

VERITAS Volume Manager 4.1

Migration Guide

HP-UX

N12981G

June 2005

Disclaimer

The information contained in this publication is subject to change without notice. VERITAS Software Corporation makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. VERITAS Software Corporation shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual.

VERITAS Legal Notice

Copyright © 2005 VERITAS Software Corporation. All rights reserved. VERITAS, the VERITAS Logo and VERITAS Storage Foundation are trademarks or registered trademarks of VERITAS Software Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

VERITAS Software Corporation
350 Ellis Street
Mountain View, CA 94043
USA
Phone 650-527-8000 Fax 650-527-2908
www.veritas.com

Third-Party Legal Notices

Data Encryption Standard (DES) Copyright

Copyright © 1990 Dennis Ferguson. All rights reserved.

Commercial use is permitted only if products that are derived from or include this software are made available for purchase and/or use in Canada. Otherwise, redistribution and use in source and binary forms are permitted.

Copyright 1985, 1986, 1987, 1988, 1990 by the Massachusetts Institute of Technology. All rights reserved.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided as is without express or implied warranty.

Contents

Preface	v
How This Guide is Organized	v
Related Documentation	vi
Conventions	vii
Getting Help	viii
Documentation Feedback	viii
Chapter 1. VxVM and LVM	1
Introducing the VERITAS Volume Manager	1
VxVM and LVM—Conceptual Comparison	4
Coexistence of VxVM and LVM Disks	7
Chapter 2. Converting LVM to VxVM	9
Converting Unused LVM Physical Volumes to VxVM Disks	10
Removing LVM Disk Information	10
Initializing Disks for VxVM Use	11
Converting LVM Volume Groups to VxVM Disk Groups	11
Volume Group Conversion Limitations	12
Conversion Process Summary	14
Conversion Steps Explained	15
Restoring the LVM Volume Group Configuration	22
Rollback to LVM Using vxvmconvert	23
Full LVM Restoration	24
Examples	25



Example: displaying the vxvmconvert menu	25
Example: listing disk information	26
Example: listing LVM volume group information	26
Example: analyzing LVM volume groups	28
Example: converting LVM volume groups to VxVM disk groups	31
Example: list, listvg, and vxprint outputs of an LVM volume group before and after conversion	35
Example: VxVM to LVM rollback	38
General Information Regarding Conversion Speed	39
Chapter 3. Command Differences	41
LVM and VxVM Command Equivalents	42
Comparison of LVM and VxVM Tasks	47
Tasks With No Direct LVM Equivalents	57
Existing Features in LVM not supported in VxVM	59
Chapter 4. SAM and the VEA	61
Listing Disk Devices in SAM	62
Listing Volume Groups and Disk Groups in SAM	63
Listing Logical Volumes in SAM	64
Appendix A. Conversion Error Messages	65
Glossary	69
Index	73



Preface

This guide describes how to migrate from HP-UX Logical Volume Manager (LVM) to VERITAS Volume Manager (VxVM) and convert disks and volumes managed by LVM to VxVM. It includes information on differences between VxVM and LVM commands and terminology. This Guide also discusses the use of both graphical user interfaces: VERITAS Enterprise Administrator (VEA), the Volume Manager GUI, and SAM (System Administration Manager), the Logical Volume Manager GUI.

This Guide assumes:

- ◆ A basic understanding of system administration
- ◆ Familiarity with HP-UX Logical Volume Manager and related concepts
- ◆ A working knowledge of the HP-UX operating system

How This Guide is Organized

This guide is organized as follows:

- ◆ [“VxVM and LVM”](#) compares the VERITAS Volume Manager to the HP Logical Volume Manager conceptually.
- ◆ [“Converting LVM to VxVM”](#) describes the procedures for identifying and converting an LVM configuration to a VxVM configuration.
- ◆ [“Command Differences”](#) describes the differences between LVM and VxVM commands. It also lists LVM tasks and equivalent or similar VxVM tasks.
- ◆ [“SAM and the VEA”](#) describes the usage of VERITAS Volume Manager’s GUI (VEA) and its relationship with HP-UX SAM (System Administration Manager).
- ◆ [“Conversion Error Messages”](#) describes the error messages that may be encountered during conversion and contains troubleshooting procedures.
- ◆ [“Glossary”](#) defines terms related to the Volume Manager and LVM.



Related Documentation

The following documents provide information about LVM on HP-UX:

- ◆ *HP-UX Managing Systems and Workgroups*
- ◆ LVM on HP-UX manual pages in reference volumes 2, 3, and 5
- ◆ HP-UX SAM online help
- ◆ *Disk and File Management Tasks on HP-UX* by Tom Madell
ISBN 0-13-518861-X, HP Press: Prentice-Hall, 1997.



Conventions

Convention	Usage	Example
monospace	Used for path names, commands, output, directory and file names, functions, and parameters.	Read tunables from the <code>/etc/vx/tunefstab</code> file. See the <code>ls(1)</code> manual page for more information.
monospace (bold)	Indicates user input.	# ls pubs C:\> dir pubs
<i>italic</i>	Identifies book titles, new terms, emphasized text, and variables replaced with a name or value.	See the <i>User's Guide</i> for details. The variable <i>system_name</i> indicates the system on which to enter the command.
bold	Depicts GUI objects, such as fields, list boxes, menu selections, etc. Also depicts GUI commands.	Enter your password in the Password field. Press Return .
blue text	Indicates hypertext links.	See " Getting Help " on page viii.
#	Unix superuser prompt (all shells).	# cp /pubs/4.0/user_book /release_mgnt/4.0/archive
C:\>	Windows user prompt.	C:\> copy \pubs\4.0\user_book c:\release_mgnt\4.0\archive



Getting Help

For technical assistance, visit <http://support.veritas.com> and select phone or email support. This site also provides access to resources such as TechNotes, product alerts, software downloads, hardware compatibility lists, and the VERITAS customer email notification service. Use the Knowledge Base Search feature to access additional product information, including current and past releases of product documentation.

Diagnostic tools are also available to assist in troubleshooting problems associated with the product. These tools are available on disc or can be downloaded from the VERITAS FTP site. See the `README.VRTSspt` file in the `/support` directory for details.

For license information, software updates and sales contacts, visit <https://my.veritas.com/productcenter/ContactVeritas.jsp>. For information on purchasing product documentation, visit <http://webstore.veritas.com>.

Documentation Feedback

Your feedback on product documentation is important to us. Send suggestions for improvements and reports on errors or omissions to foundation_docs@veritas.com. Include the title and part number of the document (located in the lower left corner of the title page), and chapter and section titles of the text on which you are reporting. Our goal is to ensure customer satisfaction by providing effective, quality documentation. For assistance with topics other than documentation, visit <http://support.veritas.com>.

This chapter provides an overview of the VERITAS Volume Manager (also referred to as VxVM) and its features. A brief description of the benefits of migrating from the HP-UX Logical Volume Manager (LVM) to VxVM, and the coexistence of VxVM disks with LVM disks is also given.

The following topics are discussed in this chapter:

- ◆ [Introducing the VERITAS Volume Manager](#)
- ◆ [VxVM and LVM—Conceptual Comparison](#)
- ◆ [Coexistence of VxVM and LVM Disks](#)

Introducing the VERITAS Volume Manager

The VERITAS Volume Manager is an alternative Volume Management product for HP-UX that includes mirroring features. It offers many capabilities that are not available with the LVM and MirrorDisk/UX products today.

- ◆ The 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, `vxvmconvert`, is described in detail in [Converting LVM to VxVM](#).
- ◆ With HP-UX 11i Version 2, the VERITAS Volume Manager is available for installation with the HP-UX 11i Version 2 Application Software. See the *Release Notes* for details of features supported in this release.
- ◆ With HP-UX 11i Version 2, the VERITAS Volume Manager is the default volume manager and can be used instead of LVM to manage the root disk. Basic volume management capabilities are included in the operating system.
- ◆ The VERITAS Volume Manager is integrated with HP MC/ServiceGuard and ServiceGuard OPS Edition for High Availability, but requires a specific version of the ServiceGuard products. Refer to the *Release Notes* for details about the required version number, as well as the availability of specific features in your release.



Notable Features of VxVM

The VERITAS Volume Manager provides many features, some of which are not available with LVM or MirrorDisk/UX. Notable VxVM features are described in the list below. See the *Release Notes* for a more detailed list of features available in each VERITAS Volume Manager product. See the other VERITAS Volume Manager documents (listed under [“Related Documentation”](#) in the [“Preface”](#)) for more details about using these features.

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 (VEA), which is a JAVA-based GUI for VxVM.
- ◆ Dynamic Multipathing (DMP) for active-passive devices, such as FC60. 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.
- ◆ 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.
- ◆ Dynamic Multipathing (DMP) for active-active devices, such as HP Surestore Disk Array xp256, HP Surestore Disk System FC10 and other disk 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 xp256, EMC Symmetrix disk array, and other OEM array devices.
- ◆ Multiple mirroring with up to 32 mirror copies of a volume’s address space.
- ◆ Mirrored stripes (RAID-0 + RAID-1) and striped mirrors (RAID-1 + RAID-0) combine the benefits of striping to improve performance by spreading data across multiple disks, and mirroring to provide redundancy of data. Striped mirror

volumes are more tolerant of disk failure and have a shorter recovery time than mirrored stripe volumes. Refer to the *VERITAS Volume Manager 3.5 Administrator's Guide* for more detailed information on these layouts.

- ◆ 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.
- ◆ 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.

Note For more information on LVM, refer to *HP-UX Managing Systems and Workgroups*, and LVM manual pages in *HP-UX Reference Volumes 2, 3, and 5*. For information on VxVM commands, refer to the VERITAS Volume Manager documentation.



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. 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 HP-UX 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.</p> <p>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, but mirrored-stripe and striped-mirror layouts are not supported.</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 concatenated, mirrored, striped, RAID-5 or combinations such as mirrored-stripe, striped-mirror, and concatenated-mirror.</p>	

Table 1. A Conceptual comparison of LVM and VxVM

LVM Term	VxVM Term
Volume Group	Disk Group
<p>LVM volume groups are conceptually similar to VxVM disk groups.</p> <p>An LVM volume group is the collective identity of a set of physical volumes, which provide disk storage for the logical volumes.</p> <p>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.</p> <p>In addition, both LVM and VxVM have the following characteristics:</p> <ul style="list-style-type: none"> ◆ 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
<p>User data is contained in physical extents in LVM and subdisks in VxVM.</p> <p>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, and is usually 4 MB.</p> <p>VxVM allocates disk space in term of subdisks which is a set of physical disk blocks representing a specific contiguous portion of a VxVM disk and is of arbitrary size.</p>	
LVM metadata	Private Region
<p>LVM metadata and the Private Region are similar conceptually.</p> <p>In LVM, metadata is stored in a reserved area in the disk.</p> <p>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.</p>	
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>	



Table 1. A Conceptual comparison of LVM and VxVM

LVM Term	VxVM Term
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>In LVM, you can create mirrors using the MirrorDisk/UX product. Mirrors allow duplicate copies of the extents to be kept on separate physical volumes. MirrorDisk/UX supports up to 3 copies of the data.</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 from <code>/etc/lvmtab</code>. The volume group must have already been deactivated.</p> <p>Similarly in VxVM, <code>deport</code> makes a disk group inaccessible by the system.</p>	
Import	Import
<p>In LVM, <code>import</code> adds a volume group to the system and the volume group information to <code>/etc/lvmtab</code> but does not make the volumes accessible. The volume group must be activated by the <code>vgchange -a y</code> command in order to make volumes accessible.</p> <p>In VxVM, <code>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.</p> <p>The bad block reallocation feature does not exist in VxVM because the vectoring of bad blocks is now done by most hardware.</p>	
<code>/etc/lvmtab</code>	No similar term
<p>The <code>/etc/lvmtab</code> file contains information about volume groups that are accessible by a system.</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.

Both LVM and VxVM utilities are aware of the other volume manager, and will not overwrite disks that are being managed by the other volume manager unless conversion or rollback is being performed between LVM logical volumes and VxVM volumes. The administrative utilities (SAM and VEA) recognize and identify all disks on the system (see [“SAM and the VEA”](#) on page 61, for details).

The `vxvmconvert` command is provided to enable LVM disks to be converted to a VxVM disk format without losing any data. For more information, see [“Converting LVM to VxVM”](#) on page 9.





Converting LVM to VxVM

2

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

- ◆ [Converting Unused LVM Physical Volumes to VxVM Disks](#)
- ◆ [Converting LVM Volume Groups to VxVM Disk Groups](#)
- ◆ [Restoring the LVM Volume Group Configuration](#)
- ◆ [Examples](#)

The basic tools for conversion are the VxVM commands, `vxvmconvert` and `vxdiskadm`, and the LVM administrative utilities such as `pvremove` and `vgcfgbackup`. 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 `pvcreate`. Conversion is done differently for these two types of disks.

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



Converting Unused LVM Physical Volumes to VxVM Disks

LVM disks which are not part of any volume group, and contain no user data, are simply cleaned up, so that there are no LVM disk headers. Then the disks are given over to VxVM through the normal means of initializing disks.

Caution Exercise caution while using this procedure to give disks over to VxVM. 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 conversion.

Removing LVM Disk Information

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

```
# pvremove disk_name
```

The `pvremove` command will not allow the removal of disk headers which indicate a Volume Group membership for the disk. If the disk fails `pvremove` for this reason, you should first make certain that the group membership information is stale. Do this by using `pvdisplay`:

```
# pvdisplay disk_name
```

If `pvdisplay` finds no valid group information associated with the disk, you can overwrite the LVM headers using `pvcreate`:

```
# pvcreate disk_name
```

Caution If *disk_name* is an alternate path to a disk that does not appear in the `lvmtab` file for this system, or is a disk that is in use on another system, but not imported onto this system, then do not `pvcreate -f` the disk. Doing so will destroy the LVM headers. You can use `pvcreate` without the `-f` option safely, as it will only succeed if the disk is not listed in the `/etc/lvmtab` file, and if the LVM headers indicate that it is not a member of any volume group. (That is, it has been `pvcreated` but not been an argument for `vgcreate` or `vgextend`.)

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 disk headers). If you attempt to use `vxdisk init`, or `vxdiskadm` on an LVM disk without using the `pvremove` command first, the command will fail.

Converting LVM Volume Groups to VxVM Disk Groups

This section outlines the process for 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.

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 `vxvmconvert`. This interactive, menu-driven program walks you through many of the steps of the process of converting volume groups for use by VxVM. Using `vxvmconvert` can reduce the downtime associated with converting from LVM to VxVM. Without the `vxvmconvert` 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 `vxvmconvert` process converts LVM volume groups to VxVM disk groups *in place*. In reality, 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 file systems, databases, etc., are not affected by the conversion.

The act of conversion changes the names by which your system refers to the logical storage. For this reason, the conversion process is necessarily an off-line one. 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.

Note The most likely scenario in which a Volume Group cannot be converted, because of insufficient private space, is when a large HP-UX system using “Extent based Striping” is being used

- ◆ A volume group containing the root volume.

`vxvmconvert` does not convert any volume group that contains a *rootable* volume, identified by the presence of the LIF area as created by `mkboot(1M)`. 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.

Note You can use the `vxcp_lvmroot(1M)` command to create a VxVM root disk on a spare physical disk from the contents of the existing LVM root disk. Similarly, you can use the `vxres_lvmroot(1M)` command to recreate an LVM root disk on a spare disk from the contents of the VxVM root disk. For more information, see the section “Rootability” in the “Administering Disks” chapter of the *VERITAS Volume Manager Administrator’s Guide*.

- ◆ A volume group containing mirrors using the Mirror Write Cache feature for volume consistency recovery.

Users 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 `vxvmconvert` utility can convert these volumes, but must use the Dirty Region Logging (DRL) feature to obtain the same level of functionality. However, since Dirty Region Logging requires some user space to be available for the log, a conversion could fail due to an MWC volume being full, leaving no space for the DRL log. However it is very unlikely that this situation would occur. Note that the MWC and DRL are used only when the system crashes or is improperly shut down, to quickly bring all mirrors in the volume back into a consistent state.

- ◆ A volume group containing the `/usr` file system.

For this release, a volume group containing the `/usr` file system cannot be converted because `vxvmconvert` needs access to files in `/usr`.

- ◆ Volume groups with any dump or primary swap volumes.

`vxvmconvert` will not convert any volume group with dump or primary swap volumes. These are volumes known to the boot process. However, swap volumes on volumes other than the root volume can be converted (as long as this volume is not in the same volume group as the root volume).

- ◆ Volume group disks used in MC/ServiceGuard clusters.

The conversion process does not support conversion of any volume group that is marked as a member of a MC/ServiceGuard or OPS Edition high availability cluster. The volume group must be deactivated and removed from membership in the high availability cluster before it can be converted.

- ◆ Volume groups used for cluster lock disks.

The conversion process does not support conversion of a volume group that contains a disk that is being used for a cluster lock disk for an MC/ServiceGuard cluster.

- ◆ 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.

- ◆ 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.

The `analyze` option in `vxvmconvert`, which is described in later sections, aids you in identifying which volume groups can be converted.



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 `vxvmconvert` 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 steps in the conversion process are:

1. Identifying LVM volume groups for conversion.
2. Analyzing an LVM group to see if conversion is possible.
3. Taking actions to make conversion possible if analysis fails.
4. Backing up your LVM configuration and user data.
5. Planning for new VxVM logical volume names.
6. Stopping application access to volumes in the volume group to be converted.
7. Converting a volume group.
8. Taking actions if conversion fails.
9. Implementing changes for new VxVM logical volume names
10. Restarting applications on the new VxVM volumes.
11. Tailoring your VxVM configuration.

These steps are described in detail in later sections of this chapter. Annotated examples on how to use `vxvmconvert` are shown in [“Examples”](#) on page 25. For information on restoring back to your original LVM configuration refer to [“Restoring the LVM Volume Group Configuration”](#) on page 22.

Conversion Steps Explained

1. Identifying 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 `vgdisplay` and SAM can help you identify candidate LVM volume groups as well as the disks that comprise them.

You can also use the `vxvmconvert` and `vxdisk` commands to examine groups and their member disks. The information presented through the `vxvmconvert` and `vxdisk` utilities and their interpretation is shown in “[Examples](#)” on page 25.

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

```
# vxdisk list
```

2. Analyzing 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 `vxvmconvert` 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 12.

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 don’t have the required space to convert the disks, the conversion would fail.

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

Note The analysis option is presented as a separate menu item in `vxvmconvert`, but there is an implicit analysis with any conversion. If you simply select the “Convert LVM Volume Groups to VxVM” menu option, `vxvmconvert` 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 11.

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](#)” on page 25.

3. Taking actions 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 `vxvmconvert` will explain the type of failure and any actions that can be taken before retrying the analysis. Refer to “[Conversion Error Messages](#)” on page 65 for complete details of specific error messages and actions.

4. Backing 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. Similarly, you should back up the LVM configuration itself.

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.

Backing up an LVM configuration

Use the `vgcfgbackup(1M)` utility before running `vxvmconvert` to save a copy of the LVM configuration.

You can back up the LVM volumes using the following command:

```
# vgcfgbackup -f pathname/filename vol_grp_name
```

Be sure to use the `-f` option to save the data into a file other than the default. `vxvmconvert` uses LVM utilities which themselves save the configuration using `vgcfgbackup`. If you do not use the `-f` option when you attempt to backup the configuration, the conversion process will overwrite your attempted backup.

A copy of this LVM configuration should be kept off-line on tape or some other medium for use in the event of a disaster during conversion.

For example, to put a copy on tape, use the following command:

```
# tar cvf /dev/rmt/c3t0d0BEST /vgbackups/vg08
```

Note The `vxvmconvert` utility itself also saves a snapshot of the LVM metadata in the process of conversion for each disk. This data is saved in a different format from that of `vgcfgbackup`. It can only be used via the `vxvmconvert` program. With certain limitations, you can reinstate the LVM volumes after they have been converted to VxVM using this data (see “[Example: displaying the vxvmconvert menu](#)” on page 25). Even though `vxvmconvert` provides this level of backup of the LVM configuration, you are advised to use `vgcfgbackup` before running `vxvmconvert`.

Backing up user data

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

Caution Before you do the backup, you should carefully review “[step 9. Implementing changes for new VxVM logical volume names](#).” 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 `fbackup(1M)` command can be used to back up the data to tape.

For example, to backup the data on `/dev/vg01/lvol3` mounted on `/foodir`, use the following command:

```
# fbackup -0i /foodir -f /dev/rmt/c0t0d0BEST
```

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.



5. Planning 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 `/etc/fstab`. To handle this problem, `vxvmconvert` will rewrite the `fstab` with the new VxVM names when conversion is done 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. `vxvmconvert` 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 labeling of the backups may record device names.
- ◆ Scripts run by `cron(1M)`.
- ◆ Other administrative scripts.

A Workaround

`vxvmconvert` 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/rdsk/vol1` through a symbolic link `/dev/vg00/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.

6. Stopping 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, etc.

Note If you are converting a volume with swap space on it, the conversion requires a reboot. The swap space cannot be taken out of control of the operating system with a shutdown to single user mode.

As described in “[Conversion and Reboot](#)” on page 19, `vxvmconvert` tries to unmount mounted file systems during the conversion. Bear in mind though, that `vxvmconvert` makes no attempt to close down running applications on those file systems, nor does it attempt to deal with applications (e.g., databases) running on raw LVM volumes.

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

To unmount a file system, use the following command:

```
# umount file-system
```

Conversion and Reboot

During conversion, after the analysis phase is complete, the disks to be converted are deemed to be *conversion ready*. The `vxvmconvert` 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 `vxvmconvert` 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, `vxvmconvert` 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, `vxvmconvert` will return to the main menu.

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

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:

```
# vgchange -a y vol_grp_name
```

7. Converting a volume group

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

`vxvmconvert` 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 `vxvmconvert`. For details on modifying VxVM volume names, see “[step 11. Tailoring your VxVM configuration](#),” on page 21.

As described earlier in “[step 2. Analyzing an LVM volume group to see if conversion is possible](#),” on page 15, the volume groups selected for conversion are analyzed to ensure that conversion is possible. After a successful analysis phase, `vxvmconvert` 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](#)” on page 25.

8. Taking actions if conversion fails

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

See “[Conversion Error Messages](#)” on page 65 for complete details of specific error messages.

9. Implementing 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 “[step 5. Planning for new VxVM logical volume names](#) on page 18” for details.

10. Restarting 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.

If you unmounted file systems before running `vxvmconvert`, you need to remount them by the new volume names. `vxvmconvert` will have updated `/etc/fstab` with the new names. When you started `vxvmconvert`, you may have left file systems mounted that are associated with the volumes you converted. `vxvmconvert` remounts these with the new VxVM volume names.

11. Tailoring your VxVM configuration

`vxvmconvert` 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, `vxvmconvert` 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, `vxvmconvert` 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`, etc. The VxVM plexes within the logical volumes will be `dg0801-01`, `dg0801-02`, etc.

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. In particular, you should never use menu option 3 of `vxvmconvert` (Roll back) after name changes. If you have chosen to set up symbolic links to the VxVM volumes as described in “step 5. [Planning for new VxVM logical volume names](#),” avoid renaming VxVM objects. These symbolic links are made invalid if the underlying VxVM device node name changes.

Restoring the LVM Volume Group Configuration

In some circumstances, you may need to restore the LVM configuration that existed before you converted to VxVM with `vxvmconvert`. For example:

- ◆ If something went wrong during the conversion, such as a system crash or a disk crash that caused the conversion to be unworkable.
- ◆ If during a conversion only some of a set of volume groups converted successfully, then you may want to restore the LVM configuration for the entire set.

It is possible to restore the original LVM configuration in one of two ways, but both have limitations and restrictions. The method you use depends on if any changes have been made to the VxVM configuration since the conversion occurred. Any of the following actions changes the VxVM configuration:

- ◆ adding or removing disks
- ◆ adding or removing volume groups
- ◆ changing the names of VxVM objects

Restoration methods include:

- ◆ rollback using `vxvmconvert`

Use rollback only if the VxVM configuration has not changed since the conversion. This method restores the LVM configuration without the need for user data restoration. See “[Rollback to LVM Using vxvmconvert](#)” on page 23 for details on using this method.

- ◆ restore user data using `vgrestore` and `frecover`

This method is a full LVM restoration which is used to restore your user data from backup when the VxVM configuration has changed since the conversion was made. First of all, this method restores the original LVM configuration information (`vgrestore`), and then restores the original user data from the backup that was made before the conversion was done (`frecover`). See “[Full LVM Restoration](#)” on page 24 for more information on using this method.

Note Restoring user data using the `vgrestore` and `frecover` method will result in the loss of all user data changes made since the conversion, and the loss of all new volumes created since the conversion.

In other words, this method of restoring data will take you back to exactly where you were before the conversion was done.

However, if no new volumes have been created, and if none of the existing volumes have been resized, you can use the `vxvmconvert rollback` option to restore the original LVM configuration. If you use this method, any user data changes made since the conversion will be retained, and you will not need to carry out a user level data restore (`frecover`).

The name changes that `vxvmconvert` makes as part of the conversion are managed by rollback, and do not count as VxVM configuration changes for the purposes of choosing a restoration method.

The `vgrestore` command should not be confused with the LVM command, `vgcfgrestore`. `vgcfgrestore` is used to restore the LVM configuration information saved by `vgcfgbackup`, but it will not restore your device files and `/etc/fstab` entries. It also will not import and activate the volume group, nor will it clean up any VxVM information left around. However, `vgrestore` will do all of this for you.

Rollback to LVM Using `vxvmconvert`

Rollback replaces the VxVM disk groups with the original LVM volume groups. During conversion, `vxvmconvert` saves a “snapshot” of the original LVM metadata and associated configuration files, such as `/etc/fstab` and LVM device files. It restores only the LVM metadata and configuration files from this snapshot; user data is not changed. This method can only be used if no changes have been made to the configuration since the conversion.

For example, if a disk has been added to the disk group or if the names of any logical volumes have changed, you cannot use the rollback method.

Note In many cases, if you choose the rollback method and the configuration has changed, you receive an error and must use the full restore method.

If you used the workaround of creating symbolic links from the old LVM names to the new VxVM names described in “[step 5. Planning for new VxVM logical volume names](#),” you must remove the symbolic links you created before beginning the rollback.



This “snapshot” is kept on the root file system. The presence of this snapshot should not be taken as assurance that full off-line backups will not be needed. See “4. Backing up your LVM configuration and user data” for specific information on backups.

To rollback to LVM from the VxVM conversion, run `vxvmconvert` and choose option 3. See “[Example: VxVM to LVM rollback](#)” on page 38 for illustration.

Caution Do not use this option unless you are certain that you want to restore LVM volume groups. Once this is run, the VxVM disks that were created as a result of the original conversion from LVM to VxVM no longer exists. This option is *not* a full complement to `vxvmconvert`. It simply writes the saved LVM metadata back on top of the disks. Those data can only be considered valid for the period of time when the logical volumes are off-line. If the VxVM configuration has been brought online, the metadata in the rollback snapshot should be considered obsolete. See “[Full LVM Restoration](#)” on page 24 for specific information.

Full LVM Restoration

If you need to restore the original LVM configuration, but changes have been made to the VxVM configuration, you cannot use the rollback option of `vxvmconvert`. In this case, you must restore the user data in addition to restoring the old LVM metadata and associated configuration files. You may need to use this method if the disks in use by the LVM/VxVM volumes were corrupted during or after conversion.

Note The snapshot of LVM internal data is kept on the root filesystem.

To use this method, 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.

The process is limited to restoring the state of the logical volumes as they existed prior to conversion to VxVM disks. If the data has changed on the volumes during the time they were VxVM volumes, those changes are lost once you restore the LVM configuration and saved user data.

To do a full restoration of the original LVM configuration, do the following:

1. Use `vgrestore` to restore LVM internal data.

```
# vgrestore vol_grp_name
```

2. Use the recovery method to restore user or application data. In preparation for conversion, the recovery method should have been done with the standard backups you made in preparation for conversion. The following example shows an `frecover` from the `fbackup` example in “step 4. Backing up your LVM configuration and user data.”

```
# mount -F vxfs /dev/vg01/lvol3 /foodir
# frecover -r -f /dev/rmt/c0t0d0BEST
```

Examples

Example: displaying the `vxvmconvert` menu

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

```
# vxvmconvert
```

The following menu is displayed:

```
Volume Manager Support Operations
Menu: Volume Manager/LVM_Conversion

1 Analyze LVM Volume Groups for Conversion
2 Convert LVM Volume Groups to VxVM
3 Roll back from VxVM to LVM
list List disk information
listvg List LVM Volume Group information
? Display help about menu
??Display help about the menuing system
q Exit from menus
```



Example: listing disk information

The `list` option of `vxvmconvert` displays information about the disks on a system. Select the `list` option from the `vxvmconvert` Main Menu:

```
Menu: Volume Manager/LVM_Conversion/list
```

```
# list
```

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

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

DEVICE	DISK	GROUP	STATUS
c0t5d0	-	-	online
c0t8d0	-	-	LVM
c0t9d0	-	-	LVM
c0t10d0	disk01	rootdg	online
c0t11d0	-	-	online

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

Example: listing LVM volume group information

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

Select the `listvg` option from the `vxvmconvert` Main Menu:

```
Menu: Volume Manager/LVM_Conversion/ListLVMVolumeGroups
```

```
# listvg
```

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

Select the Volume Group as follows:

```
Enter Volume Group (i.e.- vg08) or "all" [<address>,all,q,?]
(default: all)
```

```
LVM VOLUME GROUP INFORMATION
NAME  TYPE  PHYSICAL VOLUME
vg00  ROOT   c0t5d0
vg09Non-Rootc0t9d0
vg08Non-Rootc0t8d0
```

```
Volume Group to list in detail
```

```
[<address>,none,q,?] (default: none) none
```

To display detailed information about a volume group, select any of the volume groups from the above list.

```

Volume Group to list in detail \
  [<address>,none,q,?] (default: none) vg08

--- Volume groups ---
VG Name                /dev/vg08
VG Write Access        read/write
VG Status              available
Max LV                 255
Cur LV                2
Open LV               2
Max PV                 16
Cur PV                1
Act PV                1
Max PE per PV         1016
VGDA                  2
PE Size (Mbytes)      4
Total PE              250
Alloc PE              250
Free PE               0
Total PVG             0

--- Logical volumes ---
  LV Name                /dev/vg08/lvol1
  LV Status              available/syncd
  LV Size (Mbytes)      500
Allocated PE          125
  Used PV                1

--- Physical Extent ---
  LV Name                /dev/vg08/lvol2
  LV Status              available/syncd
  LV Size (Mbytes)      500
  Current LE            125
  Allocated PE          125
Used PV                1

--- Physical volumes ---
  PV Name                /dev/dsk/c0t8d0
  PV Status              available
  Total PE              250
  Free PE               0

List another LVM Volume Group? [y,n,q,?] (default: n)
Select an operation to perform:

```

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



Example: analyzing LVM volume groups

To analyze one or more LVM volume groups:

```
# vxvmconvert
```

```
Volume Manager Support Operations
Menu: VolumeManager/LVM_Conversion
1 Analyze LVM Volume Groups for Conversion
2 Convert LVM Volume Groups to VxVM
3 Roll back from VxVM to LVM
listList disk information
listvgList LVM Volume Group information
? Display help about menu
??Display help about the menuing system
q Exit from menus
```

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

```
Analyze one or more LVM Volume Groups
```

```
Menu: Volume Manager/LVM_Conversion/Analyze_LVM_VGs
```

```
Use this operation to analyze one or more LVM volume groups for
possible conversion using the VxVM Volume Manager. This
operation checks for problems that would prevent the conversion
from completing successfully. It calculates the space required
to add the volume groups disks to a Volume Manager disk group,
and to replace any existing partitions and volumes with Volume
Manager volumes, plexes, and sub-disks.
```

More than one volume group or pattern may be entered at the prompt.

Here are some LVM volume group selection examples:

```
all:analyze all LVM Volume Groups (all except Root VG)
```

```
listvg:list all LVM Volume Groups
```

```
list:list all disk devices
```

```
vg_name:a single LVM Volume Group, named vg_name
```

```
<pattern>:for example vg08 vg09 vg05
```

```
Select volume groups to analyze: [<pattern
```

```
list>,all,list,listvg,q,?] vg08
```

```
Name a new disk group [<group>,list,q,?] (default: dg08)
```

Each volume group will be analyzed one at a time. If there are any in this list that you do not want to analyze, you can either abort now or wait until a later time when you will be given an opportunity to skip the analysis of any group(s) in this list. The following disk has been found in the vg08 volume group and will be analyzed for VxVM conversion.

```
c8t8d0
```



To allow analysis, a new VxVM disk group, dg08, will be fabricated and the disk device c4t8d0 will be added to the disk group with the disk name dg0801.

The c4t8d0 disk has been configured for conversion.

The first stage of the Analysis process has completed successfully.

Second Stage Conversion Analysis of vg08

Analysis of vg08 found sufficient Private Space for conversion

Conversion Analysis of c4t8d0 indicates that the Volume Group is still in use, which may prevent the completion of the conversion without having to reboot the system. You may want to double check that none of the volumes in the volume group are in use before continuing with the conversion.

Volume Group Analysis Completed

Hit RETURN to continue.

Example of a failed analysis

```
# vxvmconvert
```

```
Volume Manager Support Operations
```

```
Menu: VolumeManager/LVM_Conversion
```

```
1 Analyze LVM Volume Groups for Conversion
```

```
2 Convert LVM Volume Groups to VxVM
```

```
3 Roll back from VxVM to LVM
```

```
listList disk information
```

```
listvgList LVM Volume Group information
```

```
? Display help about menu
```

```
??Display help about the menuing system
```

```
q Exit from menus
```

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

```
Analyze one or more LVM Volume Groups
```

```
Menu: Volume Manager/LVM_Conversion/Analyze_LVM_VGs
```

Use this operation to analyze one or more LVM volume groups for possible conversion using the VxVM Volume Manager. This operation checks for problems that would prevent the conversion from completing successfully. It calculates the space required to add the volume groups disks to a Volume Manager disk group, and to replace any existing partitions and volumes with Volume Manager volumes, plexes, and sub-disks.

More than one volume group or pattern may be entered at the prompt.

Here are some LVM volume group selection examples:

```
all:analyze all LVM Volume Groups (all except Root VG)
```

```
listvg:list all LVM Volume Groups
```



```
list:list all disk devices
vg_name:a single LVM Volume Group, named vg_name
<pattern>:for example  vg08 vg09 vg05
```

```
Select Volume Groups to analyze :
[<pattern-list>,all,list,listvg,q,?] vg08
```

```
Name a new disk group [<group>,list,q,?] (default: dg08)
```

The following disk has been found in the vg08 volume group and will be analyzed for VxVM conversion.

c4t8d0

To allow analysis, a new VxVM disk group, dg08, will be fabricated and the disk device c4t8d0 will be added to the disk group with the disk name dg0801.

The c4t8d0 disk has been configured for conversion.

The first stage of the Analysis process has completed successfully. Second Stage Conversion Analysis of vg08

Analysis of vg08 found insufficient Private Space for conversion

```
SMALLEST VGRA space= 176
RESERVED space sectors = 78
PRIVATE SPACE/FREE sectors = 98
AVAILABLE sector space = 49
AVAILABLE sector bytes = 50176
RECORDS neededs to convert = 399
MAXIMUM records allowable= 392
```

The smallest disk in the Volume Group (vg08) does not have sufficient private space for the conversion to succeed. There is only enough private space for 392 VM Database records and the conversion of Volume Group (vg08) would require enough space to allow 399 VxVM Database records. This would roughly translate to needing an additional 896 bytes available in the private space. This can be accomplished by reducing the number of volumes in the (vg08) Volume Group, and allowing that for every volume removed, the number of Database records required would be reduced by three. This is only a rough approximation, however.

Hit RETURN to continue.



Example: converting LVM volume groups to VxVM disk groups

To convert LVM volume groups to VxVM disk groups:

```
# vxvmconvert
```

```
Volume Manager Support Operations
Menu: VolumeManager/LVM_Conversion
1 Analyze LVM Volume Groups for Conversion
2 Convert LVM Volume Groups to VxVM
3 Roll back from VxVM to LVM
listList disk information
listvgList LVM Volume Group information
? Display help about menu
??Display help about the menuing system
q Exit from menus
```

```
Select an operation to perform: 2
```

```
Convert one or more LVM Volume Groups
```

```
Menu: VolumeManager/LVM_Conversion/Convert_LVM_VGs
```

```
Use this operation to convert one or more LVM Volume Groups to one
or more VxVM disk groups. This adds the disks to a disk group
and replaces existing partitions with volumes. LVM-VxVM Volume
Group conversion may require a reboot for the changes to take
effect. For this release, only Non-root LVM Volume Groups are
allowed to be converted.
```

```
More than one Volume Group or pattern may be entered at the prompt.
```

```
Here are some LVM Volume Group selection examples:
```

```
all:analyze all LVM Volume Groups (all except Root VG)
```

```
listvg:list all LVM Volume Groups
```

```
list:list all disk devices
```

```
vg_name:a single LVM Volume Group, named vg_name
```

```
<pattern>:for example vg08 vg09 vg05
```

```
Select Volume Groups to convert :
```

```
[<pattern-list>,all,list,listvg,q,?] listvg
```

```
LVM VOLUME GROUP INFORMATION
```

NAME	TYPE	PHYSICAL VOLUME
vg00	ROOT	c4t6d0
vg05	Non-Root	c4t5d0
vg03	Non-Root	c4t3d0 c4t2d0
vg08	Non-Root	c4t8d0

```
Select Volume Groups to convert :
```

```
[<pattern-list>,all,list,listvg,q,?] vg08
```



vg08

Convert this Volume Group? [y,n,q,?] (default: **y**)

Name a new disk group [<group>,list,q,?] (default: **dg08**)

The following disk has been found in the vg08 volume group and will be configured for conversion to a VxVM disk group.

c4t8d0

A new disk group dg08 will be created and the disk device c4t8d0 will be converted and added to the disk group with the disk name dg0801.

The c4t8d0 disk has been configured for conversion.

The first stage of the conversion operation has completed successfully. If you commit to the changes hereafter, the system will attempt to umount all of the associated file systems, stop and export each Volume Group, and then attempt to complete the conversion without having to reboot the system. If we are unable to stop and export any of the Volume Groups, then the conversion process will not be able to complete without a reboot. You would then be given the choice to either abort the conversion, or finish the conversion by rebooting the system.

The conversion process will update the /etc/fstab file so that volume devices are used to mount the file systems on this disk device. You will need to update any other references such as backup scripts, databases, or manually created swap devices. If you do not like the default names chosen for the corresponding logical volumes, you may change these to whatever you like using vxedit.

Second Stage Conversion Analysis of vg08

Analysis of vg08 found sufficient Private Space for conversion

Conversion Analysis of c4t8d0 indicates that the Volume Group is still in use, which may prevent the completion of the conversion without having to robot the system. You may want to double check that none of the volumes in the volume group are in use before continuing with the conversion.

Hit RETURN to continue.

Are you ready to commit to these changes? [y,n,q,?] (default: **y**)

Saving LVM configuration records for Volume Group vg08

Volume Group configuration for /dev/vg08 has been saved in

/etc/vx/reconfig.d/vgrecords/vg08/vg08.backup

Beginning the export process on Volume Group "/dev/vg08".

Volume group "/dev/vg08" is still active.

/dev/dsk/c4t8d0




```

/dev/vg08/lv1 will convert to /dev/vx/dsk/dg08/dg08lv1
/dev/vg08/rlv1 will convert to /dev/vx/rdisk/dg08/dg08lv1
/dev/vg08/lv2 will convert to /dev/vx/dsk/dg08/dg08lv2
/dev/vg08/rlv2 will convert to /dev/vx/rdisk/vg08dg/dg08lv2
LVM Volume Group vg08 Records Saved
Unmounting vg08 file systems
Volume group "/dev/vg08" has been successfully changed.

```

```

The Volume Manager is now reconfiguring (partition phase)...
Volume Manager: Initializing c4t8d0 as a converted LVM disk.
The system reconfiguration will now be done without rebooting.
The Volume Manager is now reconfiguring (initialization phase)...
Volume Manager: Adding dg0801 (c4t8d0) as a converted LVM disk.
Adding volumes for c4t8d0...
Starting new volumes...
Updating /etc/fstab...
The system will now Convert the LVM Volume Groups over to VxVM disk
groups.
Convert other LVM Volume Groups? [y,n,q,?] (default: n)

```

Example of a failed conversion

```
# vxvmconvert
```

```
Volume Manager Support Operations
Menu: VolumeManager/LVM_Conversion
```

```

1 Analyze LVM Volume Groups for Conversion
2 Convert LVM Volume Groups to VxVM
3 Roll back from VxVM to LVM
listList disk information
listvgList LVM Volume Group information
? Display help about menu
??Display help about the menuing system
q Exit from menus

```

```

Select an operation to perform: 2
Convert one or more LVM Volume Groups
Menu: VolumeManager/LVM_Conversion/Convert_LVM_VGs
Use this operation to convert one or more LVM Volume Groups to one
or more VxVM disk groups. This adds the disks to a disk group
and replaces existing partitions with volumes. LVM-VxVM Volume
Group conversion may require a reboot for the changes to take

```



effect. For this release, only Non-root LVM Volume Groups are allowed to be converted.

More than one Volume Group or pattern may be entered at the prompt.

Here are some LVM Volume Group selection examples:

all:analyze all LVM Volume Groups (all except Root VG)

listvg:list all LVM Volume Groups

list:list all disk devices

vg_name:a single LVM Volume Group, named vg_name

<pattern>:for example vg08 vg09 vg05

Select Volume Groups to convert :

[<pattern-list>,all,list,listvg,q,?] **listvg**

LVM VOLUME GROUP INFORMATION

NAME	TYPE	PHYSICAL VOLUME
vg00	ROOT	c4t6d0
vg05	Non-Root	c4t5d0
vg03	Non-Root	c4t3d0 c4t2d0
vg08	Non-Root	c4t8d0

Select Volume Groups to convert :

[<pattern-list>,all,list,listvg,q,?] **vg08**

vg08

Convert this Volume Group? [y,n,q,?] (default: y)

Name a new disk group [<group>,list,q,?] (default: **dg08**)

The following disk has been found in the vg08 volume group and will be configured for conversion to a VxVM disk group.

c4t8d0

A new disk group dg08 will be created and the disk device c4t8d0 will be converted and added to the disk group with the disk name dg0801.

The c4t8d0 disk has been configured for conversion.

The first stage of the conversion operation has completed

successfully. If you commit to the changes hereafter, the system will attempt to unmount all of the associated file systems, stop and export each Volume Group, and then attempt to complete the conversion without having to reboot the system. If we are unable to stop and export any of the Volume Groups, then the conversion process will not be able to complete without a reboot. You would then be given the choice to either abort the conversion, or finish the conversion by rebooting the system.

The conversion process will update the /etc/fstab file so that volume devices are used to mount the file systems on this disk device. You will need to update any other references such as backup scripts, databases, or manually created swap devices. If

you do not like the default names chosen for the corresponding logical volumes, you may change these to whatever you like using `vxedit`.

Second Stage Conversion Analysis of vg08

```
Analysis of vg08 found insufficient Private Space for conversion
SMALLEST VGRA space           = 176
RESERVED space sectors = 78
PRIVATE SPACE/FREE sectors = 98
AVAILABLE sector space = 49
AVAILABLE sector bytes = 50176
RECORDS neededs to convert= 399
MAXIMUM records allowable    = 392
```

The smallest disk in the Volume Group (vg08) does not have sufficient private space for the conversion to succeed. There is only enough private space for 392 VM Database records and the conversion of Volume Group (vg08) would require enough space to allow 399 VxVM Database records. This would roughly translate to needing an additional 896 bytes available in the private space. This can be accomplished by reducing the number of volumes in the (vg08) Volume Group, and allowing that for every volume removed, the number of Database records required would be reduced by three. This is only a rough approximation, however.

Hit RETURN to continue.

Example: list, listvg, and vxprint outputs of an LVM volume group before and after conversion

The examples given below, shows the `vxvmconvert listvg`, `list`, and `vxprint` output for an LVM volume group `vg08` converted to a VxVM disk group `dg08`.

Example of vxvmconvert listvg output before conversion of volume group vg08

```
LVM VOLUME GROUP INFORMATION
NAME          TYPE          PHYSICAL VOLUME
vg00          ROOT          c0t5d0
vg08          Non-Root     c0t8d0
vg09          Non-Root     c0t9d0
```



Example of the vxvmconvert list output which shows the disk devices on a system

```

DEVICE          DISK          GROUP          STATUS
c0t5d0          -             -              online
c0t8d0          -             -              LVM
c0t9d0          -             -              LVM
c0t10d0         disk01        rootdg         online
c0t11d0         -             -              online
Disk group: rootdg

```

What does vxmvconvert list display?

The *device* indicates a physical disk, a *disk* with a name indicates if the disk is under VxVM control, a *group* shows the disk group name, and the *status* indicates if it is an LVM disk. If the status is *online*, that means VxVM acknowledges the disk but doesn't have it under its control.

Example of the vxprint output before conversion

```

TY NAME      ASSOC      KSTATE  LENGTH  PLOFFS  STATE  TUTILO  PUTILO
dg rootdg    rootdg    -        -        -        -        -        -
dm disk01    c0t10d0  -        2079468 -        -        -        -

```

The `list` and `listvg` output is from within the `vxvmconvert` command. `vxprint` is a command line command.

Example of the listvg output after conversion of vg08

```

LVM VOLUME GROUP INFORMATION
Name      Type      Physical Volumes
vg00      ROOT      c0t5d0
vg09      Non-Root  c0t9d0

```

Volume Group to list in detail: None

Note Note that `vg08` is no longer listed under LVM information.



Example of the `vxvmconvert listvg` output after conversion of volume group `vg08` to `dg08`

```

DEVICE      DISK      GROUP      STATUS
c0t5d0      -         -          online invalid
c0t8d0      dg0801   dg08       online
c0t9d0      -         LVM
c0t10d0     disk01   rootdg     online
c0t11d0     -         -          online

```

```
Disk group: rootdg
```

Example of the `vxprint` output after conversion

```

TY NAME      ASSOC      KSTATE      LENGTH      PLOFFS      STATE      TUTILO      PUTILO
dg dg08      dg08      -           -           -          -          -
dm dg0801     c0t8d0    -           2080768    -          -          -
v  dg081v1    fsgen     ENABLED     102400     -          ACTIVE -    -
pl dg081v1-01dg081v1  ENABLED     102400     -          ACTIVE -    -
sd dg0801-01 dg081v1-01ENABLED102400 -          -          -

```

`vxprint` Explained

The `vxprint` list given above provides the following information:

- ◆ The *disk group* `dg08` contains the *VxVM disk* `dg0801` and the *volume* `dg081v1`. The *VxVM disk* `dg0801` is associated with *disk device* `c0t8d0` and is 2080768 blocks in length. The *volume* `dg081v1` is of type `fsgen`, is *enabled* in the *VxVM kernel driver*, is of *length* 102400, 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 `fsgen`.
- ◆ The *plex* `dg081v1-01` is associated with *volume* `dg081v1`, and maps the entire address range of the volume. Associated with the plex is one *subdisk*, `dg0801-01` which maps the plex address range from 0 to the entire length of the plex, i.e. 102400 blocks. As implied by the root of its name, the subdisk `dg0801-01` uses an extent from the *VxVM disk* `dg0801`.



Example: VxVM to LVM rollback

Volume Manager Support Operations
Menu: VolumeManager/LVM_Conversion

```

1 Analyze LVM Volume Groups for Conversion
2 Convert LVM Volume Groups to VxVM
3 Roll back from VxVM to LVM
listList disk information
listvgList LVM Volume Group information
? Display help about menu
?? Display help about the menuing system
q Exit from menus

```

```

Select an operation to perform: 3
Rollback one or more LVM Volume Groups
Menu: VolumeManager/LVM_Conversion/Rollback_LVM_VG

```

Use this operation to rollback from a conversion of an LVM Volume Group. This operation will tear down the VxVM disk group and recreate the LVM volume group in its original form. User data is untouched by rollback.

Caution If the VxVM configuration has changed since the conversion you should NOT use this operation. See the section “Restoring the LVM Volume Group Configuration” for more information.

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

```

all:Rollback all converted LVM Volume Groups
listvg:list all LVM Volume Groups eligible for rollback
list:list all disk devices
vg_name:a single LVM Volume Group, named vg_name
<pattern>:for example  vg08 vg09 vg05

```

```

Select Volume Group(s) to rollback :
[<pattern-list>,all,list,listvg,q,?]
vg08
Roll back this Volume Group? [y,n,q,?] (default: y)
Rolling back LVM configuration records for Volume Group vg08
Selected Volume Groups have been restored.
Hit RETURN to continue.
Rollback other LVM Volume Groups? [y,n,q,?] (default: n)

```

General Information Regarding Conversion Speed

The speed of the process of converting an existing LVM volume group to a similar VxVM disk group is largely dependent upon the size of the volume group being converted, as well as on the complexity of the volumes within that volume group.

Factors affecting conversion speed include:

- ◆ Size of volume groups. The larger the volume groups, the larger the LVM metadata on each disk. A copy must be made of the LVM metadata for each physical disk. Some areas are greater than 2MB; therefore, a 50-disk volume group requires 50 2MB reads and writes (i.e., 100 large I/Os) to complete.
- ◆ 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 of the first 5 1/2 disks can be created. 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. However, for the striped volume, 50 disks need to be checked. Also, the complexity of reproducing the VxVM commands to set up the striped volumes requires more VxVM commands to be generated to represent more smaller sub-disks 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, but LVM currently does not allow this.

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 (150 Gb) 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.





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](#)
- ◆ [Existing Features in LVM not supported in VxVM](#)

For more information on LVM commands, refer to *HP-UX Managing Systems and Workgroups*, and LVM manual pages in *HP-UX Reference Volumes 2,3, and 5*. For information on VxVM commands, refer to the VERITAS Volume Manager documentation (see the [Preface](#)).



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 (see the [Preface](#)).

Table 2. Command Comparison

LVM	Description/Action	VxVM	Description/Action
lvchange	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.
lvlnboot	Creates root, primary and secondary swap and dump volumes. It also creates boot areas on the disk.		There is no equivalent command for this release.
lvcreate	Creates a logical volume.	vxassist	Creates volumes with the make parameter. Example: vxassist make <i>vol_name</i> 100M layout=stripe
lvextend	Increases disk space allocated to a logical volume.	vxassist	Increases a volume in size with the growto or growby parameter. Example: vxassist growto <i>vol_name</i> 200M, vxassist growby <i>vol_name</i> 100M vxassist creates and modifies volumes.

Table 2. Command Comparison

LVM	Description/Action	VxVM	Description/Action
lvreduce	Decreases disk space allocated to a logical volume.	vxassist	Decreases a volume in size with the <code>shrinkto</code> or <code>shrinkby</code> parameters. Example: <code>vxassist shrinkto vol_name 200M</code> Make sure you shrink the file system before shrinking the volume.
lvremove	Removes one or more logical volumes from a volume group.	vxedit vxassist	Removes volumes with the <code>-rf rm</code> parameters. Example: <code>vxedit -rf rm vol_name</code> Removes a volume with the <code>remove volume</code> parameters. Example: <code>vxassist remove volume vol_name</code>
lvsplit	Splits a mirrored logical volume into two logical volumes.	vxassist snapshot	The snapshot operation takes one of the attached temporary mirrors and creates a new volume with the temporary mirror as its one plex. Example: <code>vxassist snapshot vol_name new_volume</code>
lvmerge	Reverses and converts the <code>lvsplit</code> logical volumes to a single logical volume.	vxassist snapback	The snapback operation returns the snapshot plex to the original volume from which it was snapped. Example: <code>vxassist snapback new_volume</code>
lvsync	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 2. Command Comparison

LVM	Description/Action	VxVM	Description/Action
pvcreate	Makes a disk an LVM disk.	vxdisksetup	Brings a disk under VxVM control. Example: <code>vxdisksetup c0t3d0</code> Option 1 in the <code>vxdiskadm</code> menu adds or initializes one or more disks.
pvdisplay	Displays information about physical volumes in a volume group.	vxdisk list	Lists information about VxVM disks. Example: <code>vxdisk list disk_name</code>
pvchange	Sets physical volume characteristics to allow/deny allocation of additional physical extents from this disk.	vxdisk vxdisk set vxedit	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.
pvmove	Moves allocated physical extents from source to destination within a volume group.	vxevac vxsd mv vxdiskadm	Moves volumes off a disk. Performs volume operations on a subdisk. Moves the contents of old subdisk onto the new subdisks and replaces old sub disk with the new subdisks for any associations. The <code>vxdiskadm</code> script presents a menu of possible operations to the user. Option 7 in the <code>vxdiskadm</code> menu moves volumes.
pvremove	Removes the LVM header information and releases the disk from LVM control.	vxdiskunsetup	Removes the VxVM header information and releases the disk from VxVM control.

Table 2. Command Comparison

LVM	Description/Action	VxVM	Description/Action
vgcreate	Creates a volume group.	vxdiskadd vxdbg init	Creates a new disk group and/or adds disks to a disk group.
vgdisplay	Displays information on all volume groups.	vxdbg list vxprint	Displays the contents of a disk group. Displays information about all objects or a subset of objects.
vgchange	Activates or deactivates one or more volume groups.	vxdbg -g <i>diskgroup</i> set activation= <i>mode</i>	Activates a shared disk group.
vgextend	Extends a volume group by adding one or more disks to it.	vxdiskadd vxdiskadm	Adds a disk to the disk group. Option 1 in the vxdiskadm menu adds disks to the disk group.
vgreduce	Reduces a volume group by removing one or more disks from it.	vxdbg rmdisk vxdisk rm vxdiskadm	Removes disks from a disk group. Removes the specified disk access record by disk access name. Option 3 in the vxdiskadm menu removes disks.
vgscan	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. Option list in the vxdiskadm menu displays disk information.
vgsync	Synchronizes mirrors that are stale in one or more logical volumes.	vxrecover	Starts resynchronization and recovery of volumes.



Table 2. Command Comparison

LVM	Description/Action	VxVM	Description/Action
vgremove	Removes the definition of a volume group from the system.	vxldg deport vxdiskadm	Deports a disk group from the system. Option 9 in the vxdiskadm menu removes a disk group.
vgexport	Removes a volume group from the system.	vxldg deport vxdiskadm	Deports a disk group from the system. Option 9 in the vxdiskadm menu removes a disk group.
vgimport	Adds a volume group to the system by scanning physical volumes which have been exported using vgexport.	vxldg import vxdiskadm	Imports a disk group. Option 8 in the vxdiskadm menu imports a disk group.
	No LVM command	vxplex	Operates on plex objects.
lvchange, lvextend, lvcreate, lvreduce	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 the VERITAS Volume Manager. You can perform the LVM tasks by using SAM or the command line interface. Similarly, you can choose to perform VxVM tasks by using the VERITAS Volume Manager Storage Administrator or from 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.*

All the VxVM tasks listed in the task comparison chart can be performed by the VERITAS Volume Manager Storage Administrator. For more information, refer to the *VERITAS Volume Manager Storage Administrator Administrator's Guide*.

For more information on LVM commands, refer to *HP-UX Managing Systems and Workgroups*, and LVM manual pages in the *Reference Volumes 2, 3, and 5*. For information on VxVM commands, refer to the VERITAS Volume Manager documentation package (see the [Preface](#)).

Note Mirroring of a VxVM root disk is supported in this release.

Mirroring in LVM is supported only if you have MirrorDisk/UX already installed as an add-on product. In addition, mirroring in VxVM requires an additional license.

Table 3. LVM and VxVM Task Comparison

Task Type	Description	Example
LVM	Create an LVM disk.	<code>pvcreate /dev/rdisk/<i>disk_name</i></code>
VxVM	Bring a disk under Volume Manager control.	<code>vxdiskadd <i>device_name</i></code> Option 1 in the <code>vxdiskadm</code> menu adds a disk and initializes it.
LVM	Create a volume group	<code>vgcreate /dev/<i>vol_grp</i> /dev/dsk/<i>disk_name</i></code>

Example for a *disk_group* = `veritasdg`, *medianame* = `disk01`, *vol_name* = `veritasvol`, *plex name* = `veritasvol-01`, *subdisk* = `disk01-01`, *devicename* = `c0t0d0`.



Table 3. LVM and VxVM Task Comparison

Task Type	Description	Example
VxVM	Create a disk group.	<pre>vxdg init <i>disk_group</i> <i>disk_name</i></pre> <p>Option 1 in the <code>vxdiskadm</code> menu performs this task.</p>
LVM	Add a new disk to the existing volume group.	<pre>vgextend /dev/<i>vol_grp</i> /dev/rdisk/<i>disk_name</i></pre>
VxVM	Add a disk to an existing disk group.	<pre>vxdg -g <i>disk_group</i> adddisk disk=<i>devicename</i></pre>

Example for a *disk_group* = veritasdg, *medianame* = disk01, *vol_name* == veritasvol, *plex name* = veritasvol-01, *subdisk* == disk01-01, *devicename* = c0t0d0.



Table 3. LVM and VxVM Task Comparison

Task Type	Description	Example
LVM	Extend a logical volume or increase space allocated to a logical volume.	<code>lvextend -l 50 /dev/vol_grp/lvol_name</code> 1—indicates the number of logical extents in the logical volume
VxVM	Increase the volume by or to a given length.	<code>vxresize -g disk_group -F vxfs vol_name length</code> <code>vxassist growto vol_name new_length</code> <code>vxassist growby vol_name length_change</code> Grow the file system after growing the volumes.
LVM	Reduce a logical volume.	<code>lvreduce -L to_size /dev/vol_grp/lvol_name</code> -L indicates the number of megabytes.
VxVM	Reduce a volume by or to a given length.	<code>vxresize -g disk_group -F vxfs vol_name to_length</code> <code>vxassist -b shrinkby vol_name length</code> <code>vxassist -b shrinkto vol_name newlength</code> Shrink the file system before reducing the volume.
LVM	Import and activate a volume group.	<code>vgimport -v /dev/vol_grp \ /dev/dsk/disk_name</code> <code>vgchange -a y /dev/vol_grp</code>
VxVM	Import a disk group to make the specified disk group accessible on the local machine.	<code>vxdg -tfc -n newname import disk_group</code> Option 8 in the <code>vxdiskadm</code> menu performs this task.

Example for a `disk_group` = `veritasdg`, `medianame` = `disk01`, `vol_name` = `veritasvol`, `plex name` = `veritasvol-01`, `subdisk` = `disk01-01`, `devicename` = `c0t0d0`.



Table 3. LVM and VxVM Task Comparison

Task Type	Description	Example
LVM	Export and deactivate an LVM volume group, and its associated logical volumes.	<code>vgchange -a n <i>vol_group</i></code> <code>vgexport /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>vxdg deport <i>disk_group</i></code> Option 9 in the <code>vxdiskadm</code> menu performs this task.
LVM	Back up volume group configuration information.	<code>vgcfgbackup -f /pathname/filename <i>vol_grp</i></code>
VxVM	Back up volume group configuration information.	<code>dgcfgbackup -f /pathname/filename <i>vol_grp</i></code>
LVM	Restore volume group configuration to a particular physical volume.	<code>vgrestore -n /dev/<i>vol_grp</i> /dev/rdisk/<i>disk_name</i></code>
VxVM	Restore volume group configuration to a particular physical volume.	<code>dgcfgrestore -n /dev/<i>vol_grp</i> /dev/rdisk/<i>disk_name</i></code>
LVM	Increase or decrease secondary swap space. Enlarge an existing swap logical volume, or add a new swap logical volume.	<code>lvextend</code> —to increase swap space <code>lvreduce</code> —to decrease swap space
VxVM	Add a new swap volume (HP-UX 11i Version 1.5 only).	<code>vxassist make <i>swapvol2 size</i> (HP-UX 11i Version 1.5 only)</code>
LVM	Remove a volume group. This destroys a volume group by removing its last disk and removing it from <code>/etc/lvmtab</code> .	<code>vgremove /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>vxdg deport <i>disk_group</i></code> <code>vxdg init <i>disk_group</i></code>

Example for a `disk_group = veritasdg`, `medianame = disk01`, `vol_name = veritasvol`, `plex name = veritasvol-01`, `subdisk = disk01-01`, `devicename = c0t0d0`.



Table 3. LVM and VxVM Task Comparison

Task Type	Description	Example
LVM	Extend a volume group by adding LVM disks to the volume group.	<code>vgextend /dev/vol_grp/\ /dev/dsk/disk_name</code>
VxVM	Add one or more disks to the disk group.	<code>vxdiskadd disk_name</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>vgreduce /dev/vol_grp /dev/dsk/disk_name</code>
VxVM	Remove a disk from disk group.	<code>vxdg -g disk_group -k rmdisk disk_name</code>
LVM	Mirroring a disk involves several steps.	<ol style="list-style-type: none"> <code>pvcreate /dev/rdisk/second_disk</code> <code>vgextend /dev/vol_grp \ /dev/dsk/second_disk</code> <code>lvextend -m no_of_mirrors \ /dev/vol_grp/lvol_name \ /dev/dsk/second_disk</code>
VxVM	Mirroring a disk To mirror volumes on a disk or control default mirroring and causes a disk to have its contents mirrored to available space on another disk.	<code>vxmirror -g disk_group -d yes no \ disk_name [new_disk_name]</code> <code>vxmirror -d yes disk_name</code> Option 6 in the <code>vxdiskadm</code> menu performs this task.
LVM	Mirroring an LVM root disk involves several steps.	<ol style="list-style-type: none"> <code>pvcreate -B /dev/rdisk/second_disk</code> <code>mkboot -l /dev/dsk/second_disk</code> <code>vgextend /dev/vol_grp \ /dev/dsk/second_disk</code> <code>lvextend -m no_of_mirrors \ /dev/vol_grp/root_lvol \ /dev/dsk/second_disk</code> <code>lvlnboot -r /dev/vol_grp/lvol_name</code>

Example for a `disk_group` = `veritasdg`, `medianame` = `disk01`, `vol_name` = `veritasvol`, `plex name` = `veritasvol-01`, `subdisk` = `disk01-01`, `devicename` = `c0t0d0`.



Table 3. LVM and VxVM Task Comparison

Task Type	Description	Example
VxVM	Mirroring the VxVM root disk.	<code>vxrootmir [-v] [-t <i>tasktag</i>] disk_access_name disk_media_name</code>
LVM	Create a logical volume in LVM volume group.	<code>lvcreate -L <i>vol_size</i> /dev/<i>vol_grp</i></code>
VxVM	Create a volume of one of these layout types: A concatenated volume A striped mirror volume A RAID-5 volume	<code>vxassist make <i>vol_name</i> <i>length</i></code> <code>vxassist make <i>vol_name</i> <i>length</i> layout=mirror, stripe</code> <code>vxassist make <i>vol_name</i> <i>length</i> layout=raid5</code>
LVM	Display information about logical volumes.	<code>lvdisplay /dev/<i>vol_grp</i>/<i>lvol_name</i></code>
VxVM	Display all volume information. Display information about a specific volume.	<code>vxprint -vt</code> <code>vxprint -ht <i>vol_name</i></code>

Example for a *disk_group* = veritasdg, *medianame* = disk01, *vol_name* == veritasvol, *plex name* = veritasvol-01, *subdisk* == disk01-01, *devicename* = c0t0d0.



Table 3. LVM and VxVM Task Comparison

Task Type	Description	Example
LVM	Display information about volume groups.	<code>vgdisplay -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>pvdisplay /dev/dsk/disk_name</code>
VxVM	Display information about Volume Manager volumes.	<code>vxinfo</code> or <code>vxprint</code>
LVM	Remove a logical volume.	<code>lvremove /dev/vol_grp/lvol_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>vgreduce /dev/vol_grp /dev/dsk/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>vgremove /dev/vol_grp</code>

Example for a *disk_group* = *veritasdg*, *medianame* = *disk01*, *vol_name* = *veritasvol*, *plex name* = *veritasvol-01*, *subdisk* = *disk01-01*, *devicename* = *c0t0d0*.



Table 3. 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>vxldg 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>vgcreate /dev/<i>vol_grp</i> \ /dev/dsk/<i>disk_name</i> \ /dev/dsk/<i>disk_name_2</i></code> To remove the link: <code>vgreduce /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>vxldmp (7)</code> and the <i>VERITAS Volume Manager Administrator's Guide</i> for more information on DMP.
LVM	Create a mirrored logical volume.	<code>lvcreate -l <i>num_log_extents</i> -m 1 -n <i>mirr_lv</i> /dev/<i>vol_grp</i></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>lvreduce -m 0 /dev/<i>vol_grp</i>/<i>mirr_lv</i></code> <code>lvremove /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 = c0t0d0`.

Table 3. 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>lvextend -m 2 /dev/<i>vol_grp</i>/<i>lvol_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>lvsplit -s backup /dev/<i>vol_grp</i>/<i>lvol_name</i></code>
VxVM	Snapshot a volume and create a new volume.	<code>vxassist snapshot <i>vol_name</i> <i>new_vol_name</i></code>
LVM	Combine two logical volumes back into a mirrored logical volume	<code>lvmerge /dev/<i>vol_grp</i>/<i>split_vol_name</i> \ /dev/<i>vol_grp</i>/<i>lvol_name</i> <i>split_vol_name</i>= active logical volume</code>
VxVM	Returns the snapshot plex to the original volume from which it was snapped.	<code>vxassist snapback <i>new_vol_name</i></code>
LVM	Move a mirrored logical volume from one disk to another.	<code>pvmove -n /dev/<i>vol_grp</i>/<i>lvol_name</i> \ /dev/dsk/<i>disk_name</i> /dev/dsk/<i>disk_name2</i></code>
VxVM	Move a plex.	<code>vxplex mv <i>orig_plex</i> <i>new_plex</i></code>

Example for a *disk_group* = veritasdg, *medianame* = disk01, *vol_name* = veritasvol, *plex name* = veritasvol-01, *subdisk* = disk01-01, *devicename* = c0t0d0.



Table 3. LVM and VxVM Task Comparison

Task Type	Description	Example
LVM	Synchronize a mirrored logical volume. Synchronize extents within a mirrored logical volume.	<code>lvsync /dev/<i>vol_grp</i>/<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>vgsync /dev/<i>vol_grp</i></code>
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 <i>vol_name</i></code>
LVM	Start a volume.	<code>lvchange -a y /dev/<i>vol_grp</i>/<i>lvol_name</i></code>
VxVM	Start a volume.	<code>vxrecover -s <i>vol_name</i> vxvol start <i>vol_name</i></code>
LVM	Stop a volume.	<code>lvchange -a n /dev/<i>vol_grp</i>/<i>lvol_name</i></code>
VxVM	Stop a volume.	<code>vxvol stop <i>vol_name</i></code>
LVM	Make a disk available as a hot spare.	<code>pvchange -z y /dev/<i>disk</i>/<i>disk_name</i></code>
VxVM	Make a disk available as a hot spare.	<code>vxedit set spare=on <i>disk_name</i></code>

Example for a *disk_group* = veritasdg, *medianame* = disk01, *vol_name* = veritasvol, *plex name* = veritasvol-01, *subdisk* = disk01-01, *devicename* = c0t0d0.



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) GUI, or the command line interface. For more information, refer to the *VERITAS Enterprise Administrator (VEA 500 Series) Getting Started* guide and the *VERITAS Volume Manager Administrator's Guide*.

Table 4. Additional VxVM Tasks with no LVM equivalents

Task Description	Example
Hot-relocation: in addition to using disks as hot spares, the hot relocation facility can also use any available free space in the disk group. If no disks have been designated as spares when a failure of a redundant object occurs, VxVM automatically uses any available free space in the disk group in which the failure occurs. If there is not enough spare disk space, a combination of spare space and free space is used. After a disk is repaired, you can move all the hot-relocated subdisks back to the original disk using the <code>vxunrelloc (1M)</code> utility.	No action needed for hot relocation. To move hot-relocated subdisks back to the original disk: <code>vxunrelloc disk_name</code>
Rename a disk	<code>vxedit rename old_diskname new_diskname</code>
Offline a disk.	<code>vxdisk offline disk_name</code> Alternatively, menu option 12 of <code>vxdiskadm</code> performs 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>



Table 4. Additional VxVM Tasks with no LVM equivalents

Task Description	Example
Rename a disk group.	<code>vxchg -tC -n <i>newdg_name</i></code>
Rename a volume.	<code>vxedit -v rename <i>name newname</i></code> Update the <code>/usr/fstab</code> file with the new name.
Add a DRL log to a volume.	<code>vxassist addlog <i>vol_name</i></code>
Create a snapshot copy of a volume.	<code>vxassist snapshot <i>vol_name</i></code> <code><i>temp_vol_name</i></code>
Recover a volume.	<code>vxrecover -g <i>disk_group volume</i></code> <code><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>

Existing Features in LVM not supported in VxVM

Some of the existing 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 5. 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.
Powerfail timeout feature: Automatically re-enable a disk or a path to a disk, after temporary error condition (resulting in EPOWERF error on I/Os) disappears on that disk or path.	Powerfail timeout feature: After the EPOWERF error condition disappears, the reconfiguration command must be run manually to re-enable the paths and the disks which were disabled due to EPOWERF error. See the <code>pfto</code> feature in the <code>vxdtl(1M)</code> manual page for more information.
Logical Volume Timeout (LVTO). If LVTO on a logical volume is set to zero, which is the default, an I/O is retried forever.	VxVM does not support the LVTO feature. However, VxVM supports the powerfail timeout feature to handle transient error conditions. VxVM tries an I/O only on active paths to a disk; hence, it never retries indefinitely. See the powerfail timeout feature and also refer to the <code>pfto</code> feature in the <code>vxpfto</code> manual pages for more details.
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.





This chapter describes the VERITAS Enterprise Administrator (VEA) graphical user interface (GUI), and its relationship with the LVM GUI, and the System Administration Manager (SAM).

The following topics are discussed in this chapter:

- ◆ [Listing Disk Devices in SAM](#)
- ◆ [Listing Volume Groups and Disk Groups in SAM](#)
- ◆ [Listing Logical Volumes in SAM](#)

SAM and the VEA coexist as independent entities. The VEA recognizes and labels LVM volumes and disks, but does not manage them. Similarly, SAM recognizes and labels VxVM disks, but does not manage them. To manage VxVM disks graphically, you must use the VEA.

Note If you run the VEA client from the command line, you must explicitly refresh SAM (with Options->Refresh) to see VxVM changes reflected in SAM screens.

Only privileged users can run the VEA client.

At present, it is not possible to launch the VEA client from SAM. This will be corrected in a future SAM patch.

For information about the VEA, see the *VERITAS Enterprise Administrator (VEA 500 Series) Getting Started* guide and the online help that is available from within the VEA.



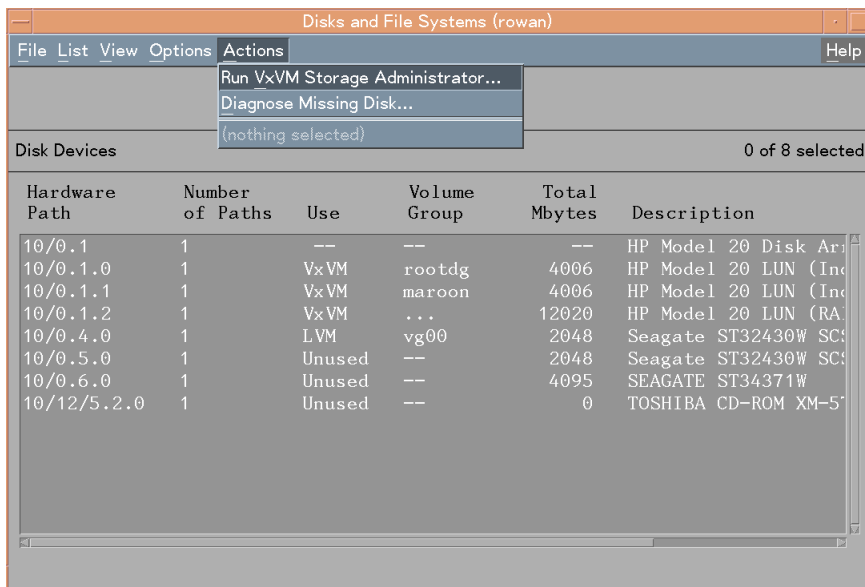
Listing Disk Devices in SAM

To list disk devices in SAM, from the Disks and File Systems SAM area, select Disk Devices. The Disk Devices screen lists the system's disk devices. When VxVM is installed on the system, SAM includes a "Use" 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 "Volume Group" column. If a VxVM disk is initialized, but not yet part of a disk group, it is listed under the "Volume Group" column.

The following figure shows a Disk Devices screen for a system with three VxVM disks, one LVM disk, and a CD device. One of the VxVM disks is part of the rootdg disk group. The second disk is part of the maroon disk group. the third disk, which is listed under the "Volume Group" column, is not yet part of a disk group.

Displaying Disk Devices Using SAM



Hardware Path	Number of Paths	Use	Volume Group	Total Mbytes	Description
10/0.1	1	--	--	--	HP Model 20 Disk Array
10/0.1.0	1	VxVM	rootdg	4006	HP Model 20 LUN (Incl)
10/0.1.1	1	VxVM	maroon	4006	HP Model 20 LUN (Incl)
10/0.1.2	1	VxVM	...	12020	HP Model 20 LUN (RAID)
10/0.4.0	1	LVM	vg00	2048	Seagate ST32430W SCSI
10/0.5.0	1	Unused	--	2048	Seagate ST32430W SCSI
10/0.6.0	1	Unused	--	4095	SEAGATE ST34371W SCSI
10/12/5.2.0	1	Unused	--	0	TOSHIBA CD-ROM XM-50

Listing Volume Groups and Disk Groups in SAM

To list volume groups and disk groups in SAM, from the Disks and File Systems SAM area, select Volume Groups. The Volume Groups screen lists the LVM volume groups and the VxVM disk groups on the system. The following figure shows a Volume Groups screen for the same example system. vg00 is an LVM volume group with eight logical volumes; rootdg and maroon are VxVM disk groups, with one volume each. See the figure “[Listing LVM Volume Groups and VxVM Disk Groups with SAM](#)” on page 63.

Listing LVM Volume Groups and VxVM Disk Groups with SAM

Name	Type	Mbytes Available	Physical Volumes	Logical Volumes
vg00	LVM	0 of 2044	1	8
rootdg	VxVM	3606 of 4006	1	1
maroon	VxVM	3506 of 4006	1	1

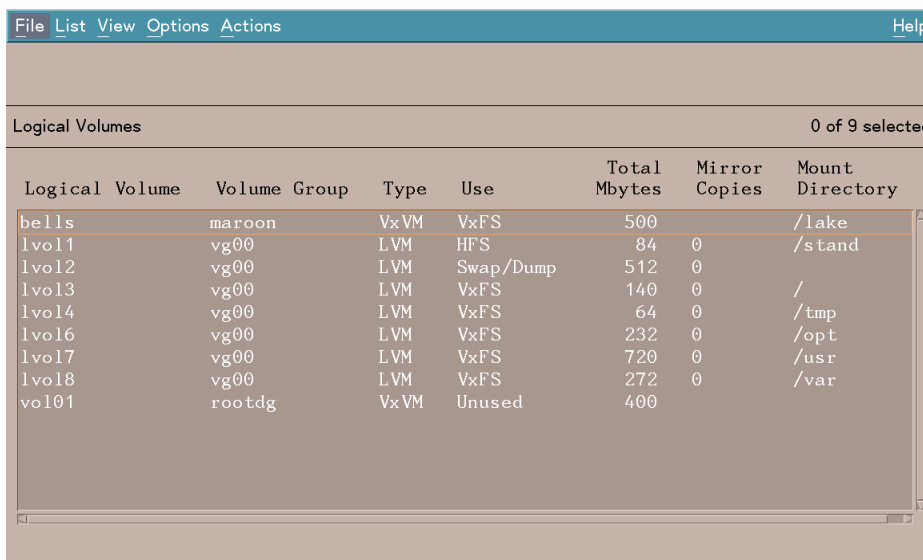


Listing Logical Volumes in SAM

To list logical volumes in SAM, from the Disks and File Systems SAM area, select Logical Volumes. The Logical Volumes screen lists the LVM logical volumes and the VxVM volumes on the system. The “Type” column indicates whether a volume is controlled by LVM or VxVM. The “Use” column shows whether a volume is in use and if so, what it is used for.

The following figure shows a Logical Volumes screen for the same example system. The eight LVM logical volumes in vg00 are used for HFS and VxFS file systems and for swap and dump. The bells VxVM volume in maroon is used for VxFS and has a directory mounted on it. The vol01 VxVM volume in rootdg is not currently in use. See the figure “[Listing LVM Logical Volumes and VxVM Volumes with SAM](#)” on page 64.

Listing LVM Logical Volumes and VxVM Volumes with SAM



The screenshot shows a window titled "Logical Volumes" with a menu bar (File, List, View, Options, Actions) and a "Help" button. Below the menu bar, it says "0 of 9 selected". The main area contains a table with the following columns: Logical Volume, Volume Group, Type, Use, Total Mbytes, Mirror Copies, and Mount Directory. The table lists 9 logical volumes: bells (VxVM, VxFS, 500 Mbytes, /lake), lvol1 (LVM, HFS, 84 Mbytes, /stand), lvol2 (LVM, Swap/Dump, 512 Mbytes, /), lvol3 (LVM, VxFS, 140 Mbytes, /), lvol4 (LVM, VxFS, 64 Mbytes, /tmp), lvol6 (LVM, VxFS, 232 Mbytes, /opt), lvol7 (LVM, VxFS, 720 Mbytes, /usr), lvol8 (LVM, VxFS, 272 Mbytes, /var), and vol01 (VxVM, Unused, 400 Mbytes, /).

Logical Volume	Volume Group	Type	Use	Total Mbytes	Mirror Copies	Mount Directory
bells	maroon	VxVM	VxFS	500		/lake
lvol1	vg00	LVM	HFS	84	0	/stand
lvol2	vg00	LVM	Swap/Dump	512	0	/
lvol3	vg00	LVM	VxFS	140	0	/
lvol4	vg00	LVM	VxFS	64	0	/tmp
lvol6	vg00	LVM	VxFS	232	0	/opt
lvol7	vg00	LVM	VxFS	720	0	/usr
lvol8	vg00	LVM	VxFS	272	0	/var
vol01	rootdg	VxVM	Unused	400		

Conversion Error Messages

A

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.

Error messages that you may encounter during conversion include the following:

- ◆ tAnalysis 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
- ◆ tAnalysis shows that there is insufficient private space available to convert this volume group
- ◆ tDevice device_name has the following bad blocks... Cannot convert LVM Volume Group
- ◆ tThe conversion process was unable to deactivate the volume group vol_grp_name
- ◆ tThis Volume Group contains one or more logical volumes with mirrored data
- ◆ tToo many LVM Volumes to convert in this LVM Volume Group
- ◆ tvgchange: Couldn't deactivate volume group /dev/vol_grp
- ◆ tvxdiskadm or vxvmconvert is already being run and these programs cannot run concurrently



▼ **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**

◆ *Description*

For successful conversion, all physical volumes in a volume group must be on-line, and all logical volumes must be active and accessible.

◆ *Action*

Make sure the physical volumes in a volume group are on-line and the logical volumes are active and not in use.

▼ **Analysis shows that there is insufficient private space available to convert this volume group**

◆ *Description*

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.

◆ *Action*

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.

▼ **Device device_name has the following bad blocks... Cannot convert LVM Volume Group**

◆ *Description*

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.

◆ *Action*

You have the following choices:

- ◆ Place a hardware call to have the disk replaced or have the hardware engineer use the diagnostics (if applicable for this disk type) to re-vector the block. Then clear the BBDIR as described below.
- ◆ If the errors were transient, or after the disk has been replaced, clear the BBDIR as follows:

a. Back up all the data associated with the physical volume.

b. Make a current backup of the LVM headers, for example:

```
# vgcfgbackup /dev/vg01
```

-
- c. Close all the logical volumes associated with the volume group and then deactivate the volume group.
 - d. Create a dummy link to the raw device, for example:

```
# ln /dev/rdisk/c0t0d0 /dev/rdisk/temp
```
 - e. Destroy the LVM headers using `pvcreate -f` on the dummy link, for example:

```
# pvcreate -f /dev/rdisk/temp
```
 - f. Remove the dummy link to the raw device, for example:

```
# rm /dev/rdisk/temp
```
 - g. Restore the headers from the lvmconf backup, for example:

```
# vgcfgrestore -n /dev/vg01 /dev/rdisk/c0t0d0
```
 - h. Activate the volume group and integrity check the data or recover from backup.

▼ **The conversion process was unable to deactivate the volume group vol_grp_name**

◆ *Description*

This indicates that the conversion process cannot deactivate the volume group.

◆ *Action*

The conversion cannot be completed without rebooting the machine. If you cannot afford to reboot, then choose abort and try again later.

▼ **This Volume Group contains one or more logical volumes with mirrored data**

◆ *Description*

If an administrator attempts to convert a Mirrored LVM Volume Group without a valid VxVM license installed, the conversion is not allowed.

◆ *Action*

Install the required license before attempting the conversion.



▼ **Too many LVM Volumes to convert in this LVM Volume Group**

◆ *Description*

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.

◆ *Action*

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.

▼ **vgchange: Couldn't deactivate volume group /dev/vol_grp**

◆ *Description*

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.

◆ *Action*

The system responds with an option to complete the conversion by rebooting the system.

▼ **vxdiskadm or vxvmconvert is already being run and these programs cannot run concurrently**

◆ *Description*

The system detects that the `vxdiskadm` or `vxvmconvert` program is already running.

◆ *Action*

Retry at a later time. Otherwise, if you are certain that no other users are running either of these programs, remove the file `.DISKADD.LOCK` from the `/var/spool/locks` directory to allow you to run `vxvmconvert`.

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 HP-UX product, similar to VxVM.

LIF

The Logical Interchange Format (LIF) is a HP mass-storage format that can be used for interchange of files among various HP computer systems. Each boot disk has a LIF directory that contains boot programs.



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 disk/UX

HP-UX software product that allows disk mirroring as part of LVM functionality. Mirror Disk/UX allows up to 2 mirror copies in a volume.

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 boot disks, the VGRA starts at sector 2144. 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 54
- B**
 - back up
 - volume group 50
 - Backup
 - vgcfgbackup 16
- C**
 - coexistence
 - VxVM and LVM disks 7
 - commands
 - vxedit 42
 - configuration
 - LVM 9
 - configuration VxVM 9
 - conversion
 - errors 65
 - speed 39
 - vxvmconvert 11
- D**
 - deactivate
 - disk group 50
 - volume group 50
 - deport
 - disk group 50
 - destroy
 - disk group 50
 - disable
 - mirror 58
 - disk
 - evacuate 57
 - offline 57
 - online 57
 - recover
 - 57
 - rename 57
 - replace 57
 - disk group 48
 - rename 58
 - disk groups 1
 - disk headers 11
 - disks 1
 - coexistence 7
 - mirroring 51
 - display
 - disk group 52
 - DMP 57
 - logical volume 52
 - physical volume 52
 - volume 52
 - VxVM volumes 52
 - DMP
 - display 57
 - Multipathing 54
 - Dynamic Multipathing 47, 54
 - E**
 - equivalent command 42
 - error messages
 - Analysis indicates that this volume group cannot be converted 66
 - Analysis shows that there is insufficient private space available to convert this volume group 66
 - Cannot convert LVM Volume Group 66
 - Device device_name has the following bad blocks 66
 - The conversion process was unable to deactivate the volume group vol_grp_name 67
 - This Volume Group contains one or more logical volumes with mirrored data 67
 - Too many LVM Volumes to convert in this LVM Volume Group 68
 - vgchange Couldn't deactivate volume



-
- group /dev/vol_grp 68
 - vxdiskadm or vxvmconvert is already being run and these programs cannot run concurrently 68
 - Example
 - analyze LVM groups 26
 - conversion 26
 - failed conversion 26
 - list 26
 - list disk information 26
 - list LVM volume group information 26
 - listvg 26
 - VLM to VxVM 26
 - vxprint output 26
 - example
 - Failed Analysis 26
 - export
 - volume group 50
 - extend
 - volume group 51
 - F**
 - File System 17
 - file system 49
 - G**
 - GUI
 - SAM 47, 61
 - Storage Administrator 47
 - VERITAS Volume Manager Storage Administrator 61
 - H**
 - Hot Relocation 47
 - I**
 - import
 - disk group 49
 - volume group 49
 - increase
 - disk group 50
 - swap 50
 - J**
 - join
 - subdisk 58
 - L**
 - list LVM 26
 - Logical Volume 4, 5
 - logical volume
 - convert 55
 - split 55
 - synchronize 55
 - Logical Volume Manager 1
 - lvchange 42
 - lvcreate 42
 - lvextend 42
 - lvlnboot 42
 - LVM 1
 - metadata 12
 - LVM metadata 5
 - LVM names
 - symbolic names 19
 - LVM VGRA 11
 - lvmerge 43
 - lvreduce 43
 - lvremove 43
 - lvsplit 43
 - lvsync 43
 - M**
 - manual page 41
 - mapping
 - LVM device nodes 18
 - VxVM device nodes 18
 - messages
 - error 65
 - mirror
 - disable 58
 - disks 1
 - logical volume 54
 - plex 54
 - remove 58
 - repair 58
 - MirrorDisk/UX 47
 - mirroring 47
 - mirroring and striping 47
 - mirrors
 - creating 51
 - creating for root disk 51
 - move
 - subdisk 58
 - O**
 - Online Migration 47
 - P**
 - physical volumes 4, 5
 - Private Region 5
 - pvchange 44
 - pvcreate 44



-
- pvdisk 44
 - pvmove 44
- R**
- RAID-5 47
 - reduce 49
 - volume group 51
 - remove
 - disk 53
 - volume 53
 - volume group 50, 53
 - rename
 - disk group 58
 - repair
 - mirror 58
 - restore
 - volume group 50
 - resynchronize
 - volumes 56
 - root disk
 - configuring for VxVM 12
 - reconfiguring for LVM 12
 - root disks
 - mirroring 51
 - rootability
 - configuring a VxVM root disk 12
 - restoring a LVM root disk 12
 - root disk 12
 - root volume 12
- S**
- SAM 47
 - vgdisplay 15
 - split
 - subdisk 58
 - Storage 47
 - Storage Administrator 47
 - subdisk
 - join 58
 - move 58
 - split 58
- T**
- Technical assistance viii
 - tool
 - vxxmconvert 11
 - tools
 - vxdiskadm 9
 - vxxmconvert 9
 - troubleshoot
- V**
- errors 65
 - VERITAS Volume Manager 1
 - vgchange 45
 - vgcreate 45
 - vgdisplay 45
 - vgexport 46
 - vgextend 45
 - vgimport 46
 - vgreduce 45
 - vgremove 46
 - vgscan 45
 - vgsync 45
 - volume
 - concatenated 52
 - logical 52
 - RAID-5 52
 - reduce 49
 - striped 52
 - Volume Manager
 - features 2
 - volumes 1
 - Logical Volume 4, 5
 - physical volumes 4, 5
 - vxassist 42, 43
 - vxchg 45
 - vxdisk 44
 - vxdisk set 44
 - vxdiskadd 44, 45
 - vxdiskadm 44
 - vxedit 42, 43, 44
 - vxevac 44
 - VxFS 49
 - vxinfo 45
 - vxmend 46
 - vxplex 46
 - vxprint 45
 - vxrecover 43
 - vxresize 42
 - VxVM 1
 - features 2
 - metadata 12
 - VxVM names
 - symbolic link 19
 - VxVM volumes
 - resynchronize 56
 - vxvol 43, 46

