

# Veritas Storage Foundation<sup>™</sup> Migration Guide

AIX

5.0

# Veritas Storage Foundation Migration Guide

Copyright © 2006 Symantec Corporation. All rights reserved.

Veritas Storage Foundation 5.0

Symantec, the Symantec Logo, Veritas and Veritas Storage Foundation are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Symantec Corporation and its licensors, if any.

THIS DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID, SYMANTEC CORPORATION SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be “commercial computer software” and “commercial computer software documentation” as defined in FAR Sections 12.212 and DFARS Section 227.7202.

Symantec Corporation  
20330 Stevens Creek Blvd.  
Cupertino, CA 95014  
[www.symantec.com](http://www.symantec.com)

## Third-party legal notices

Third-party software may be recommended, distributed, embedded, or bundled with this Symantec product. Such third-party software is licensed separately by its copyright holder. All third-party copyrights associated with this product are listed in the accompanying release notes.

AIX is a registered trademark of IBM Corporation.

## Licensing and registration

Veritas Storage Foundation is a licensed product. See the *Veritas Storage Foundation Installation Guide* for license installation instructions.

## Technical support

For technical assistance, visit <http://support.veritas.com> and select phone or email support. Use the Knowledge Base search feature to access resources such as TechNotes, product alerts, software downloads, hardware compatibility lists, and our customer email notification service.



# Contents

Chapter 1	VxVM and LVM	
	Introducing Veritas Volume Manager .....	7
	Notable features of VxVM .....	8
	VxVM and LVM—conceptual comparison .....	11
	Coexistence of VxVM and LVM disks .....	14
Chapter 2	Converting LVM, JFS and JFS2 to VxVM and VxFS	
	Initializing unused LVM physical volumes to VxVM disks .....	16
	Removing LVM disk information .....	16
	Initializing disks for VxVM use .....	16
	Converting LVM volume groups to VxVM disk groups .....	17
	Volume group conversion limitations .....	18
	Conversion process summary .....	19
	Conversion of JFS and JFS2 file systems to VxFS .....	20
	Conversion steps explained .....	21
	Restoring the LVM volume group configuration .....	28
	Examples of using vxconvert .....	29
	Displaying the vxconvert main menu .....	29
	Listing disk information .....	29
	Listing LVM volume group information .....	30
	Analyzing LVM volume groups, JFS and JFS2 for conversion .....	31
	Converting LVM volume groups, and JFS or JFS2 file systems .....	33
	Sample output before and after conversion .....	35
	General information regarding conversion speed .....	37
Chapter 3	Command differences	
	LVM and VxVM command equivalents .....	40
	Comparison of LVM and VxVM tasks .....	43
	Tasks with no direct LVM equivalents .....	51
	LVM features not supported in VxVM .....	54
Chapter 4	System Management Interface Tool (SMIT)	
	Launching SMIT .....	56
	Administering disk groups in SMIT .....	57

Administering disk devices in SMIT .....	58
Administering volumes in SMIT .....	59
Administering VxVM tunables in SMIT .....	60

Appendix A Conversion error messages

# VxVM and LVM

This chapter provides an overview of Veritas Volume Manager by Symantec (also referred to as VxVM) and its features. A brief description of the benefits of migrating from the AIX Logical Volume Manager (LVM) to VxVM (including migration of JFS or JFS2 file systems to the Veritas File System, VxFS), and the coexistence of VxVM disks with LVM disks is also given.

The following topics are discussed in this chapter:

- [Introducing Veritas Volume Manager](#)
- [VxVM and LVM—conceptual comparison](#)
- [Coexistence of VxVM and LVM disks](#)

## Introducing Veritas Volume Manager

Veritas Volume Manager is an alternative Volume Management product for AIX that includes mirroring features. It offers many capabilities that are not available with the AIX LVM products today.

- Veritas Volume Manager can coexist with LVM. Users can decide which volumes they want managed by each volume manager. For users who want to migrate LVM volume groups to VxVM disk groups, a conversion utility is included. This utility, `vxconvert`, is described in detail in “[Converting LVM, JFS and JFS2 to VxVM and VxFS.](#)”
- Users may choose to use Veritas Volume Manager for their non-root disks.

## Notable features of VxVM

Veritas Volume Manager provides many features, some of which are not available with LVM. Notable VxVM features are described in the list below. See the other Veritas Volume Manager documents (listed in the Preface) for more details about using these features. The section, “[Tasks with no direct LVM equivalents](#)”, lists those features in VxVM that are not available under LVM.

Veritas Volume Manager includes the following features:

- Concatenation, the combining of discontinuous disk regions into virtual devices.
- Spanning, concatenation across different physical media.
- Striping, distribution of storage mappings for a virtual device so that multi-threaded accesses tend to cause even use of all physical media.
- The VERITAS Enterprise Administrator GUI.
- Dynamic Multipathing (DMP) for Active/Active and Active/Passive devices. DMP provides higher availability to data on disks with multiple host-to-device pathways by providing a disk/device path failover mechanism. In the event of a loss of one connection to a disk, the system continues to access the data over the other available connections to the disk. DMP also provides in some cases, improved I/O performance from disks with multiple concurrently available pathways by balancing the I/O load uniformly across multiple I/O paths to the disk device. LVM supports path failover but does not support I/O balancing. DMP support may be used with devices that show improved performance when I/O is balanced across the multiple paths such as IBM SHARK, EMC Symmetrix disk array, and other OEM array devices.
- Free Space Management, providing simple goal-based allocation of storage.
- Task Monitor, which tracks the progress of system recovery by monitoring task creation, maintenance, and completion. The Task Monitor allows you to pause, resume, and stop as desired to adjust the impact on system performance.
- Support for VxFS (not available with LVM).

The following Veritas Volume Manager features may require an additional license:

- Multiple mirroring with up to 32 mirror copies of a volume’s address space.
- Mirrored stripes, enabling mirroring of individual data stripes to spread data across multiple disks while providing data redundancy. This layout dramatically increases the ability to handle multiple disk failures and reduces the amount of resynchronization needed. If a disk

fails, the data on the surviving disks can be used to reconstruct and recover the lost data.

- Hot-relocation, which allows a system to react automatically to I/O failures on redundant (mirrored or RAID-5) VxVM objects, restoring redundancy and access to those objects without administrative intervention. VxVM detects I/O failures on objects and relocates the affected subdisks. The `vxunreloc` utility can be used to restore the system to the same configuration that existed before the disk failure.
- RAID-5, which provides data redundancy by using parity, at a lower storage cost than mirroring. RAID-5 provides data redundancy by using parity. Parity is a calculated value used to reconstruct data after a failure. While data is being written to a RAID-5 volume, parity is calculated by doing an exclusive OR (XOR) procedure on the data. The resulting parity is then written in an interleaved fashion to the RAID-5 array established by the volume. If a portion of a RAID-5 volume fails, the data that was on that portion of the failed volume can be recreated from the remaining data and parity information.
- Online Data Migration, which allows for regions of storage on physical media to be dynamically moved to other physical devices.
- Online Relayout or Dynamic Restriping, the ability to change logical data configuration while online, for example, to change RAID-5 to a mirrored layout or to change a stripe unit size. The volume data remains available during the relayout.
- Striped mirrors (RAID-0 + RAID-1), which provide a layered volume structure that tolerates failure better and which has greater redundancy than a mirrored stripe (RAID-1 + RAID-0) structure. Each subdisk is mirrored so that recovery is quicker (only the subdisk is recovered instead of a full mirror). Refer to the *Veritas Volume Manager Administrator's Guide* for more detailed information on these layouts.
- Improved RAID-5 subdisk, using layered volume technology where the RAID-5 subdisk move operation leaves the old subdisk in place while the new one is being synchronized, thus maintaining redundancy and resiliency to failures during the move.
- Veritas FlashSnap™ which includes the disk group split and join, and Persistent FastResync features of Veritas Volume Manager as well as the Storage Checkpoint features of Veritas File System. Refer to the *Veritas FlashSnap Point-In-Time Copy Solutions Administrator's Guide* for more information about these features.

For more information on LVM, refer to *AIX logical Volume Manager, from A to Z: Introduction and Concepts*. For information on VxVM commands, refer to the

Veritas Volume Manager documentation. A full list of the many new features that are supported in VxVM 5.0 is given in the *Release Notes*.

## VxVM and LVM—conceptual comparison

The following section compares the terminology used in LVM and VxVM at a conceptual level. For more information, refer to the glossary of this Guide for precise and detailed definitions of these terms.

**Table 1-1** A conceptual comparison of LVM and VxVM

LVM term	VxVM term
LVM	VxVM
<p>Both LVM and VxVM enable online disk storage management. They both build virtual devices, called volumes, on physical disks. Volumes are not limited by the underlying physical disks, and can include other virtual objects such as mirrors. Volumes are accessed through the AIX file system, a database, or other applications in the same manner as physical disks would be accessed.</p>	
Physical volume	VxVM disk
<p>An LVM physical volume and a VxVM disk are conceptually the same. A physical disk is the basic storage device (media) where the data is ultimately stored. You can access the data on a physical disk by using a device name (<code>devname</code>) to locate the disk.</p> <p>In LVM, a disk that has been initialized by LVM becomes known as a physical volume. A VxVM disk is one that is placed under the Volume Manager control and is added to a disk group.</p> <p>VxVM can place a disk under its control without adding it to a disk group. The VxVM Storage Administrator shows these disks as “free space pool”.</p>	
Logical volume	Volume
<p>An LVM logical volume and a VxVM volume are conceptually the same. Both are virtual disk devices that appear to applications, databases, and file systems like physical disk devices, but do not have the physical limitations of physical disk devices. Due to its virtual nature, a volume (LVM or VxVM) is not restricted to a particular disk or a specific area of a disk.</p> <p>An LVM volume is composed of fixed length extents. LVM volumes can be mirrored or striped.</p> <p>VxVM volumes consist of one or more plexes/mirrors holding a copy of the data in the volume which in turn are made up of subdisks with arbitrary length. The configuration of a volume can be changed by using the VxVM user interfaces. See the <i>Veritas Volume Manager Administrator's Guide</i> for more information. VxVM volumes can be one of four types: mirrored, RAID-5, striped, or concatenated.</p>	

**Table 1-1** A conceptual comparison of LVM and VxVM

LVM term	VxVM term
Volume group	Disk group

LVM volume groups are conceptually similar to VxVM disk groups.

An LVM volume group is the collective identity of a set of physical volumes, which provide disk storage for the logical volumes.

A VxVM disk group is a collection of VxVM disks that share a common configuration. A configuration is a set of records with detailed information about related VxVM objects, their attributes, and their associations.

In addition, both LVM and VxVM have the following characteristics:

Volumes can be mapped to multiple VxVM disks or LVM physical volumes.

VxVM disks must reside in only one disk group, and LVM physical volumes must reside in one volume group.

Physical extent	Subdisk
-----------------	---------

User data is contained in physical extents in LVM and subdisks in VxVM.

The LVM physical extents are of a fixed length. LVM allocates space in terms of physical extents which is a set of physical disk blocks on a physical volume. The extent size for all physical volumes within a volume group must be the same.

VxVM allocates disk space in term of subdisks which is a set of physical disk blocks representing a specific portion of a VxVM disk and is of arbitrary size.

VGRA	Private region
------	----------------

VGRA and the Private Region are similar conceptually.

In LVM, the VGRA is the region in the disk which stores metadata.

In VxVM, the private region of a disk contains various on-disk structures that are used by the Volume Manager for various internal purposes. Private regions can also contain copies of a disk group's configuration, and copies of the disk group's kernel log.

**Table 1-1** A conceptual comparison of LVM and VxVM

LVM term	VxVM term
Unused physical extent	Free space
<p>LVM contains unused physical extents that are not part of a logical volume, but are part of the volume group.</p>	
<p>Similarly, free space is an area of a disk under VxVM that is not allocated to any subdisk or reserved for use by any other Volume Manager object.</p>	
Mirrors	Mirrors (plexes)
<p>Both LVM and VxVM support mirrors. Mirrors can be used to store multiple copies of a volume's data on separate disks.</p>	
<p>Mirrors allow duplicate copies of the extents to be kept on separate physical volumes. AIX LVM supports up to 3 mirrors.</p>	
<p>A VxVM mirror consists of plexes. Each plex is a copy of the volume. A plex consists of one or more subdisks located on one or more disks. VxVM volumes can have up to 32 mirrors (where each plex is a copy of data). Mirroring features are available with an additional license.</p>	
Export	Deport
<p>In LVM, exporting removes volume group information. The volume group must have already been deactivated.</p>	
<p>Similarly in VxVM, deport makes a disk group inaccessible by the system.</p>	
Import	Import
<p>In LVM, import adds a volume group to the system and the volume group information to /etc/filesystems but does not make the volumes accessible. The volume group must be activated by the <code>varyonvg</code> command in order to make volumes accessible.</p>	
<p>In VxVM, <code>vxdg import</code> imports a disk group and makes the diskgroup accessible by the system.</p>	
Bad block pool	No similar term
<p>In LVM, the bad block pool provides for the transparent detection of bad disk sectors, and the relocation of data from bad to good disk sectors. The bad block reallocation feature does not exist in VxVM because the vectoring of bad blocks is now done by most hardware.</p>	

## Coexistence of VxVM and LVM disks

Both LVM disks and VxVM disks can exist together on a system. The LVM disks are detected and displayed as such by VxVM. LVM disks are not selected by VxVM for initialization, addition, or replacement.

The `vxconvert` command is provided to enable LVM disks to be converted to a VxVM disk format without losing any data. It also includes support for the migration of JFS or JFS2 to VxFS. For more information, see “[Converting LVM, JFS and JFS2 to VxVM and VxFS](#)” on page 15.

# Converting LVM, JFS and JFS2 to VxVM and VxFS

This chapter explains how to convert your LVM, JFS and JFS2 configuration to a VxVM and VxFS configuration and presents the following main topics:

- [Initializing unused LVM physical volumes to VxVM disks](#)
- [Converting LVM volume groups to VxVM disk groups](#) (the conversion process also includes the conversion of JFS and JFS2 file systems stored in LVM volume groups to VxFS)
- [Restoring the LVM volume group configuration](#)
- [Examples of using vxconvert](#)

The basic tools for conversion are the VxVM commands, `vxconvert` and `vxdiskadm`. The discussion here details how to use these tools and gives some insights into how these tools work.

The disks on your system managed by LVM can be of two types: LVM disks in volume groups, and unused disks.

The former are disks that contain logical volumes and volume groups. Unused disks contain no user data, and are not used by any volume group, but have LVM disk headers written by `cfgmgr`. Conversion is done differently for these two types of disks.

For unused LVM disks you can use `vxdiskadm`. For LVM disks in volume groups, the primary tool for conversion is the `vxconvert` command. For information on `vxdiskadm`, see the man page `vxdiskadm(1M)` or the *Veritas Volume Manager Administrator's Guide*.

## Initializing unused LVM physical volumes to VxVM disks

LVM disks that are not part of any volume group and contain no user data are cleaned up, so that there are no LVM disk headers. Then the disks are put under VxVM control through the normal means of initializing disks.

---

**Caution:** You must be absolutely certain that the disks are not in use in any LVM configuration. If there is any user data on these disks, it will be lost during initialization.

---

### Removing LVM disk information

To remove LVM disk header information from the disks, use the following command:

```
# chpv -c diskname
```

where *diskname* is any physical disk, such as `hdisk4`.

### Initializing disks for VxVM use

To initialize the disk for VxVM use, use the `vxdiskadm` command, selecting the option:

```
1) Add or initialize one or more disks
```

Or use the command:

```
# vxdisk init disk_name
```

VxVM utilities will not tamper with disks that are recognized as owned by LVM (by virtue of the LVM VGRA disk headers). The `vxdisk init` or `vxdiskadm` commands fail if you attempt to use them on an LVM disk without first using the `chpv` command.

# Converting LVM volume groups to VxVM disk groups

---

**Note:** It is recommended that you read through this section carefully before beginning any volume group conversion.

A mounted JFS or JFS2 file system cannot be converted. Unmount such file systems before proceeding with the conversion.

By default `vxconvert` converts LVM volume groups to VxVM disk groups, which are not compatible with the Cross-platform Data Sharing (CDS) feature. You can use the `vxcdsconvert` command to convert such a disk group to a CDS-compatible disk group. For more information, see the `vxcdsconvert(1M)` manual page and the *Veritas Storage Foundation Cross-Platform Data Sharing Administrator's Guide*.

---

This section outlines the process for converting LVM volume groups to VxVM disk groups. During the conversion process, all JFS or JFS2 file systems in a specified LVM volume group are converted to VxFS.

The conversion process involves many steps. Though there are tools to help you with the conversion, some of these steps cannot be automated. You should be sure to understand how the whole conversion process works, and what you will need to do in the process before beginning a volume group conversion.

The tool used for conversion is `vxconvert`. This interactive, menu-driven program walks you through many of the steps of the process of converting volume groups for use by VxVM. Using `vxconvert` can reduce the downtime associated with converting from LVM to VxVM. Without the `vxconvert` tool, the only possible method of conversion would be to take full backups of user data, destroy the existing LVM configuration leaving only raw disks, recreate the configuration in VxVM, and then reload the user data.

The `vxconvert` process converts LVM volume groups to VxVM disk groups *in place*. This means, that the utility changes disks within LVM volume groups to VxVM disks by taking over the areas of the disks used for LVM configuration information, and creating the equivalent VxVM volume configuration information. User data, the portions of the disks used for databases, and so on, are not affected by the conversion. Both JFS and JFS2 data is converted during the conversion.

The act of conversion changes the names by which your system refers to the logical storage. Therefore, the conversion process is necessarily performed off-line. There can be no application access to user data in the volume groups undergoing conversion. Access to the LVM configuration itself (the metadata of LVM) must also be limited to the conversion process.

## Volume group conversion limitations

There are certain LVM volume configurations that cannot be converted to VxVM. Some of the reasons a conversion could fail are:

- A volume group with insufficient space for metadata.  
In the conversion of LVM to VxVM, the areas of the disks used to store LVM metadata are overwritten with VxVM metadata. If the VxVM metadata that needs to be written will not fit the space occupied by the LVM metadata, the group containing the disk cannot be converted. If you have just enough space for the conversion, you probably would want to have more space for future configuration changes.
- A volume group containing the root volume.  
The current release of VxVM on AIX does not support VxVM root volumes. Because of this, `vxconvert` does not convert any volume group that contains a *rootable* volume. Not only is the current root volume off limits, but any volume that might be used as an alternate root volume is rejected as well.
- A volume group containing mirrors using the Mirror Write Cache feature for volume consistency recovery.  
You should be aware that when converting mirrored LVM volumes to VxVM, some of these volumes will likely have the Mirror Write Cache consistency recovery method in force on the volume. The `vxconvert` utility can convert these volumes, but in some cases, it might not be able to create an equivalent level of consistency. Therefore, `vxconvert` will detect this case and warn the user that converting this volume group would lose this MWC functionality and leave the resultant VxVM mirrored disk group operating in a comparatively degraded state.
- Volume groups with any `dump` or primary `swap` volumes.  
Because VxVM does not support rootability, `vxconvert` will not convert swap or paging space on any type of volume to VxVM.
- A volume group containing the `/usr` file system.  
For this release, a volume group containing the `/usr` file system cannot be converted because `vxconvert` needs access to files in `/usr`.
- Volume groups with any disks that have bad blocks in the bad block directory.  
Unlike LVM, VxVM does not support bad block revectoring at the physical volume level. If there appear to be any valid bad blocks in the bad block directory of any disk used in an LVM volume group, the group cannot be converted. See Appendix A, Conversion Error Messages, for actions to take in this situation.

- Not enough disk space on the root LVM volume group to save a copy of each physical disks VGRA area.  
 For large LVM volume groups, especially those with large VGDA sizes, the required space could be greater than 64MB per physical volume. So, for a Volume Group with 128 disks, the required storage space could be greater than 8 GB.  
 The default save area is `/etc/vx/reconfig.d`.
- Volume groups with mirrored volumes.  
 A conversion fails if the LVM volume group being converted has mirrored volumes, but the system does not have a valid license installed that enables mirroring for VxVM.

## Conversion process summary

Several steps are used to convert LVM volume groups to VxVM disk groups. Most of these steps can be done with the `vxconvert` utility. All the steps are not compulsory, and some may have to be followed only if there are problems during conversion. Some of them (e.g. backing up user data) are left to you to accomplish through your regular administrative processes.

The order of the steps in the conversion process is:

- Identify LVM volume groups for conversion.
- Analyze an LVM volume group, and then analyzing JFS or JFS2 file systems, if any, on the volume group to see if conversion is possible.
- Take action to make conversion possible if analysis fails.
- Back up your LVM configuration and user data.
- Plan for new VxVM logical volume names.
- Stop application access to volumes in the volume group to be converted.
- Convert the JFS or JFS2 file systems, if any, on a specified volume group, and then converting the volume group.
- Take action if conversion fails.
- Implement changes for new VxVM logical volume names.
- Restart applications on the new VxVM volumes.
- Tailor your VxVM configuration.

These steps are described in detail in later sections of this chapter. Examples of how to use `vxconvert` are shown in [“Examples of using vxconvert”](#) on page 29. For information on restoring back to your original LVM configuration refer to [“Restoring the LVM volume group configuration”](#) on page 28.

## Conversion of JFS and JFS2 file systems to VxFS

The `vxconvert` utility converts JFS and JFS2 file systems to VxFS file systems with a Version 7 disk layout.

Conversion from a JFS or JFS2 file system to a VxFS file system requires that there is sufficient free space within the file system or immediately after the end of the file system to convert the existing metadata. The space must be available on the same device or volume on which the file system resides. The amount of free space that is required is approximately 12-15% of the total size of the file system size, but the exact amount depends on the number and sizes of files and directories, and on the number of allocated inodes. The conversion process takes up to 3 times longer than running a file system check (`fsck`) on the file system.

After conversion, you can use utilities such as `fsadm` and `vxresize` to reorganize the file system.

The ability to shrink a file system that has been converted to VxFS depends on the amount and location of the remaining free space in the file system. If an attempt to shrink a converted file system fails, specify a smaller shrink size.

JFS or JFS2 log devices and JFS2 snapshot devices are not touched by the conversion process. After the file systems are converted, you can recover the space used by these devices for other purposes.

### Conversion limitations

- You cannot use the `vxconvert` utility to reverse the conversion of a JFS or JFS2 file system to VxFS. Instead, you must recreate the original file system and restore the data from a backup.
- Compressed JFS file systems cannot be converted. You must first decompress a compressed JFS file system before starting the conversion process.
- A JFS file system with a fragment size of 512 bytes cannot be converted.
- The quota files in JFS or JFS2 file systems are not converted to the VxFS quota file format.
- The extended attributes of a JFS file system are not converted to VxFS extended attributes.
- A JFS2 file-system with a block size of 512 bytes cannot be converted.
- A JFS2 file system with inode numbers larger than  $2^{32}$  cannot be converted.
- JFS2 v1 file systems with extended attributes can be converted, but these attributes are not preserved.
- JFS2 v2 file systems with named attributes can be converted, but ACLs and DMAPi attributes are not preserved.

- JFS2 file systems with snapshots may be converted, but the snapshots are not preserved.

## Conversion steps explained

### Identify LVM disks and volume groups for conversion

The obvious first step in the conversion process is to identify what you want to convert. The native LVM administrative utilities like `lsvg` and `SMIT` can help you identify candidate LVM volume groups as well as the disks that comprise them.

You can also use the `vxconvert` and `vxdisk` commands to examine groups and their member disks. The information presented through the `vxconvert` and `vxdisk` utilities and their interpretation is shown in “[Examples of using vxconvert](#)” on page 29.

You can also list the LVM disks with the following VxVM command:

```
# vxdisk list
```

### Analyze an LVM volume group to see if conversion is possible

After you have selected a volume group for conversion, you need to analyze it to determine if conversion for VxVM use is possible.

Use the `analyze` option of `vxconvert` to check for problems that would prevent the conversion from completing successfully. This option checks for all the conditions listed in “[Volume group conversion limitations](#)” on page 18.

The analysis calculates the space required to add the volume group disks to a VxVM disk group, and to replace any existing disks and volumes with VxVM volumes, plexes, and subdisks. If you do not have the required space to convert the disks, the conversion fails. The analysis also calculates the space required to convert volumes containing JFS or JFS2 file systems to VxFS. If there is insufficient space in any of these volumes, the conversion is aborted.

Analysis can be run on a live system while users are accessing their data. To analyze LVM volume groups, choose option 1 of the `vxconvert` utility.

---

**Note:** The analysis option is presented as a separate menu item in `vxconvert`, but there is an implicit analysis with any conversion. If you simply select the “Convert LVM and JFS to VxVM and VxFS” menu option, `vxconvert` will go through analysis on any group you specify. When you are using the convert option directly, you are given a chance to abort the conversion after analysis, and before any changes are committed to disk. For more information, see “[Converting LVM volume groups to VxVM disk groups](#)” on page 17.

The analysis option is useful when you have a large number of groups/disks for conversion and some amount of planning is needed before the actual conversion. Installations with many users or critical applications can use the analyze option on a running system. Then conversion downtime can be better planned and managed. Smaller configurations may be better served by using the convert option directly while in a downtime period.

---

Sample examples of the analyze option are shown in “[Examples of using vxconvert](#)” on page 29.

## Take action to make conversion possible if analysis fails

Analysis may fail for any of the reasons listed in the section “Volume group conversion limitations”.

Messages from `vxconvert` will explain the type of failure and any actions that can be taken before retrying the analysis. Refer to “[Conversion error messages](#)” on page 61 for complete details of specific error messages and actions.

## Back up your LVM configuration and user data

After analysis you know which volume group or groups you want to convert to VxVM disk groups. Up to this point, you have not altered your LVM configuration.

By taking the next step (completing the conversion to VxVM), you are significantly changing access to your storage.

Although the conversion process does not move, or in any other way affect user data, you are strongly encouraged to back up all data on the affected disks.

During a conversion, any spurious reboots, power outages, hardware errors or operating system bugs can have unpredictable and undesirable consequences. You are advised to be on guard against disaster with a set of verified backups.

---

**Note:** The `vxconvert` utility itself also saves a snapshot of the LVM metadata in the process of conversion for each disk. It can only be used via the `vxconvert` program. With certain limitations, you can reinstate the LVM volumes after they have been converted to VxVM using this data. (See “[Displaying the vxconvert main menu](#)” on page 29”.) Even though `vxconvert` provides this level of backup of the LVM configuration, you are advised to back up your data before running `vxconvert`.

---

### Backing up user data

To back up user data, use your regular backup processes.

---

**Caution:** Before you do the backup, you should carefully review “[Implement changes for new VxVM logical volume names](#)” on page 27. Backup processes and systems themselves may have dependencies on the volume names currently in use on your system. The conversion to VxVM changes those names. You are advised to understand the implications name changes have for restoring from the backups you are about to make.

---

### File system back up of user data

You can use the backup utility that you normally use to back up data on your logical volumes. For example, to back up logical volumes that contain file systems, the `backup(1M)` command can be used to back up the data to tape.

For example, to backup the data on `/dev/lvol3` mounted on `/foodir`, use the following command:

```
# backup -0i /foodir -f /dev/rmt#
```

### Non-file system back up

If a logical volume you are converting does not contain a file system, and is being used directly by an application (such as a database application), use the backup facilities provided by the application. If no such facility exists, consider using the `dd` command.

### Plan for new VxVM logical volume names

When you change from LVM volumes to VxVM volumes, the device names by which your system accesses data are changed. LVM creates device nodes for its logical volumes in `/dev` under directories named for the volume group. VxVM creates its device nodes in `/dev/vx/dsk` and `/dev/vx/rdsk`. When conversion is complete, the old LVM device nodes are gone from the system, and the system will access data on the device nodes in `/dev/vx`.

This change in names can present problems. Any application that refers to specific device node names will be at risk when these names change. Similarly, any files that record specific device node names for use by applications can be problematic.

The most obvious area where this problem arises is in the `/etc/filesystems` file. To handle this problem, `vxconvert` rewrites `/etc/filesystems` with the new VxVM names when conversion is complete so that `fsck`, `mount`, and related utilities will behave as they did prior to the conversion.

There are potentially many other applications, though, that may be put at risk by the name changes in conversion. `vxconvert` cannot help with these. The system administrator must examine the mechanisms used in each of the following areas to see if they reference LVM device names:

- Databases run on raw logical devices may record the name of that device node.
- Backup systems may do device level backups based on device node names recorded in private files. Also labelling of the backups may record device names.
- Scripts run by cron(1M).
- Other administrative scripts.

### Workaround

`vxconvert` records a mapping between the names of the LVM device nodes and VxVM device nodes. This data can be used to create symbolic links from the old LVM volume to the new VxVM device names. The mapping is recorded in the file:

```
/etc/vx/reconfig.d/vgrecords/vol_grp_name/vol_grp_name.trans
```

This file provides information on how to proceed further to link the old LVM volume names to the new VxVM device names.

---

**Caution:** This method of resolving the naming problem has risks. The symbolic links can become *stale*. For example, if a database refers to `/dev/vx/rdisk/vol1` through a symbolic link `/dev/rvol1` (“the old LVM name”), and if the underlying VxVM volume configuration is changed in any way, the database could refer to a missing or different volume.

---

---

**Note:** You may want to use this symbolic link approach to ease the transition to VxVM. You can set up the symbolic links after the successful conversion to VxVM. Then, you can do the investigation on a case by case basis for each volume. Once you are satisfied that there are no problems introduced by the name change, the symbolic link to that volume can be removed. You must be careful to maintain a static VxVM volume configuration during this transition period.

---

Over time, the ultimate goal should be that the underlying VxVM naming is used by all applications, and that there are no indirect references to those volumes.

## Stop application access to volumes in the volume group to be converted

No applications can be active on the LVM volume group undergoing conversion. Before attempting to convert any volume group, you must ensure that applications using that group are down. This involves stopping databases, unmounting file systems, and so on.

---

**Note:** You need check and update the `/etc/filesystems` file for valid and supported options for the VxFS file systems before mounting.

---

As described in “[Conversion and reboot](#)” on page 26, `vxconvert` tries to unmount mounted file systems during the conversion. However, `vxconvert` makes no attempt to close down running applications on those file systems, also, it does not attempt to deal with applications (e.g., databases) running on raw LVM volumes.

---

**Note:** It is strongly recommended that you do not rely on `vxconvert`'s mechanisms for unmounting file systems. Conversion will be simpler if you close applications, and unmount file systems before running `vxconvert`.

---

To unmount a file system, use the following command:

```
# umount file_system_device
```

For example:

```
# umount /dev/1v01
```

Having unmounted a file system, use the `fsck` command to check its integrity:

```
# fsck -y file_system_device
```

For example:

```
# fsck -y /dev/1v01
```

## Conversion and reboot

During conversion, after the analysis phase is complete, the disks to be converted are deemed to be *conversion ready*. The `vxconvert` program asks if you are ready to commit to the conversion changes. If you choose to complete the conversion, the system will try to unmount all of the associated mounted file systems, stop and export the volume group, and then install the VxVM configuration.

If `vxconvert` is unable to stop and export volume groups or unmount file systems, the conversion cannot be completed without rebooting the system. You will have the option of aborting the conversion or completing the conversion by rebooting the system. If you choose to reboot, `vxconvert` will trigger the completion of the conversion automatically, during reboot, when it can be guaranteed that no processes have access to the volumes that are being converted.

If you choose to abort rather than reboot to complete the conversion, `vxconvert` will return to the main menu.

---

**Note:** The LVM logical volumes to be converted must all be available to the `vxconvert` process. You should not deactivate the volume group or any logical volumes before running `vxconvert`.

---

## To activate a volume group

If you are not certain if the LVM volumes or the corresponding volume groups are active, you can activate them with the following command:

```
# varyonvg vol_grp_name
```

## Convert a volume group

To do the actual conversion of LVM volume groups to VxVM disk groups, choose option 2 of the `vxconvert` utility.

`vxconvert` will prompt for a name for the VxVM disk group that will be created to replace the LVM volume group you are converting. This is the only object naming that is done through `vxconvert`. For details on modifying VxVM volume names, see “[Tailor your VxVM configuration](#)” on page 27.

As described in “[Analyze an LVM volume group to see if conversion is possible](#)” on page 21, the volume groups selected for conversion are analyzed to ensure that conversion is possible. After a successful analysis phase, `vxconvert` will ask you to commit to the change or abort the conversion. When you select to commit to conversion, the new VxVM metadata is written.

---

**Note:** It is good practice to convert one volume group at a time to avoid errors during conversion.

---

The details of the conversion process are shown in “[Examples of using vxconvert](#)” on page 29.

## Take action if conversion fails

Conversion can fail for any of the reasons detailed in the “Volume group conversion limitations” section. Messages from `vxconvert` will explain the type of failure, and any actions you can take before retrying the conversion.

See “[Conversion error messages](#)” on page 61 for complete details of specific error messages.

## Implement changes for new VxVM logical volume names

You must be sure that all applications and configuration files refer properly to the new VxVM logical volumes. See “[Plan for new VxVM logical volume names](#)” on page 23 for details.

## Restart applications on the new VxVM volumes

Once the conversion to VxVM is complete, file systems can be mounted on the new devices and applications can be restarted.

For the file systems that you unmounted before running `vxconvert`, remount them using the new volume names. `vxconvert` will have updated `/etc/filesystems` with the new names.

After conversion, remove any VxVM log volumes that have been converted from corresponding JFS or JFS2 log volumes.

## Tailor your VxVM configuration

`vxconvert` provides a default name for naming the newly formed VxVM disk group during conversion only as an option. However, you will be given the choice of choosing your own VxVM disk group name. By default, `vxconvert` renames the LVM volume group by replacing the prefix `vg` in the volume group name with the prefix `dg`. For example, `vg08` would become `dg08`. If there is no `vg` in the LVM volume group name, `vxconvert` simply uses the same volume group name for its disk group.

The disks in the new VxVM disk group are given VxVM disk media names (see `vxintro(1M)`) based on this disk group name. If your new VxVM disk group is `dg08`, it will have VxVM disks with names like `dg0801`, `dg0802`, and so on. The

VxVM plexes within the logical volumes will be `dg0801-01`, `dg0801-02`, and so on.

If you do not like the default object names generated by the conversion, use the standard VxVM utilities to rename these objects. See the `rename` option in the `vxedit(1M)` man page for more details on renaming the disk groups.

---

**Note:** You must only rename objects in the VxVM configuration after you are fully satisfied with that configuration. If you have chosen to set up symbolic links to the VxVM volumes (as described in “[Plan for new VxVM logical volume names](#)” on page 23), avoid renaming VxVM objects. These symbolic links are made invalid if the underlying VxVM device node name changes.

---

## Restoring the LVM volume group configuration

If you need to restore the original LVM configuration, you must restore the user data in addition to restoring the old LVM metadata and associated configuration files.

---

**Note:** The snapshot of LVM internal data is kept on the `root` file system.

You must have backed up data located on all the volume groups’ logical volumes before conversion to VxVM.

---

Restoration of LVM volume groups is a two-step process consisting of a restoration of LVM internal data (metadata and configuration files), and restoration of user or application data.

## Examples of using vxconvert

The following sections contain examples of using the `vxconvert` utility.

### Displaying the vxconvert main menu

To display the `vxconvert` menu, use the following command:

```
# vxconvert
...
Press return to continue

VERITAS Storage Foundation Operations
Menu: Volume Manager and File System Conversion

    1      Analyze LVM Volume Groups and JFS/JFS2 File Systems
           for Conversion
    2      Convert LVM and JFS/JFS2 to VxVM and VxFS
    3      Set path for saving VGRA records and JFS/JFS2 meta
           data
list     List disk information
listvg  List LVM Volume Group information

    ?      Display help about menu
    ??     Display help about the menuing system
    q      Exit from menus
```

Select an operation to perform:

### Listing disk information

The `list` option of `vxconvert` displays information about the disks on a system.

```
...
Select an operation to perform: # list
```

```
List disk information
Menu: Volume Manager/LVM_Conversion/ListDisk
```

Use this menu option to display a list of disks. You can also choose to list detailed information about the disk at a specific disk device address.

Enter disk device or "all" [<address>,all,q,?](default: **all**)

DEVICE	DISK	GROUP	STATUS
EXT_DISKS_0	-	-	LVM
EXT_DISKS_1	-	-	online invalid
EXT_DISKS_2	-	-	online invalid
EXT_DISKS_3	-	-	online invalid
EXT_DISKS_4	-	-	online invalid
EXT_DISKS_5	-	-	online invalid

```
EXT_DISKS_6 - - LVM
EXT_DISKS_7 - - online invalid
EXT_DISKS_8 - - LVM
EXT_DISKS_9 disk01 rootdg online invalid
```

Device to list in detail [<address>,none,q,?] (default: none)  
**none**

## Listing LVM volume group information

To list LVM volume group information, use the `listvg` option.

```
...
Select an operation to perform: # listvg

List LVM Volume Group information
Menu: Volume Manager/LVM_Conversion/ListLVMVolumeGroups

Use this menu option to display a list of LVM volume groups. You
can also choose to list detailed information about the LVMVG
at a specific disk device address.

Enter Volume Group (i.e.- vg04) or "all" [<address>,all,q,?]
(default: all) all

LVM VOLUME GROUP INFORMATION
NAME      TYPE      PHYSICAL VOLUME
rootvg    ROOT      hdisk0
vgbig     Non-Root  hdisk14

Volume Group to list in detail
[<address>,none,q,?] (default: none) vgbig

VOLUME GROUP: vgbig VG IDENTIFIER:
                0002d57f00004c00000000e90925243c
VG STATE: active  PP SIZE: 64 megabyte(s)
VG PERMISSION:read/write  TOTAL PPs:135 (8640 megabytes)
MAX LVs:      512      FREE PPs: 82 (5248 megabytes)
LVs:          3        USED PPs: 53 (3392 megabytes)
OPEN LVs:     0        QUORUM: 2
TOTAL PVs:    1        VG DESCRIPTORS: 2
STALE PVs:    0        STALE PPs:0
ACTIVE PVs:   1        AUTO ON: yes
MAX PPs per PV:1016    MAX PVs: 128
LTG size:     256 kilobyte(s)AUTO SYNC:no
HOT SPARE:    no

List another LVM Volume Group? [y,n,q,?] (default: n)
```

---

**Note:** The volume groups you want to convert must not be a root volume group or have bootable volumes in the group.

---

## Analyzing LVM volume groups, JFS and JFS2 for conversion

To analyze one or more LVM volume groups and any JFS or JFS2 file systems, select option 1.

```
...
Select an operation to perform: 1
```

```
Analyze LVM Volume Groups and JFS/JFS2 File Systems
Menu: Volume Manager and File System
Conversion/Analyze_LVM_VG_VXFS_JFSs
```

Use this operation to analyze one or more LVM Volume Groups for possible conversion to VxVM disk groups thus by analyzing the JFS filesystems and LVM volumes present in a Volume Group. This step checks for problems that would prevent the conversion from completing successfully. For example, it calculates the space required to add an LVM Volume Group's disks to a VxVM disk group and to replace any existing LVM partitions and volumes with VxVM Volume Manager volumes, plexes, and sub-disks. Also calculates the space required for JFS/JFS2 filesystem to be converted to VxFS, if any on the specified Volume Group.

For this release, conversion is only allowed for Non-root LVM Volume Groups and unmounted JFS/JFS2 filesystems present on it. Hence, analysis is only allowed on Non-root LVM Volume Groups and an unmounted JFS/JFS2 filesystems.

More than one Volume Group may be entered at the prompt. Here are some LVM Volume Group selection examples:

```
listvg:      list all LVM Volume Groups
list:       list all disk devices
vg_name:    a single LVM Volume Group, named vg_name
```

```
Select Volume Groups to analyze:
[list,listvg,q,?] vgbig
```

```
vgbig
```

```
Analyze this Volume Group for File Systems and Logical Volumes?
[y,n,q,?] (default: y)
```

The following disk has been found in the volume group and will be analyzed for FS and VM conversion.

```
EXT_DISKS_6
```

```
The following are the JFS Filesystems found in the  
Volume Group (vgbig), to be analysed.
```

```
/dev/lv01 /dev/lv02
```

```
NOTE: This operation might take a long time to complete  
depending on the number of JFS file systems and LVM volumes.
```

```
Continue with operation? [y,n,q,?] (default: y)
```

```
Analysis of JFS FileSystems present in the VolumeGroup, vgbig  
is completed successfully.
```

```
To allow analysis, a new VxVM disk group, vgbig, will be  
fabricated and the disk device hdisk6 will be added to the  
disk group with the disk name vgbig01.
```

```
The first stage of the Analysis process completed successfully.
```

```
Second Stage Conversion Analysis of vgbig
```

```
Analysis of vgbig found sufficient Private Space for conversion
```

```
Conversion Analysis of the following devices was successful.
```

```
hdisk6
```

```
Volume Group Analysis Completed
```

## Example of failed analysis for JFS

```
Analysis of (/dev/lv01) from JFS to VXFS has failed due to the  
error:
```

```
vxfs vxfsconvert: /dev/lv01 is mounted
```

```
Please check the error and do accordingly.
```

## Example of failed analysis for LVM volumes

```
Second Stage Conversion Analysis of vgbig
```

```
There is not enough free space on the /etc/vx device  
to complete the conversion of the LVM Volume Group (vgbig).  
You will need to have at least 17530 blocks free.
```

## Converting LVM volume groups, and JFS or JFS2 file systems

To convert LVM volume groups to VxVM disk groups, and JFS or JFS2 file systems to VxFS, select option 2.

...

Select an operation to perform: **2**

Convert one or more LVM Volume Groups and JFS/JFS2 File Systems  
 Menu: Volume Manager and File System Conversion/Convert

Use this operation to convert LVM Volume Groups to VxVM disk groups along with the conversion of JFS/JFS2 File Systems to VxFS File Systems. This adds the disks to a disk group and replaces existing LVM volumes with VxVM volumes and JFS/JFS2 filesystems with VxFS filesystems. This conversion may require a reboot for the changes to take effect. Before reboot make sure to edit the /etc/filesystems and run full VxFS fsck on converted VxFS file systems. Only Non-root LVM Volume Groups can be converted.

Here are some LVM Volume Group selection examples:

```
listvg:      list all LVM Volume Groups
list:        list all disk devices
vg_name:     a single LVM Volume Group, named vg_name
```

Select Volume Groups to convert : [list,listvg,q,?] **listvg**

### LVM VOLUME GROUP INFORMATION

Name	Type	Physical Volumes
jfs2vg	Non-Root	hdisk6
rootvg	ROOT	hdisk0

Select Volume Groups to convert : [list,listvg,q,?] **jfs2vg**

Convert this Volume Group for File Systems and Logical Volumes?  
 [y,n,q,?] (default: y) **y**

Specify a name for the new VxVM disk group (default: jfs2vg)  
**jfs2vg**

VxVM NOTICE V-5-2-2741 Second Stage Conversion Analysis of  
 jfs2vg

VxVM INFO V-5-2-2733 Analysis found sufficient Private Space for  
 conversion

VxVM INFO V-5-2-2228 Conversion Analysis of the following  
 devices was successful.

hdisk0

```
Are you ready to commit to these changes? [y,n,q,?] (default: y)
y
VxVM vxsave_lvmrecs INFO V-5-2-799 Saving LVM configuration
records for Volume Group jfs2vg ...
VxVM vxsave_lvmrecs INFO V-5-2-775 LVM Volume Group jfs2vg
records saved.

Hit RETURN to continue
```

## Example of a failed JFS conversion

```
Found insufficient space to convert the JFS (/dev/lv01) to VxFS.
At least 2972 kilobytes space is required. To allow conversion,
please go back and increase the filesystem size.
```

```
Analysis of (/dev/lv01) from JFS to VXFS has failed due to the
error:
```

```
vxfs vxfsconvert: Total of 2972K bytes required to complete
the conversion
```

```
Please check the error and do accordingly.
```

## Example of a failed LVM conversion

```
Second Stage Conversion Analysis of vgbig
```

```
There is not enough free space on the /etc/vx device
to complete the conversion of the LVM Volume Group (vgbig).
You will need to have at least 17530 blocks free.
```

## Sample output before and after conversion

### Before conversion

```

# vxprint
Disk group: rootdg
TY      NAME      ASSOC      KSTATE     LENGTH     PLOFFS     STATE  TUTILO    PUTILO
dg      rootdg    rootdg     -           -           -          -         -
dm      disk01    EXT_DISKS_9 -           17780936   -          -         -

# vxconvert
...
Select an operation to perform: listvg
...
Enter Volume Group (i.e.- vg04) or "all"
[<address>,all,q,?] (default: all)

LVM VOLUME GROUP INFORMATION
Name          Type          Physical Volumes

rootvg        ROOT          hdisk0
vgbig         Non-Root     hdisk14
...
Select an operation to perform: list
...
Enter disk device or "all" [<address>,all,q,?] (default: all)

DEVICE      DISK          GROUP        STATUS
EXT_DISKS_0 -             -            LVM
EXT_DISKS_1 -             -            online invalid
EXT_DISKS_2 -             -            online invalid
EXT_DISKS_3 -             -            online invalid
EXT_DISKS_4 -             -            online invalid
EXT_DISKS_5 -             -            online invalid
EXT_DISKS_6 -             -            online invalid
EXT_DISKS_7 -             -            online invalid
EXT_DISKS_8 -             -            online invalid
EXT_DISKS_9 disk01        rootdg       online

DEVICE indicates a physical disk.
DISK names a disk if it is under VxVM control.
GROUP shows the disk group name.
STATUS is LVM if the disk is under LVM control. If the status is
online invalid, that means VxVM acknowledges the disk but does
not have it under its control.

```

## After conversion

```
# vxconvert
...
Select an operation to perform: listvg
...
Enter Volume Group (i.e.- vg04) or "all"
[<address>,all,q,?] (default: all)

LVM VOLUME GROUP INFORMATION
Name      Type      Physical Volumes
rootvg    ROOT      hdisk0
...
Select an operation to perform: q
```

```
# vxprint
Disk group: rootdg
```

TY	NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTIL0	PUTIL0
dg	rootdg	rootdg	-	-	-	-	-	-
dm	disk01	EXT_DISKS_9	-	35589724	-	-	-	-

Disk group: dgbig

TY	NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTIL0	PUTIL0
dg	dgbig	dgbig	-	-	-	-	-	dm
dgbig01	EXT_DISKS_6	-	-	17765832	-	-	-	-

v	loglv00	gen	ENABLED	131072	-	ACTIVE	-	-
pl	loglv00-01	loglv00	ENABLED	131072	-	ACTIVE	-	-
sd	dgbig01-01	loglv00-01	ENABLED	131072	0	-	-	-
v	lv01	fsgen	ENABLED	655360	-	ACTIVE	-	-
pl	lv01-01	lv01	ENABLED	655360	-	ACTIVE	-	-
sd	dgbig01-03	lv01-01	ENABLED	655360	0	-	-	-
v	lv02	fsgen	ENABLED	65536	-	ACTIVE	-	-
pl	lv02-01	lv02	ENABLED	65536	-	ACTIVE	-	-
sd	dgbig01-02	lv02-01	ENABLED	65536	0	-	-	-

The disk group dgbig contains the VxVM disk dgbig01 and the volume loglv00. The VxVM disk dgbig01 is associated with disk device EXT\_DISKS\_6 and is 17765832 blocks in length. The volume loglv00 is of type gen, is enabled in the VxVM kernel driver, is of length 131072, and is in the ACTIVE state. This means that the volume is started, and the plex is enabled. Operations to the volume such as recovery and data access will be governed by the usage type gen. The plex loglv00-01 is associated with volume loglv00, and maps the entire address range of the volume. Associated with the plex is one subdisk, dgbig01-01 which maps the plex address range from 0 to the entire length of the plex, that is, 131072 blocks. As implied by the first part of its name, the subdisk dgbig01-01 uses an extent from the VxVM disk dgbig01.

## General information regarding conversion speed

Factors affecting conversion speed include:

- Size of volume groups. The larger the volume groups, the larger the VGRA area on each disk. A copy must be made of the VGRA area of each physical disk. Some areas are greater than 64MB; therefore a 50-disk volume requires 64MB reads and writes (that is, 100 large I/O requests) to complete. Some volume groups have 128 disks.
- Individual size of a logical volume in a volume group, and the complexity of the logical volume layout. For example, for a system with 50 9GB drives, a simple 50GB logical volume can be created using 6 disks. But a 50GB striped logical volume that takes the first 1GB of all 50 disks can also be created. The first and simple logical volume takes less time to convert than the striped volume since only 5 disks need to be checked for metadata. However, for the striped volume, 50 disks need to be checked and 50 VGRAs to be copied. In addition, the complexity of reproducing the VxVM commands to set up the striped volumes requires more VxVM commands to be generated to represent more smaller subdisks representing the same amount of space. Another factor in converting stripes is that stripes create more work for the converter. In some cases, stripes require 1GB volume, although only the metadata is being changed. In other cases, where there are more physical disks in one volume than another, there is more metadata to deal with. The converter has to read every physical extent map to ensure there are no holes in the volume; if holes are found, the converter maps around them.
- Number of volumes. While it takes longer to convert one 64GB volume than one 2GB volume, it also takes longer to convert 64 1GB volumes than one 64GB volume, providing that the volumes are of similar type.
- Mirrored volumes. Mirrored volumes typically do not take more time to convert than simple volumes. Volumes that are mirrored and striped at the same time would take longer.  
 Currently, after conversion, mirrored volumes are not automatically synchronized because a large mirror could take hours to complete.  
 For example, in tests, a 150 GB volume group consisting of 20 simple logical volumes takes approximately 35-40 minutes to convert. In contrast, the same volume group (150GB) consisting of mirrored volumes that need to be synchronized can take 30-40 hours to convert.

---

**Note:** If you convert mirrored volumes, you must synchronize them in a separate step.

---



# Command differences

This chapter describes the differences between LVM and VxVM commands, and tasks. It includes a task comparison chart which lists some of the tasks performed using LVM with a *near equivalent* task performed using VxVM. It also provides a list of VxVM tasks which are not available with LVM, and the LVM features currently not supported in VxVM. The following topics are discussed in this chapter:

- [LVM and VxVM command equivalents](#)
- [Comparison of LVM and VxVM tasks](#)
- [Tasks with no direct LVM equivalents](#)
- [LVM features not supported in VxVM](#)

For more information on LVM commands, refer to *AIX Logical Volume Manager, from A to Z: Introduction and Concepts*. For information on VxVM commands, refer to the Veritas Volume Manager documentation.

## LVM and VxVM command equivalents

The table below lists the LVM commands and a *near equivalent* command to use in VxVM. For more information, refer to the Task Comparison chart. For information on VxVM commands, refer to the Veritas Volume Manager documentation package.

**Table 3-1** Command comparison

LVM	Description/action	VxVM	Description/action
chlv	Changes the characteristics of logical volumes.	vxedit or vxvol set	Creates, removes, and modifies Volume Manager records.
	There is no single equivalent LVM command.	vxresize	Resizes a file system and its underlying volume at the same time.
mklv	Creates a logical volume.	vxassist	Creates volumes with the <code>make</code> parameter.  Example: <code>vxassist make \ vol_name 100M layout=stripe</code>
extendlv	Increases disk space allocated to a logical volume.	vxassist	Increases a volume in size with the <code>growto</code> or <code>growby</code> parameter.  Example: <code>vxassist growto \ vol_name 200M,</code> <code>vxassist growby vol_name \ 100M</code>  <code>vxassist</code> creates and modifies volumes.
syncvg -l	Synchronizes mirrors that are stale in one or more logical volumes.	vxrecover  vxvol start	The <code>vxrecover</code> command performs resynchronize operations for the volumes, or for volumes residing on the named disks (medianame or the VxVM name for the disk).  Example: <code>vxrecover vol_name \ media_name</code>

**Table 3-1** Command comparison

LVM	Description/action	VxVM	Description/action
<code>lspv</code>	Displays information about physical volumes in a volume group.	<code>vxdisk list</code>	Lists information about VxVM disks.  Example: <code>vxdisk list \ disk_name</code>
<code>chpv</code>	Sets physical volume characteristics to allow/deny allocation of additional physical extents from this disk.	<code>vxdisk</code>  <code>vxdisk set</code>  <code>vxedit</code>	The <code>vxdisk</code> utility performs basic administrative operations on VxVM disks. Operations include initializing and replacing disks, as well as taking care of some book-keeping necessary for the disk model presented by the Volume Manager.
<code>chpv -C</code>	Removes the LVM header information and releases the disk from LVM control.	<code>vxdiskunsetup</code>	Removes the VxVM header information and releases the disk from VxVM control.
<code>mkvg</code>	Creates a volume group.	<code>vxdiskadd</code>    <code>vxvg init</code>	Creates a new disk group and/or adds disks to a disk group.
<code>lsvg</code>	Displays information on all volume groups.	<code>vxvg list</code>    <code>vxprint</code>	Displays the contents of a disk group.   Displays information about all objects or a subset of objects.
<code>chvg</code>	Activates or deactivates one or more volume groups.	<code>vxvg -g diskgroup set activation=mode</code>	Activates a shared disk group.
<code>extendvg</code>	Extends a volume group by adding one or more disks to it.	<code>vxdiskadd</code>    <code>vxdiskadm</code>	Adds a disk to the disk group.  Use option 1 in the <code>vxdiskadm</code> menu to add disks to the disk group.

**Table 3-1** Command comparison

LVM	Description/action	VxVM	Description/action
reducevg	Reduces a volume group by removing one or more disks from it.	vxvg rmdisk  vxdiskadm	Removes disks from a disk group.  Use option 2 in the vxdiskadm menu to remove disks.
lsvg	Scans all disks and looks for logical volume groups.	vxinfo  vxprint  vxdiskadm	Displays information about volumes.  Displays complete or partial information from records in VxVM disk group configurations.  Use option list in the vxdiskadm menu to display disk information.
syncvg	Synchronizes mirrors that are stale in one or more logical volumes.	vxrecover	Starts resynchronization and recovery of volumes.
reducevg	Removes the definition of a volume group from the system.	vxvg deport  vxdiskadm	Deports a disk group from the system.  Use option 9 in the vxdiskadm menu to remove a disk group.
exportvg	Removes a volume group from the system.	vxvg deport  vxdiskadm	Deports a disk group from the system.  Use option 8 in the vxdiskadm menu to remove a disk group.
importvg	Adds a volume group to the system by scanning physical volumes which have been exported using vgexport.	vxvg import  vxdiskadm	Imports a disk group.  Use option 7 in the vxdiskadm menu to import a disk group.

**Table 3-1** Command comparison

LVM	Description/action	VxVM	Description/action
	No LVM command	vxplex	Operates on plex objects.
chlv, extendlv, mklv, rmlv	Performs operations on logical volumes.	vxvol	Operates on volume objects.
	No LVM command	vxsd	Operates on subdisk objects.
	No LVM command	vxmend	Fixes simple misconfigurations.

## Comparison of LVM and VxVM tasks

This section contains a list of tasks which you can perform using LVM, and near equivalent tasks which you can perform using Veritas Volume Manager. You can perform the LVM tasks by using SMIT or the command line interface. Similarly, you can choose to perform VxVM tasks by using the VERITAS Enterprise Administrator (VEA) or the command line interface. This document focuses on the command line interface.

---

**Note:** The following features in VxVM require an additional license: *Mirroring, Mirroring and Striping, Dynamic Multipathing of Active/Active Devices, Hot-relocation, Online Migration, and RAID-5.*

---

For more information on LVM commands, refer to *AIX Logical Volume Manager from A-Z: Troubleshooting and Commands*. For information on VxVM commands, refer to the Veritas Volume Manager documentation package.

**Note:** VxVM is not rootable for this release, so you cannot perform these tasks on the root disk or root volume.

Mirroring in VxVM requires an additional license.

**Table 3-2** LVM and VxVM task comparison

Task type	Description	Example
VxVM	Bring a disk under Volume Manager control.	<code>vxdiskadd device_name</code> Use option 1 in the <code>vxdiskadm</code> menu to add a disk and initialize it.
LVM	Create a volume group	<code>mkvg /dev/vol_grp /dev/dsk/disk_name</code>
VxVM	Create a disk group.	<code>vx dg init disk_group disk_name</code> Use option 1 in the <code>vxdiskadm</code> menu to perform this task.
LVM	Add a new disk to the existing volume group.	<code>extendvg /dev/vol_grp \ /dev/rdisk/disk_name</code>
VxVM	Add a disk to an existing disk group.	<code>vx dg -g disk_group adddisk \ disk=devicename</code>

Example for a *disk\_group* = veritasdg, *medianame* = disk01, *vol\_name* = veritasvol, *plex name* = veritasvol-01, *subdisk* = disk01-01, *devicename* = hdisk5.

**Table 3-2** LVM and VxVM task comparison

Task type	Description	Example
LVM	Extend a logical volume or increase space allocated to a logical volume.	<pre>extendlv -l 50 /dev/<i>vol_grp</i>/<i>lvol_name</i></pre> <p>The <code>-l</code> option indicates the number of logical extents in the logical volume</p>
VxVM	Increase the volume by or to a given length.	<pre>vxresize -g <i>disk_group</i> -F vxfs \<i>vol_name</i> <i>length</i></pre> <pre>vxassist growto <i>vol_name</i> <i>new_length</i></pre> <pre>vxassist growby <i>vol_name</i> \<i>length_change</i></pre> <p>Grow the file system after growing the volumes.</p>
VxVM	Reduce a volume by or to a given length.	<pre>vxresize -g <i>disk_group</i> -F vxfs \<i>vol_name</i> <i>to_length</i></pre> <pre>vxassist -b shrinkby <i>vol_name</i> <i>length</i></pre> <pre>vxassist -b shrinkto <i>vol_name</i> <i>newlength</i></pre> <p>Shrink the file system before reducing the volume.</p>
LVM	Import and activate a volume group.	<pre>importvg /dev/<i>vol_grp</i> /dev/dsk/<i>disk_name</i></pre> <pre>varyonvg /dev/<i>vol_grp</i></pre>
VxVM	Import a disk group to make the specified disk group accessible on the local machine.	<pre>vxdg -tFC -n <i>newname</i> import \<i>disk_group</i></pre> <p>Use option 7 in the <code>vxdiskadm</code> menu to perform this task.</p>

Example for a *disk\_group* = veritasdg, *medianame* = disk01, *vol\_name* = veritasvol, *plex name* = veritasvol-01, *subdisk* = disk01-01, *devicename* = hdisk5.

**Table 3-2** LVM and VxVM task comparison

Task type	Description	Example
LVM	Export and deactivate an LVM volume group, and its associated logical volumes.	<code>varyoffvg <i>vol_group</i></code>  <code>exportvg /dev/<i>vol_group</i></code>
VxVM	Deport a disk group to disable access to the specified disk group. A disk group cannot be deported if any volumes in the disk group are currently open.	<code>vxvg deport <i>disk_group</i></code>  Option 9 in the <code>vxdiskadm</code> menu performs this task.
LVM	Restore volume group configuration to a particular physical volume.	<code>vgrestore -n /dev/<i>vol_grp</i> /dev/<i>rdsk/disk_name</i></code>
VxVM	Restore disk group configuration.	<code>vxconfigrestore</code>
LVM	Remove a volume group. This destroys a volume group by removing its last disk and removing it from <code>/etc/filesystems</code> .	<code>exportvg /dev/<i>vol_grp</i></code>  This is preceded by <code>lvremove</code> and <code>vgreduce</code> down to the last disk.
VxVM	Destroy a disk group.	<code>vxvg destroy <i>disk_group</i></code>
LVM	Extend a volume group by adding LVM disks to the volume group.	<code>extendvg /dev/<i>vol_grp</i> \ /dev/<i>disk_name</i></code>
VxVM	Add one or more disks to the disk group.	<code>vxdiskadd <i>disk_name</i></code>  Option 1 in the <code>vxdiskadm</code> main menu performs this task.
LVM	Reduce a volume group by reducing the number of disks in a volume group.	<code>reducevg /dev/<i>vol_grp</i> <i>disk_name</i></code>
VxVM	Remove a disk from disk group.	<code>vxvg -g <i>disk_group</i> -k rmdisk \ <i>disk_name</i></code>
LVM	Create a logical volume in LVM volume group.	<code>mklv /dev/<i>vol_grp</i></code>

Example for `disk_group = veritasdg`, `medianame = disk01`, `vol_name = veritasvol`, `plex name = veritasvol-01`, `subdisk = disk01-01`, `devicename = hdisk5`.

**Table 3-2** LVM and VxVM task comparison

Task type	Description	Example
VxVM	Create a volume of one of these layout types:  A concatenated volume  A striped mirror volume  A RAID-5 volume	<pre>vxassist make <i>vol_name</i> <i>length</i> vxassist make <i>vol_name</i> <i>length</i> \ layout=mirror, stripe vxassist make <i>vol_name</i> <i>length</i> \ layout=raid5</pre>
LVM	Display information about logical volumes.	<code>lslv /dev/<i>lvol_name</i></code>
VxVM	Display all volume information.  Display information about a specific volume.	<pre>vxprint -vt vxprint -ht <i>vol_name</i></pre>

Example for a *disk\_group* = veritasdg, *medianame* = disk01, *vol\_name* = veritasvol, *plex name* = veritasvol-01, *subdisk* = disk01-01, *devicename* = hdisk5.

**Table 3-2** LVM and VxVM task comparison

Task type	Description	Example
LVM	Display information about volume groups.	<code>lsvg -v /dev/vol_grp</code>
VxVM	Display disk group information.  Display information about a specific disk group.	<code>vxdisk list</code>  <code>vxprint -g disk_group</code>  <code>vxdg list</code>  <code>vxdisk list disk_group</code>
LVM	Display information about physical volumes.	<code>lspv disk_name</code>
VxVM	Display information about Volume Manager volumes.	<code>vxinfo</code> or <code>vxprint</code>
LVM	Remove a logical volume.	<code>rmlv /dev/vol_name</code>
VxVM	Remove a volume.	<code>vxedit rm vol_name</code>
LVM	Remove disks from a volume group or reduce the number of disks in the volume group.	<code>reducevg /dev/vol_grp disk_name</code>
VxVM	Remove disks from a disk group.	<code>vxdisk rm disk_group</code>  <code>vxdg rmdisk disk_name</code>  <code>vxdg -g group_name rmdisk disk_name</code>
LVM	Remove an entire volume group. Before attempting to remove the volume group, you must remove the logical volumes using <code>lvremove</code> , and all physical volumes except the last one using <code>vgreduce</code> .	<code>reducevg /dev/vol_grp last_disk</code>

Example for a *disk\_group* = veritasdg, *medianame* = disk01, *vol\_name* = veritasvol, *plex name* = veritasvol-01, *subdisk* = disk01-01, *devicename* = hdisk5.

**Table 3-2** LVM and VxVM task comparison

Task type	Description	Example
VxVM	Deport a disk group. You must unmount and stop any volumes in the disk group first.	<code>vxvg deport <i>disk_group</i></code>
LVM	Set up alternate links to a physical volume.  If a disk has two controllers, you can make one primary and the other an alternate link.	<code>mkvvg /dev/<i>vol_grp</i> <i>disk_name</i> \ <i>disk_name_2</i></code>  To remove the link: <code>reducevg /dev/<i>vol_grp</i> \ /dev/dsk/ <i>disk_name</i></code>
VxVM	The Multipathing disk devices in the Volume Manager represent virtual devices with one or more physical access paths to a particular physical disk. Dynamic Multipathing provides reliability of disk access by dynamically switching to another physical path in the event of failure of a path.	The DMP feature in VxVM sets up links automatically. It is not required to set up links separately.  Refer to the manual page <code>vxvmp(7)</code> and the <i>Veritas Volume Manager Administrator's Guide</i> for more information on DMP.
LVM	Create a mirrored logical volume.	<code>mklv -c 2 vgmirror hdisk1 hdisk2</code>
VxVM	Create a mirrored volume/plex or add a mirror to an existing volume.	<code>vxassist make <i>vol_name</i> <i>length</i> \ layout=mirror</code>
LVM	Reduce a single/double mirrored logical volume to an unmirrored logical volume.  Remove a mirrored logical volume.	<code>rmlvcopy <i>mirr_lv</i></code>  <code>rmlv /dev/<i>vol_grp</i>/<i>mirr_lv</i></code>

Example for a *disk\_group* = veritasdg, *medianame* = disk01, *vol\_name* = veritasvol, *plex name* = veritasvol-01, *subdisk* = disk01-01, *devicename* = hdisk5.

**Table 3-2** LVM and VxVM task comparison

Task type	Description	Example
VxVM	Remove mirrors or reduce the number of plexes/mirrors.  Remove a volume with the plexes associated with it.	<code>vxplex -o rm dis <i>plex_name</i></code>  <code>vxedit -rf rm <i>vol_name</i></code>
LVM	Increase the number of mirror copies.	<code>extendlv -m 2 /dev/<i>vol_grp</i>/<i>vol_name</i></code>
VxVM	Add mirrors to a volume or increase the number of plexes.	<code>vxassist mirror <i>vol_name</i></code>
LVM	Convert a mirrored logical volume into two logical volumes.  Split a logical volume.	<code>splitlvcopy lvol 2 hdisk1</code>
VxVM	Snapshot a volume and create a new volume.	<code>vxassist snapshot <i>vol_name</i> \ <i>new_vol_name</i></code>
LVM	Move a mirrored logical volume from one disk to another.	Use the <code>migratepv</code> command
VxVM	Move a plex.	<code>vxplex mv <i>orig_plex</i> <i>new_plex</i></code>
LVM	Synchronize a mirrored logical volume.  Synchronize extents within a mirrored logical volume.	<code>syncvg -l <i>lvol_name</i></code>
VxVM	Resynchronize operations for the given volumes.	<code>vxvol resync</code>
LVM	Synchronize extents within mirrored logical volumes in a volume group.	<code>syncvg /dev/<i>vol_grp</i></code>

Example for a *disk\_group* = veritasdg, *medianame* = disk01, *vol\_name* = veritasvol, *plex name* = veritasvol-01, *subdisk* = disk01-01, *devicename* = hdisk5.

**Table 3-2** LVM and VxVM task comparison

Task type	Description	Example
VxVM	Resynchronize operations for the named volumes, or for volumes residing on the named disks.  If no medianame or volume operands are specified, then the operation applies to all volumes.	<code>vxrecover -s vol_name</code>
LVM	Start a volume.	Use the <code>varyonvg</code> command
VxVM	Start a volume.	<code>vxrecover -s vol_name</code> <code>vxvol start vol_name</code>
LVM	Stop a volume.	Use the <code>varyoffvg</code> command
VxVM	Stop a volume.	<code>vxvol stop vol_name</code>

Example for a *disk\_group* = veritasdg, *medianame* = disk01, *vol\_name* = veritasvol, *plex name* = veritasvol-01, *subdisk* = disk01-01, *devicename* = hdisk5.

## Tasks with no direct LVM equivalents

The following table lists tasks which have no direct LVM equivalent. Most of these tasks can be performed either with the VERITAS Enterprise Administrator (VEA) or the command line interface.

**Note:** This list includes only common tasks, and is not exhaustive. For more information, refer to the *Veritas Volume Manager Administrator's Guide*, the *Veritas Cross-platform Data Sharing Administrator's Guide*, the *Veritas Intelligent Storage Provisioning Administrator's Guide*, and the *Veritas Enterprise Administrator User's Guide*.

**Table 3-3** Additional VxVM tasks with no LVM equivalents

Task description	Example
Hot-relocation designates a disk as a hot-relocation spare and allows the system to automatically react to I/O failure by relocating redundant subdisks to other disks. The <code>vxunreloc</code> utility can be used to restore the system to the same configuration that existed before the disk failure.	<code>vxedit set spare=on disk_name</code>  Alternatively, use menu option 12 of <code>vxdiskadm</code> to perform this task.
Rename a disk.	<code>vxedit rename old_diskname new_diskname</code>
Offline a disk.	<code>vxdisk offline disk_name</code>  Alternatively, use menu option 12 of <code>vxdiskadm</code> to perform this task.
Online a disk.	<code>vxdisk online disk_name</code>  Select menu option 10 of <code>vxdiskadm</code> .
Evacuate a disk.	<code>vxevac -g disk_group medianame \ new_medianame</code>
Replace a disk.	Select menu option 4 of <code>vxdiskadm</code> .
Recover volumes on a disk.	<code>vxrecover -g disk_group vol_name \ medianame</code>
Display a DMP node.	<code>vxdisk list meta_device</code>
Rename a disk group.	<code>vxdbg -tC -n newdgroup_name</code>
Rename a volume.	<code>vxedit -v rename name newname</code>  Update the <code>/etc/filesystems</code> file with the new name.

**Table 3-3** Additional VxVM tasks with no LVM equivalents

Task description	Example
Prepare a volume for DRL or instant snapshot operations.	<code>vxsnap prepare <i>vol_name</i></code>
Create a full-sized instant snapshot copy of a volume.	<code>vxsnap make \ source=<i>vol_name</i>/snapvol=<i>temp_vol_name</i></code>
Create a space-optimized snapshot copy of a volume.	<code>vxsnap make \ source=<i>vol_name</i>/newvol=<i>temp_vol_name</i>/ \  cache=<i>cache_object</i></code>
Recover a volume.	<code>vxrecover -g <i>disk_group volume</i> \ <i>medianame</i></code> <code>vxmend fix clean <i>plex_name</i></code>
Repair a mirror.	<code>vxplex att <i>plex_name</i></code>
Disable a mirror.	<code>vxplex det <i>plex_name</i></code>
Remove a log from a volume.	<code>vxassist remove log <i>vol_name</i></code>
Move a subdisk.	<code>vxsd mv <i>old_subdisk new_subdisk</i></code>
Decrease the disk space allocated to a logical volume with the <code>shrinkto</code> or <code>shrinkby</code> parameters.  <b>Note:</b> Make sure you shrink the file system before shrinking the volume.	<code>vxassist shrinkto <i>vol_name</i>  <i>new_size</i></code>
Remove one or more volumes from a volume group.	<code>vxedit -rf rm <i>vol_name</i></code>  <code>vxassist remove volume <i>vol_name</i></code>
Refresh the contents of an instant snapshot from a source volume.	<code>vxsnap refresh <i>snap_volume</i> \ source=<i>vol_name</i></code>
Reattach the plexes of a full-sized instant snapshot.	<code>vxsnap reattach <i>snap_volume</i> \ source=<i>vol_name</i></code>
Move volumes off a disk.	<code>vxevac</code>  Use option 7 in the <code>vxdiskadm</code> menu to move volumes.

**Table 3-3** Additional VxVM tasks with no LVM equivalents

Task description	Example
Move the contents of a subdisk onto new subdisks and replace the old sub disk with the new subdisks for any associations.	<code>vxsd mv</code>

## LVM features not supported in VxVM

Some of the features in LVM are not supported in the current release of VxVM. Given below is a table with the unsupported LVM features, and possible workarounds in VxVM.

**Table 3-4** LVM features and VxVM equivalents

LVM feature	VxVM equivalent
Physical volume groups	VxVM has no equivalent feature. The disk group feature of VxVM combines the logical volume group (VG) and physical volume group (PVG) of LVM.
Bad media block relocation.	VxVM relocates whole subdisks. Smaller granularity relocation is not supported. The bad block reallocation feature does not exist in VxVM because the vectoring of bad blocks is now done by most hardware.

# System Management Interface Tool (SMIT)

This chapter describes how to use the AIX System Management Interface Tool (SMIT) to administer VxVM, and how to launch SMIT from the command line.

The following topics are discussed in this chapter:

- [Launching SMIT](#)
- [Administering disk devices in SMIT](#)
- [Administering disk groups in SMIT](#)
- [Administering volumes in SMIT](#)

---

**Note:** Only privileged users can run SMIT.

---

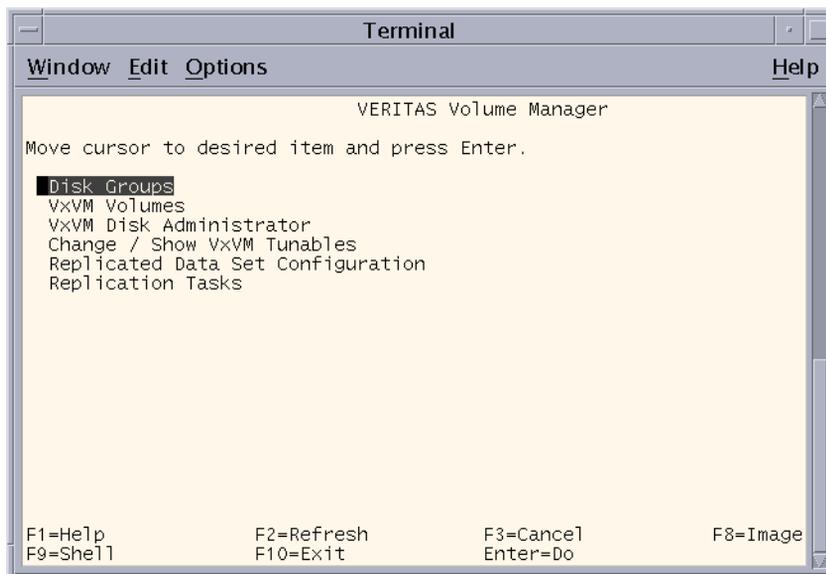
## Launching SMIT

Launch the SMIT interface to VxVM by entering the command:

```
$ smit vxvm
```

and you will be presented with the following top-level menu:

Figure 4-1 VxVM main menu



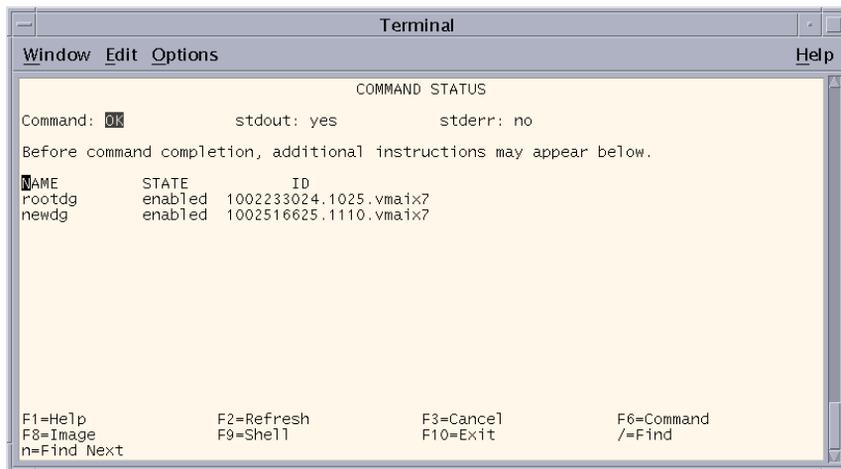
From this menu you can perform administrative tasks on VxVM components such as disk groups, disks and volumes, including:

- operations on VxVM disk groups (list/add/remove/modify/import/deport)
- operations on VxVM Volumes (list/add/remove/change/snapshot)
- a subset of the operations that are available through the `vxdiskadm` command
- changing VxVM tunables
- replication tasks

## Administering disk groups in SMIT

To list disk groups in SMIT, select from the Disks and File Systems SAM area, select **VxVM Disk Administrator->List all Disk Groups**. Output similar to that shown in [Figure 4-2](#) will be displayed:

**Figure 4-2** Listing disk groups



```
Terminal
Window Edit Options Help
COMMAND STATUS
Command: OK          stdout: yes          stderr: no
Before command completion, additional instructions may appear below.
NAME      STATE      ID
rootdg    enabled    1002233024.1025.vmaix7
newdg     enabled    1002516625.1110.vmaix7

F1=Help      F2=Refresh    F3=Cancel    F6=Command
F8=Image     F9=Shell     F10=Exit     /=Find
n=Find Next
```

The Disk Groups interface lists the VxVM disk groups on the system. In this example, there are two disk groups, `rootdg` and `newdg`.

## Administering disk devices in SMIT

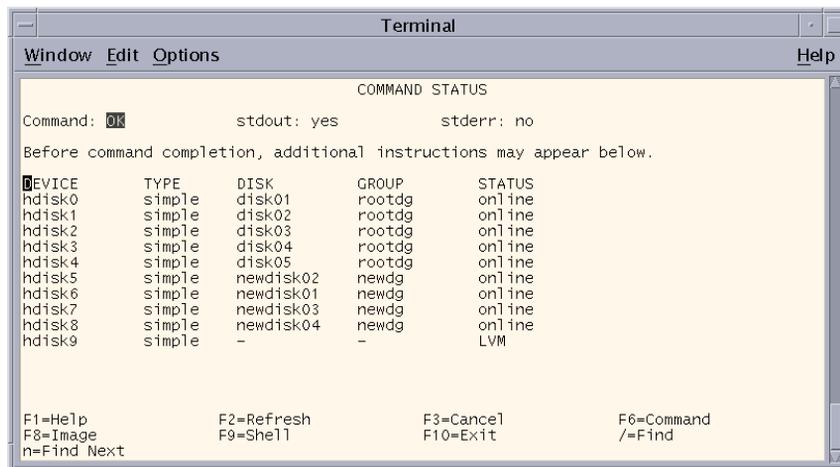
To list all disks on the system, select **VxVM Disk Administrator->List all disks in the system**, and output similar to that shown in [Figure 4-3](#) will be displayed.

When VxVM is installed on the system, SMIT includes a “STATUS” column to indicate whether a disk is under LVM or VxVM control, or whether it is unused.

If a VxVM disk is online and part of a disk group, the disk group name is listed under the “Group” column. If a VxVM disk is initialized, but not yet part of a disk group, the entries in the “Disk” and “Group” columns are blank.

In the example, devices `hdisk0` through `hdisk4` are part of the `rootdg` disk group, devices `hdisk5` through `hdisk8` are part of the `newdg` disk group, and device `hdisk9` is an LVM disk.

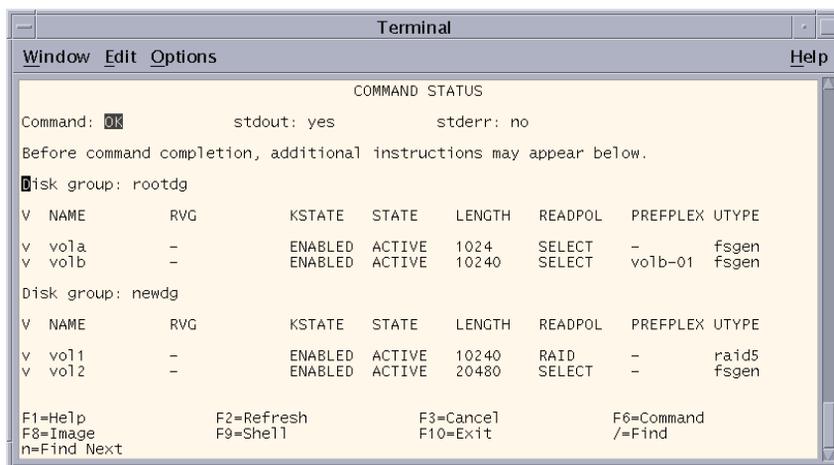
**Figure 4-3** Listing disks



## Administering volumes in SMIT

To list logical volumes in SMIT, select **VxVM Volumes** from the main menu, and then select **List VxVM Volumes in all or specific Disk Groups**. You will then be prompted to enter the name of a disk group. The screen shown in [Figure 4-4](#) is the result of entering **All** (The Default).

**Figure 4-4** Listing volumes in a system



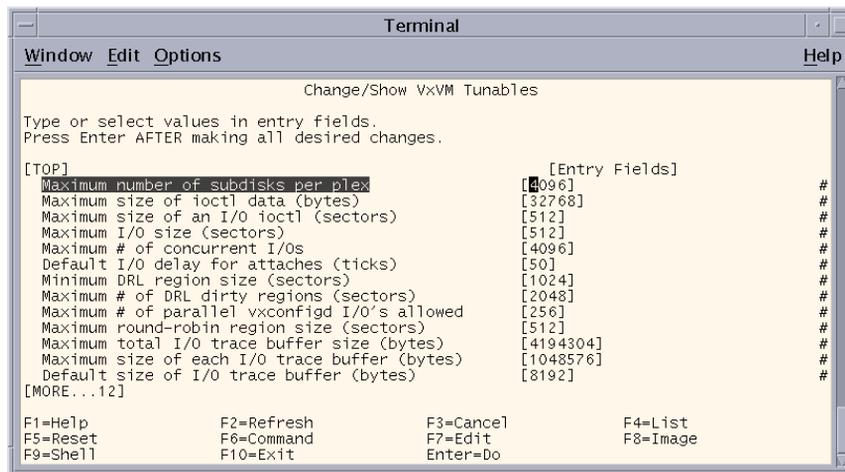
```
Terminal
Window Edit Options Help
COMMAND STATUS
Command: OK          stdout: yes          stderr: no
Before command completion, additional instructions may appear below.
Disk group: rootdg
V NAME      RVG      KSTATE  STATE  LENGTH  READPOL  PREFPLEX  UTYPE
v vol1a    -        ENABLED ACTIVE  1024    SELECT   -         fsgen
v vol1b    -        ENABLED ACTIVE  10240   SELECT   vol1b-01 fsgen
Disk group: newdg
V NAME      RVG      KSTATE  STATE  LENGTH  READPOL  PREFPLEX  UTYPE
v vol11    -        ENABLED ACTIVE  10240   RAID    -         raid5
v vol12    -        ENABLED ACTIVE  20480   SELECT   -         fsgen
F1=Help      F2=Refresh  F3=Cancel   F6=Command
F8=Image     F9=Shell    F10=Exit    /=Find
n=Find Next
```

This figure shows that the system consists of two disk groups, each of which in turn consists of two volumes. Full information on the displayed output can be found in the *Veritas Volume Manager System Administrator's Guide*.

## Administering VxVM tunables in SMIT

A final example of a SMIT interface is a screen that allows you to list and amend VxVM tunable parameters. From the main menu, select **Change/Show VxVM Tunables**, and a screen similar to that shown in [Figure 4-5](#) will be displayed.

**Figure 4-5**      Displaying tunable parameters and their current values



By using this display, you can configure your system to meet your own particular requirements.

For more information on Veritas Volume Manager, refer to the Veritas Volume Manager documentation package.

# Conversion error messages

This appendix lists the error messages that you may encounter when converting LVM volume groups to VxVM disk groups and volumes. For each error message, a description is provided of the problem, and the action that you can take to troubleshoot it.

[Table A-1](#) shows the error messages that you may encounter during conversion.

**Table A-1** Conversion error messages

Message	Description
Analysis indicates that this volume group cannot be converted because not all of the disks and/or volumes in the LVM volume group are currently accessible	<p>For successful conversion, all physical volumes in a volume group must be on-line, and all logical volumes must be active and accessible.</p> <p>Make sure the physical volumes in a volume group are on-line and the logical volumes are active and not in use.</p>
Analysis shows that there is insufficient private space available to convert this volume group	<p>The error message indicates the maximum amount of records that can be stored in the private space, and how many records are needed to convert this particular volume group.</p> <p>You can reduce the number of records needed by reducing the number of logical volumes in volume group by combining some of the logical volumes together.</p>

**Table A-1** Conversion error messages

Message	Description
The conversion process was unable to deactivate the volume group <i>vol_grp_name</i>	<p>This indicates that the conversion process cannot deactivate the volume group.</p> <p>The conversion cannot be completed without rebooting the machine. If you cannot afford to reboot, then choose abort and try again later.</p>
This Volume Group contains one or more logical volumes with mirrored data	<p>If you attempt to convert a Mirrored LVM Volume Group without a valid VxVM license installed, the conversion is not allowed.</p> <p>Install the required license before attempting the conversion.</p>
Too many LVM Volumes to convert in this LVM Volume Group	<p>If there is insufficient private space, the conversion is not allowed to continue. Also, the conversion records already generated are removed such that in the event of an unexpected crash and reboot, the conversion cannot proceed automatically.</p> <p>You can reduce the number of logical volumes in volume group by combining some of the logical volumes together, or by aborting. You can restart the conversion process later with fewer volumes in the group.</p>
vgchange: Couldn't deactivate volume group /dev/vol_grp	<p>The conversion process was unable to deactivate the volume group. The conversion cannot proceed without reboots being done. If you choose to not reboot your system, the conversion is aborted.</p> <p>The system responds with an option to complete the conversion by rebooting the system.</p>

**Table A-1** Conversion error messages

<b>Message</b>	<b>Description</b>
vxdiskadm or vxconvert is already being run and these programs cannot run concurrently	The system detects that the vxdiskadm or vxconvert program is already running. Retry at a later time. Otherwise, if you are certain that no other users are running either of these programs, remove the file <code>.DISKADD.LOCK</code> from the <code>/var/spool/locks</code> directory to allow you to run vxconvert.



# Glossary

**block**

A unit of space for data on a disk, typically having a size of 1024-bytes.

**Dirty Region Logging**

Dirty Region Logging (DRL) is an optional property of a volume, used to provide a speedy recovery of mirrored volumes after a system failure. DRL keeps track of the regions that have changed due to I/O writes to a mirrored volume.

**file system**

The organization of files on storage devices. The term file system can refer either to the entire file system or to a subsection of that file system, contained within a disk section or a logical volume that can be mounted or unmounted from that tree.

**logical volume**

A logical structure that is a map of storage areas on physical volumes. A logical volume can be conceptualized as a storage device of flexible size. The data in a logical volume can be mapped to one or more physical volumes. A virtual disk device that represents a contiguous virtual disk space that maps to single or multiple areas on a single or multiple physical volumes.

**LVM**

The Logical Volume Manager (LVM) is a subsystem for managing disk space. LVM is an AIX product, similar to VxVM.

**logical Extent**

A set of logical blocks that maps to one physical extent and is a basic unit of access in a logical volume.

**LVMREC**

The LVM record, which is an identifier that is set on each disk when it is initialized the first time by LVM. There are two copies of the LVMREC; one is at sector 8, while the other is at sector 72.

**Mirror Write Cache (MWC)**

A MirrorDisk/UX mechanism whose use is optional, that tracks outstanding mirror write requests and provides a basis for the resynchronization of data blocks after a system crash or power failure.

**physical volume**

A disk that has been initialized by LVM becomes known as a physical volume.

**public region**

A region of a physical disk managed by VxVM that contains available space and is used for allocating subdisks.

**private region**

A region of a physical disk used to store private, structured VxVM information. The *private region* contains a disk header, a table of contents, and a configuration database. The table of contents maps the contents of the disk.

**physical extent**

A set of physical disk blocks on a physical volume that forms a basic unit of access in LVM. This also forms the allocation unit for logical volumes.

**snapshot**

A temporary extra copy (plex/mirror) created in a volume. A separate volume is created once its contents are brought in sync with the original volume.

**set of continuous physical extents [PEs]**

Set of physical sectors (blocks) contained within a single physical volume. A physical extent is a specific, contiguous region of the disk where data resides. This is of a constant size usually of 4 MB and has no partitions.

**VGRA**

The Volume Group Reserved Area (VGRA) is a region on an LVM disk that holds LVM configuration information and is at a fixed location. The location of this fixed location depends upon whether the disk is a boot disk or a non-boot disk. For non-boot disks, the VGRA starts at sector 128.

**volume group**

The collective identity of a set of physical volumes, which provide disk storage for the logical volumes. A set of physical volumes whose space can be combined and logically divided up into logical volumes. Only logical volumes and physical volumes that are a part of a volume group can map together, a physical volume can belong to only one volume group.

# Index

## A

alternate links 49  
analyze 31

## C

coexistence  
    VxVM and LVM disks 14  
commands  
    vxedit 40  
configuration  
    LVM 15  
configuration VxVM 15  
conversion  
    errors 61  
    speed 37  
    vxconvert 17

## D

deactivate  
    disk group 46  
    volume group 46  
deport  
    disk group 46  
destroy  
    disk group 46  
disable  
    mirror 53  
disk  
    designate 52  
    disk space 19  
    evacuate 52  
    offline 52  
    online 52  
    recover  
        52  
    rename 52  
    replace 52  
disk group 44  
    rename 52  
disk groups 7

disk headers 16  
disks 7  
    coexistence 14  
display  
    disk group 47  
    DMP 52  
    logical volume 47  
    physical volume 47  
    volume 47  
    VxVM volumes 47  
DMP  
    display 52  
    Multipathing 49  
Dynamic Multipathing 43, 49

## E

equivalent command 40  
error messages  
    Analysis indicates that this volume group  
        cannot be converted 61  
    Analysis shows that there is insufficient  
        private space available to convert this  
        volume group 61  
    The conversion process was unable to  
        deactivate the volume group  
        vol\_grp\_name 62  
    This Volume Group contains one or more  
        logical volumes with mirrored data 62  
    Too many LVM Volumes to convert in this LVM  
        Volume Group 62  
    vgchange Couldn't deactivate volume group /  
        dev/vol\_grp 62  
    vxdiskadm or vxconvert is already being run  
        and these programs cannot run  
        concurrently 63  
Example  
    analyze LVM groups 29  
    conversion 29  
    failed conversion 29  
    list 29  
    list disk information 29

- list LVM volume group information 29
- listvg 29
- LVM to VxVM 29
- vxprint output 29
- example
  - Failed Analysis 29
- export
  - volume group 46
- extend
  - volume group 46

## F

- features
  - task monitor 8
- File System 23
- file system 45

## H

- Hot Relocation 43

## I

- import
  - disk group 45
  - volume group 45

## J

- join
  - subdisk 53

## L

- list LVM 30
- Logical Volume 11, 12
- logical volume
  - convert 50
  - split 50
  - synchronize 50
- Logical Volume Manager 7
- lvchange 40
- lvcreate 40
- lvextend 40
- LVM 7
  - metadata 18
- LVM names
  - symbolic names 25
- LVM VGRA 16
- lvsync 40

## M

- mapping
  - LVM device nodes 24
  - VxVM device nodes 24
- messages
  - error 61
- mirror
  - disable 53
  - disks 7
  - logical volume 49
  - plex 49
  - remove 53
  - repair 53
- MirrorDisk/UX 44
- mirroring 43
- mirroring and striping 43
- move
  - subdisk 53

## O

- Online Migration 43

## P

- physical volumes 11, 12
- Public Region 12
- pvchange 41
- pvdisplay 41

## R

- RAID-5 43
- reduce
  - volume group 46
- remove
  - disk 48
  - volume 48
  - volume group 46, 48
- rename
  - disk group 52
- repair
  - mirror 53
- restore
  - volume group 46
- resynchronize
  - volumes 50
- rootability
  - root disk 18
  - root volume 18

**S**

SAM  
 vgdisplay 21  
 SMIT 55  
 split  
   subdisk 53  
 subdisk  
   join 53  
   move 53  
   split 53

**T**

task monitor 8  
 tool  
   vxconvert 17  
 tools  
   vxconvert 15  
   vxdiskadm 15  
 troubleshoot  
   errors 61

**V**

Veritas Volume Manager 7  
 vgchange 41  
 vgcreate 41  
 vgdisplay 41  
 vgexport 42  
 vextend 41  
 vgimport 42  
 VGRA 12  
 vgreduce 42  
 vgremove 42  
 vgscan 42  
 vgsync 42  
 volume  
   concatenated 47  
   logical 47  
   RAID-5 47  
   striped 47  
 Volume Manager  
   features 8  
 volumes 7  
   Logical Volume 11, 12  
   physical volumes 11, 12  
 vxassist 40  
 vxconvert 31  
 vxdg 41  
 vxdisk 41

vxdisk set 41  
 vxdiskadd 41  
 vxedit 40, 41  
 VxFS 45  
 vxinfo 42  
 vxmend 43  
 vxplex 43  
 vxprint 41, 42  
 vxrecover 40  
 vxresize 40  
 VxVM 7  
   features 8  
   metadata 18  
 VxVM names  
   symbolic link 25  
 VxVM volumes  
   resynchronize 50  
 vxxvol 40, 43

