"</i>	In SLES 11 SP2, the <code>rr_min_io</code> multipath attribute is obsoleted and replaced by the <code>rr_min_io_rq</code> attribute.
<i>Section 7.19.1, "PRIO Settings for Individual Devices Fail After Upgrading to Multipath 0.4.9"</i>	This section is new.
<i>Section 7.19.2, "PRIO Settings with Arguments Fail After Upgrading to multipath-tools-0.4.9"</i>	This section is new.
<i>Section 7.19.3, "Technical Information Documents"</i>	Fixed broken links.

B.12.2 Overview of File Systems in Linux

Location	Change
<i>Section 1.2.1, "Btrfs"</i>	The Btrfs tools package is <code>btrfsprogs</code> .

B.13 April 12, 2012

Updates were made to the following sections. The changes are explained below.

- *Section B.13.1, “Managing Multipath I/O for Devices”*
- *Section B.13.2, “Resizing Software RAID Arrays with mdadm”*

B.13.1 Managing Multipath I/O for Devices

This section was updated to use the `/dev/disk/by-id` directory for device paths in all examples.

B.13.2 Resizing Software RAID Arrays with mdadm

Location	Change
<i>Section 11.3, “Decreasing the Size of a Software RAID”</i>	Revised the order of the procedures so that you reduce the size of the RAID before you reduce the individual component partition sizes.

B.14 February 27, 2012 (SLES 11 SP2)

Updates were made to the following sections. The changes are explained below.

- *Section B.14.1, “Fibre Channel Storage over Ethernet Networks: FCoE”*
- *Section B.14.2, “GNU Licenses”*
- *Section B.14.3, “LVM Configuration”*
- *Section B.14.4, “Managing Access Control Lists over NFSv4”*
- *Section B.14.5, “Managing Multipath I/O for Devices”*
- *Section B.14.6, “Managing Software RAIDs 6 and 10 with mdadm”*
- *Section B.14.7, “Mass Storage over IP Networks: iSCSI”*
- *Section B.14.8, “Overview of File Systems on Linux”*

- *Section B.14.9, “Resizing File Systems”*
- *Section B.14.10, “Resizing Software RAID Arrays with mdadm”*

B.14.1 Fibre Channel Storage over Ethernet Networks: FCoE

This section is new. Open Fibre Channel over Ethernet (OpenFCoE) is supported beginning in SLES 11.

B.14.2 GNU Licenses

This section is new.

B.14.3 LVM Configuration

Location	Change
<i>Section 4.7, “Tagging LVM2 Storage Objects”</i>	This section is new.

B.14.4 Managing Access Control Lists over NFSv4

This section is new.

B.14.5 Managing Multipath I/O for Devices

Location	Change
<i>Section 7.7, “Configuring Default Policies for Polling, Queueing, and Fail-back”</i>	The default <code>getuid</code> path for SLES 11 is <code>/lib/udev/scsi_id</code> .

Location	Change
<p><i>Section 7.17, "Managing I/O in Error Situations"</i></p> <p><i>Section 7.18, "Resolving Stalled I/O"</i></p>	In the dmsetup message commands, the 0 value represents the sector and is used when sector information is not needed.
<i>Section 7.11.2.1, "Understanding Priority Groups and Attributes"</i>	Recommendations were added for the no_path_retry and failback settings when multipath I/O is used in a cluster environment.
<i>Section 7.11.2.1, "Understanding Priority Groups and Attributes"</i>	<p>The path-selector option names and settings were corrected:</p> <p>round-robin 0</p> <p>least-pending 0</p> <p>service-time 0</p> <p>queue-length 0</p>


B.14.6 Managing Software RAID6 and 10 with mdadm

Location	Change
<i>Section 10.3.1.6, "Offset Layout"</i>	This section is new.

B.14.7 Mass Storage over IP Networks: iSCSI

Location	Change
<i>Section 14.4, "Using iSCSI Disks when Installing"</i>	This section is new.
<i>Section 14.5.4, "iSCSI Targets Are Mounted When the Configuration File Is Set to Manual"</i>	This section is new.

B.14.8 Overview of File Systems on Linux

Location	Change
<i>Section 1.2.1, "Btrfs"</i>	<p>This section is new.</p> <p>With SUSE Linux Enterprise 11 SP2, the Btrfs file system is supported as root file system, that is, the file system for the operating system, across all architectures of SUSE Linux Enterprise 11 SP2.</p>
<i>Section 1.2.4, "ReiserFS"</i>	<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 new ReiserFS file systems starting with SUSE Linux Enterprise Server 12.</p>
<i>Section 1.2.6, "File System Feature Comparison"</i>	<p>This section is new.</p>
<i>Section 1.4, "Large File Support in Linux"</i>	<p>The values in this section were updated to current standards.</p>
<i>Section 1.6, "Managing Devices with the YaST Partitioner"</i>	<p>This section is new.</p>

B.14.9 Resizing File Systems

Location	Change
<i>Section 5.1, "Guidelines for Resizing"</i> <i>Section 5.4, "Decreasing the Size of an Ext2 or Ext3 File System"</i>	<p>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</p>

Location	Change
	is mounted or unmounted. The Ext2/4 file systems must be unmounted for increasing or decreasing the volume size.

B.14.10 Resizing Software RAID Arrays with mdadm

Location	Change
<i>Section 11.2.2, "Increasing the Size of the RAID Array"</i>	The <code>--assume-clean</code> option is new.

B.15 July 12, 2011

Updates were made to the following section. The changes are explained below.

- *Section B.15.1, "Managing Multipath I/O for Devices"*

B.15.1 Managing Multipath I/O for Devices

Location	Change
<i>Section 7.2.4, "Using LVM2 on Multipath Devices"</i>	Running <code>dracut</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.
<i>Section 7.13, "Configuring Multipath I/O for an Existing Software RAID"</i>	

B.16 June 14, 2011

Updates were made to the following section. The changes are explained below.

- *Section B.16.1, “Managing Multipath I/O for Devices”*
- *Section B.16.2, “What’s New for Storage in SLES 11”*

B.16.1 Managing Multipath I/O for Devices

Location	Change
<i>Section 7.11.2.1, “Understanding Priority Groups and Attributes”</i>	The default setting for path_grouping_policy changed from multibus to failover in SLES 11.

B.16.2 What’s New for Storage in SLES 11

Location	Change
<i>Section 2.4.13, “Change from Multibus to Failover as the Default Setting for the MPIO Path Grouping Policy”</i>	This section is new.

B.17 May 5, 2011

This release fixes broken links and removes obsolete references.

B.18 January 2011

Updates were made to the following section. The changes are explained below.

- *Section B.18.1, “LVM Configuration”*
- *Section B.18.2, “Managing Multipath I/O for Devices”*
- *Section B.18.3, “Resizing File Systems”*

B.18.1 LVM Configuration

Location	Change
<i>Section 4.3, "Creating Volume Groups"</i>	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.18.2 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.18.3 Resizing File Systems

Location	Change
<i>Section 11.3.1, "Decreasing the Size of the File System"</i>	Decreasing the size of the file system is supported when the file system is unmounted.

B.19 September 16, 2010

Updates were made to the following sections. The changes are explained below.

- *Section B.19.1, "LVM Configuration"*

B.19.1 LVM Configuration

Location	Change
<i>Section 4.2, "Creating LVM Partitions"</i>	The discussion and procedure were expanded to explain how to configure a partition that uses the entire disk. The procedure was modified to use the Hard Disk partitioning feature in the YaST Partitioner.
All LVM Management sections	Procedures throughout the chapter were modified to use Volume Management in the YaST Partitioner.
<i>Section 4.8, "Resizing a Volume Group"</i>	This section is new.
<i>Section 4.9, "Resizing a Logical Volume with YaST"</i>	This section is new.
<i>Section 4.11, "Deleting a Volume Group"</i>	This section is new.
<i>Section 4.12, "Deleting an LVM Partition (Physical Volume)"</i>	This section is new.

B.20 June 21, 2010

Updates were made to the following sections. The changes are explained below.

- *Section B.20.1, "LVM Configuration"*
- *Section B.20.2, "Managing Multipath I/O"*
- *Section B.20.3, "Managing Software RAID6 and 10 with mdadm"*
- *Section B.20.4, "Mass Storage on IP NetWork: iSCSI"*

B.20.1 LVM Configuration

Location	Change
<i>Section 4.2, "Creating LVM Partitions"</i>	Details were added to the procedure.

B.20.2 Managing Multipath I/O

Location	Change
<i>Section 7.9, "Configuring User-Friendly Names or Alias Names"</i>	Using user-friendly names for the root device can result in data loss. Added alternatives from TID 7001133: <i>Recommendations for the usage of user_friendly_names in multipath configurations</i> (http://www.suse.com/support/kb/doc.php?id=7001133) ↗.

B.20.3 Managing Software RAID6 and 10 with mdadm

Location	Change
<i>Section 10.3.1.5, "Far Layout"</i>	Errata in the example were corrected.

B.20.4 Mass Storage on IP NetWork: iSCSI

Location	Change
<i>Section 14.5.1, "Hotplug Doesn't Work for Mounting iSCSI Targets"</i>	This section is new.

B.21 May 2010 (SLES 11 SP1)

Updates were made to the following sections. The changes are explained below.

- *Section B.21.1, “Managing Multipath I/O for Devices”*
- *Section B.21.2, “Mass Storage over IP Networks: iSCSI”*
- *Section B.21.3, “Software RAID Configuration”*
- *Section B.21.4, “What’s New”*

B.21.1 Managing Multipath I/O for Devices


Location	Change
<i>Section 7.2.4, “Using LVM2 on Multipath Devices”</i>	The example in <i>Step 3</i> was corrected.
<i>Section 7.2.8, “SAN Timeout Settings When the Root Device Is Multipathed”</i>	This section is new.
<i>Section 7.3.2, “Multipath I/O Management Tools”</i>	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 Specifications Package Descriptions Web page</i> (http://www.suse.com/products/server/technical-information/), find your architecture and select <i>Packages Sorted by Name</i> , then search on “multipath-tools” to find the package list for that architecture.
<i>Section 7.4.1, “Preparing SAN Devices for Multipathing”</i>	If the SAN device will be used as the root device on the server, modify the timeout settings for the device as described in <i>Section 7.2.8, “SAN Timeout Settings When the Root Device Is Multipathed”</i> .
<i>Section 7.6.3, “Verifying the Multipath Setup in the /etc/multipath.conf File”</i>	Added example output for -v3 verbosity.
<i>Section 7.12.1.1, “Enabling Multipath I/O at Install Time on an Active/Active Multipath Storage LUN”</i>	This section is new.

Location	Change
<i>Section 7.12.1.2, “Enabling Multipath I/O at Install Time on an Active/Passive Multipath Storage LUN”</i>	This section is new.

B.21.2 Mass Storage over IP Networks: iSCSI

Location	Change
<i>Step 7.g in Section 14.2.2, “Creating iSCSI Targets with YaST”</i>	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 to provide this information to consumers of the resources.
<i>Section 14.5, “Troubleshooting iSCSI”</i>	This section is new.

B.21.3 Software RAID Configuration

Location	Change
<i>Section 8.4, “For More Information”</i>	The Software RAID HOW-TO has been deprecated. Use the <i>Linux RAID wiki</i> (https://raid.wiki.kernel.org/index.php/Linux_Raid)  instead.

B.21.4 What’s New

Location	Change
<i>Section 2.3, “What’s New in SLES 11 SP1”</i>	This section is new.

Location	Change

B.22 February 23, 2010

Updates were made to the following sections. The changes are explained below.

- *Section B.22.1, "Configuring Software RAID for the Root Partition"*
- *Section B.22.2, "Managing Multipath I/O"*

B.22.1 Configuring Software RAID for the Root Partition

Location	Change
<i>Section 9.1, "Prerequisites for Using a Software RAID1 Device for the Root Partition"</i>	Corrected an error in the RAID 0 definition..

B.22.2 Managing Multipath I/O

Location	Change
<i>Section 7.14, "Scanning for New Devices without Rebooting"</i>	Added information about using the <code>rescan-sc-si-bus.sh</code> script to scan for devices without rebooting.
<i>Section 7.15, "Scanning for New Partitioned Devices without Rebooting"</i>	Added information about using the <code>rescan-sc-si-bus.sh</code> script to scan for devices without rebooting.

B.23 December 1, 2009

Updates were made to the following sections. The changes are explained below.

- *Section B.23.1, "Managing Multipath I/O for Devices"*
- *Section B.23.2, "Resizing File Systems"*
- *Section B.23.3, "What's New"*

B.23.1 Managing Multipath I/O for Devices

Location	Change
<i>Section 7.2.4, "Using LVM2 on Multipath Devices"</i> <i>Section 7.13, "Configuring Multipath I/O for an Existing Software RAID"</i>	The -f mpath option changed to -f multipath: mkinitrd -f multipath
prio_callout in <i>Section 7.11.2.1, "Understanding Priority Groups and Attributes"</i>	Multipath prio_callout programs are located in shared libraries in <u>/lib/libmultipath/lib*</u> . By using shared libraries, the callout programs are loaded into memory on daemon startup. <div></div>

B.23.2 Resizing File Systems

Location	Change
<i>Section 5.1.1, "File Systems that Support Resizing"</i>	The resize2fs utility supports online or offline resizing for the ext3 file system. <div></div>

Location	Change

B.23.3 What's New

Location	Change
<i>Section 2.4.11, "Location Change for Multipath Tool Callouts"</i>	This section is new.
<i>Section 2.4.12, "Change from mpath to multipath for the mkinitrd -f Option"</i>	This section is new.

B.24 October 20, 2009

Updates were made to the following sections. The changes are explained below.

- *Section B.24.1, "LVM Configuration"*
- *Section B.24.2, "Managing Multipath I/O for Devices"*
- *Section B.24.3, "What's New"*

B.24.1 LVM Configuration

Location	Change
<i>Section 4.1, "Understanding the Logical Volume Manager"</i>	In the YaST Control Center, select <i>System > Partitioner</i> .

Location	Change

B.24.2 Managing Multipath I/O for Devices

Location	Change
<i>Section 7.8, "Blacklisting Non-Multipath Devices"</i>	The keyword <code>devnode_blacklist</code> has been deprecated and replaced with the keyword <code>blacklist</code> .
<i>Section 7.7, "Configuring Default Policies for Polling, Queueing, and Fail-back"</i>	Changed <code>getuid_callout</code> to <code>getuid</code> . <div></div>
<i>Section 7.11.2.1, "Understanding Priority Groups and Attributes"</i>	Changed <code>getuid_callout</code> to <code>getuid</code> . <div></div>
<i>Section 7.11.2.1, "Understanding Priority Groups and Attributes"</i>	Added descriptions for <code>path_selector</code> of least-pending, length-load-balancing, and service-time options. <div></div>

B.24.3 What's New

Location	Change
<i>Section 2.4.10, "Advanced I/O Load-Balancing Options for Multipath"</i>	This section is new. <div></div>

B.25 August 3, 2009

Updates were made to the following section. The change is explained below.

- *Section B.25.1, "Managing Multipath I/O"*

B.25.1 Managing Multipath I/O

Location	Change
<i>Section 7.2.7, "Using --noflush with Multipath Devices"</i>	This section is new. <div></div>

B.26 June 22, 2009

Updates were made to the following sections. The changes are explained below.

- *Section B.26.1, "Managing Multipath I/O"*
- *Section B.26.2, "Managing Software RAIDs 6 and 10 with mdadm"*
- *Section B.26.3, "Mass Storage over IP Networks: iSCSI"*

B.26.1 Managing Multipath I/O

Location	Change
<i>Section 7.12, "Configuring Multipath I/O for the Root Device"</i>	Added <i>Step 4</i> and <i>Step 6</i> for System Z. <div></div>
<i>Section 7.15, "Scanning for New Partitioned Devices without Rebooting"</i>	Corrected the syntax for the command lines in <i>Step 2</i> .

Location	Change
<i>Section 7.15, "Scanning for New Partitioned Devices without Rebooting"</i>	<i>Step 7</i> replaces old Step 7 and Step 8.

B.26.2 Managing Software RAID6 and 10 with mdadm

Location	Change
<i>Section 10.4, "Creating a Degraded RAID Array"</i>	<p>To see the rebuild progress while being refreshed every second, enter</p> <pre>watch -n 1 cat /proc/mdstat</pre>

B.26.3 Mass Storage over IP Networks: iSCSI

Location	Change
<i>Section 14.3.1, "Using YaST for the iSCSI Initiator Configuration"</i>	<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 target device will be evaluated from the initrd on server boots.

B.27 May 21, 2009

Updates were made to the following section. The changes are explained below.

- *Section B.27.1, "Managing Multipath I/O"*

B.27.1 Managing Multipath I/O

Location	Change
<i>Section 7.2.11.1, "Storage Arrays That Are Automatically Detected for Multipathing"</i>	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 <i>Section 7.10, "Configuring Default Settings for zSeries Devices"</i> .
<i>Section 7.3.1, "Device Mapper Multipath Module"</i>	Multipathing is supported for the <code>/boot</code> device in SUSE Linux Enterprise Server 11 and later.
<i>Section 7.10, "Configuring Default Settings for zSeries Devices"</i>	This section is new.
<i>Section 7.12, "Configuring Multipath I/O for the Root Device"</i>	DM-MP is now available and supported for <code>/boot</code> and <code>/root</code> in SUSE Linux Enterprise Server 11.