

# **VERITAS Storage Foundation™ 4.1 Cross-Platform Data Sharing**

## **Administrator's Guide**

N13116C

March 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](http://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.

### Sun Microsystems Trademarks

Sun, Solaris, SunOS, Java, Sun Java System Cluster, Sun StorEdge, Solstice DiskSuite, Sun Fire, Sun Enterprise, Online: Backup, and Netra are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.



---

# Contents

<b>Preface</b> .....	<b>vii</b>
How This Guide Is Organized .....	viii
Conventions .....	ix
Getting Help .....	x
Documentation Feedback .....	x
 <b>Chapter 1. Overview</b> .....	<b>1</b>
General Concepts .....	1
Sharing Data Across Platforms .....	2
Sector Size .....	2
Block Size .....	3
Operating System Data .....	3
CDS Disk Access and Format .....	3
CDS Disk Types .....	4
Private and Public Regions .....	4
Disk Access Type auto .....	5
Platform Block .....	5
AIX Coexistence Label .....	5
HP-UX Coexistence Label .....	5
VxVM ID Block .....	5
CDS Disk Groups .....	5
Device Quotas .....	6
Minor Device Numbers .....	7
Non-CDS Disk Groups .....	7
Disk Group Alignment .....	7
Alignment Values .....	8



Alignment Value 1 .....	8
Alignment Value 8K .....	8
Dirty Region Log (DRL) Alignment .....	8
<b>Chapter 2. Setting up Your System .....</b>	<b>11</b>
Creating CDS Disks from Uninitialized Disks .....	11
Using vxdisksetup .....	11
Using vxdiskadm .....	12
Creating CDS Disks from Initialized VxVM Disks .....	12
Disk Not in a Disk Group .....	12
Disk Already in a Disk Group .....	12
Creating CDS Disk Groups .....	13
Using vxdg init to Create CDS Disk Groups .....	13
Using vxdiskadm to Create CDS Disk Groups .....	13
Converting a Non-CDS Disk to a CDS Disk .....	14
Converting a Non-CDS Disk Group to a CDS Disk Group .....	15
Notes .....	16
Examples .....	16
Licensing .....	17
Defaults Files .....	17
<b>Chapter 3. Maintaining Your System .....</b>	<b>21</b>
Disk Tasks .....	22
Changing the Default Disk Format Setting .....	22
Restoring CDS Disk Labels .....	22
Disk Group Tasks .....	24
Changing the Alignment of a Disk Group by Encapsulation .....	24
Object Alignment During Volume Creation .....	24
Changing the Alignment of a Non-CDS Disk Group .....	25
Determining the Setting of the CDS Attribute .....	26
Joining Disk Groups .....	26



---

Moving Objects Between CDS and Non-CDS Disk Groups .....	26
Moving Objects Between CDS Disk Groups .....	27
Changing Default CDS Setting for Disk Group Creation .....	27
Creating Non-CDS Disk Groups .....	27
Upgrading an Older Version Non-CDS Disk Group .....	27
Replacing a Disk in a CDS Disk Group .....	28
Setting Device Quotas for CDS Disk Groups .....	28
Setting DRL Map Size and Log Size .....	28
Creating a DRL Log .....	29
Using vxassist to Create a DRL Log .....	29
Using vxvol to Set a DRL Map Length .....	30
Displaying Information .....	30
Displaying Traditional DRL Map Size and Log Size .....	31
Displaying the Disk Group Alignment .....	31
Displaying Volume Log Map Values .....	31
Listing Offset and Length Information .....	32
Listing CDS Disk Groups .....	32
Listing Disks .....	32
Displaying Device Quotas for CDS Disk Groups .....	33
Default Activation of Shared Disk Groups .....	33
Creating CDS Disk Groups Using vxdg split .....	33
Additional Considerations When Importing CDS Disk Groups .....	34
<b>Chapter 4. File System Considerations .....</b>	<b>35</b>
Considerations About Data in the File System .....	36
File System Migration .....	36
Using the fscdstask Script .....	37
One-Time Migration of a File System .....	38
Ongoing Migration of a File System .....	39
Ceasing Ongoing Migration .....	40



---

When to Convert a File System .....	40
Converting a File System .....	41
<b>Appendix A. Cross-Platform Transfer .....</b>	<b>45</b>
Alignment Value and Block Size .....	45
Default Activation of Shared Disk Groups .....	45
Disk Group Alignment and Encapsulated Disks .....	45
Importing Disk Groups Between Linux and Non-Linux Machines .....	46
<b>Appendix B. Error Codes and Error Recovery .....</b>	<b>47</b>
<b>Glossary .....</b>	<b>51</b>
<b>Index .....</b>	<b>57</b>



# Preface

---

The *VERITAS Storage Foundation Cross-Platform Data Sharing Administrator's Guide* describes how to use the VERITAS Cross-platform Data Sharing (CDS) feature.

This guide is intended for system administrators responsible for configuring and maintaining systems using the VERITAS CDS feature.

This guide assumes that you have a:

- ◆ Basic understanding of system administration.
- ◆ Basic knowledge of the operating systems on all supported platforms.
- ◆ Basic familiarity with VERITAS Volume Manager (VxVM) and storage management concepts.

---

**Note** Most VERITAS Volume Manager commands require superuser or other appropriate privileges.

---



## How This Guide Is Organized

This guide contains information about the following topics:

- ◆ Chapter 1, “[Overview](#),” gives an introduction to CDS concepts.
- ◆ Chapter 2, “[Setting up Your System](#),” presents considerations when transferring data between platforms.
- ◆ Chapter 3, “[Maintaining Your System](#),” describes how to use CDS.
- ◆ Chapter 4, “[File System Considerations](#),” presents File System considerations
- ◆ Appendix A, “[Cross-Platform Transfer](#)” contains cross-platform considerations for the various supported platforms.
- ◆ Appendix B, “[Error Codes and Error Recovery](#),” describes the error messages output by CDS.
- ◆ The [Glossary](#) defines terms that relate to CDS.

Refer to the *Release Notes* for information about the other documentation that is provided with this product.





# 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.
<b>monospace (bold)</b>	Indicates user input.	<b># ls pubs</b> <b>C:\&gt; dir pubs</b>
<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.
<b>bold</b>	Depicts GUI objects, such as fields, list boxes, menu selections, etc. Also depicts GUI commands.	Enter your password in the <b>Password</b> field. Press <b>Return</b> .
blue text	Indicates hypertext links.	See " <a href="#">Getting Help</a> " on page x.
#	Unix superuser prompt (all shells).	<b># cp /pubs/4.0/user_book /release_mgnt/4.0/archive</b>
C:\>	Windows user prompt.	<b>C:\&gt; copy \pubs\4.0\user_book c:\release_mgnt\4.0\archive</b>



## 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](mailto: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>.

# Overview

---

1

This chapter presents an overview of Cross-platform Data Sharing (CDS):

- ◆ [General Concepts](#)
- ◆ [CDS Disk Access and Format](#)
- ◆ [CDS Disk Groups](#)

## General Concepts

CDS provides you with a foundation for moving data between different systems within a heterogeneous environment. The machines may be running HP-UX, AIX, Linux or the Solaris™ operating system (OS), and they may all have direct access to physical devices holding data. CDS allows VERITAS products and applications to access data storage independently of the operating system platform, enabling them to work transparently in heterogeneous environments.

The following levels in the device hierarchy, from disk through file system, must provide support for CDS to be used:

- ◆ End-user applications — application level
- ◆ VERITAS File System (VxFS) — file system level
- ◆ VERITAS Volume Manager (VxVM) — volume level
- ◆ Operating system — device level

Although CDS allows data volumes and their contents to be easily migrated between heterogeneous systems, it does not enable concurrent access from different types of platform unless this is supported at all levels that are required.

CDS utilizes a new disk type (*auto:cdsdisk*). To effect data sharing, VxVM supports a new disk group attribute (*cds*) and also supports different OS block sizes.

CDS is a license-enabled feature applied at the disk group level.

---

**Note** VERITAS File System 4.1 for Solaris x86 Platform Edition does not support CDS at the file system level.

---



## Sharing Data Across Platforms

It must be emphasized that, while volumes can be exported across platforms, the data on the volumes can be shared only if data sharing is supported at the application level. That is, to make data sharing across platforms possible, it must be supported throughout the *entire* software stack.

For example, if a VxFS file system on a VxVM volume contains files comprising a database, then:

- ◆ Disks can be recognized (as *cds* disks) across platforms.
- ◆ Disk groups can be imported across platforms
- ◆ The file system can be mounted on different platforms

However, it is very likely that, because of the inherent characteristics of databases, you may not be able to start up and use the database on a platform different from the one on which it was created.

Another example is where an executable file, compiled on one platform, can be accessed across platforms (using CDS), but may not be executable on a different platform.

---

**Note** You do not need a file system in the stack if the operating system provides access to raw disks and volumes, and the application can utilize them. In this way, databases and other applications can have their data components built on top of raw volumes without having a file system to store their data files.

---

## Sector Size

Sector size is an attribute of a disk drive (or SCSI LUN for an array-type device), which is set when the drive is formatted. Sectors are the smallest addressable unit of storage on the drive, and are the units in which the device performs I/O. The sector size is significant because it defines the atomic I/O size at the device level. Any multi-sector writes which VxVM submits to the device driver are not guaranteed to be atomic (by the SCSI subsystem) in the case of system failure.

## Block Size

The block size is a platform-dependent value that is greater than or equal to the sector size. Each platform accesses the disk on block boundaries and in quantities that are multiples of the block size. Data that is created on one platform, and then accessed by a platform of a different block size, can suffer from the following problems:

- ◆ Addressing issues
  - ◆ The data may not have been created on a block boundary compatible with that used by the accessing platform.
  - ◆ The accessing platform cannot address the start of the data.
- ◆ Bleed-over issues

The size of the data written may not be an exact multiple of the block size used by the accessing platform. Therefore the accessing platform cannot constrain its I/O within the boundaries of the data on disk.

## Operating System Data

Some Operating Systems (OS) require OS-specific data on disks in order to recognize and control access to the disk.

## CDS Disk Access and Format

For a disk to be accessible by multiple platforms, the disk must be consistently recognized by the platforms, and all platforms must be capable of performing I/O on the disk. CDS disks contain specific content at specific locations to identify or control access to the disk on different platforms. The same content and location are used on all CDS disks, independent of the platform on which the disks are initialized.

In order for a disk to be initialized as, or converted to a CDS disk, it must satisfy the following requirements:

- ◆ Must be a SCSI disk that supports Mode Sense
- ◆ Cannot be an EFI disk
- ◆ Must be the entire physical disk (LUN)
- ◆ Only one volume manager (such as VxVM) can manage a physical disk (LUN)
- ◆ There can be no disk partition (slice) which is defined, but which is not configured on the disk
- ◆ Cannot contain a volume whose use-type is either `root` or `swap` (for example, it cannot be a boot disk)



**Note** The CDS conversion utility, `vxcdsconvert`, is provided to convert non-CDS VM disk formats to CDS disks, and disk groups with a version number less than 110 to disk groups that support CDS disks. See [“Setting up Your System”](#) on page 11 for more details.

---

## CDS Disk Types

The CDS disk format, `cdsdisk`, is recognized by all VxVM platforms (including Windows). This is the default disk format for all newly created VM disks unless overridden in a defaults file (see [“Defaults Files”](#) on page 17). The `vxcdsconvert` utility is provided to convert other disk formats and types to CDS.

**Note** Disks with format `cdsdisk` can only be added to disk groups with version 110 or later.

---

## Private and Public Regions

A VM disk usually has a private and a public region.

The private region is a small area on the disk where VxVM configuration information is stored, such as a disk header label, configuration records for VxVM objects (such as volumes, plexes and subdisks), and an intent log for the configuration database. The default private region size is 2048 blocks (equivalent to 1024 kilobytes on AIX, Linux and the Solaris OS, and 2048 kilobytes on HP-UX), which is large enough to record the details of about 4000 VxVM objects in a disk group.

The public region covers the remainder of the disk, and is used for the allocation of storage space to subdisks.

The private and public regions are aligned and sized in multiples of 8K to permit the operation of CDS. The alignment of VxVM objects within the public region is controlled by the disk group alignment attribute. The value of this attribute must also be 8k to permit the operation of CDS.

**Note** With other (non-CDS) VxVM disk formats, the private and public regions are aligned to the platform-specific OS block size.

---

## Disk Access Type *auto*

The disk access (DA) disk type *auto* supports multiple disk formats, including `cdsdisk`, which is supported across all platforms. It is associated with the DA records created by the VxVM auto-configuration mode. Disk type *auto* automatically determines which format is on the disk.

## Platform Block

The Platform Block resides on disk sector 0, and contains data specific to the Operating System for the platforms. It is necessary for proper interaction with each of those platforms. The platform block allows a disk to perform as if it was initialized by each of the specific platforms.

## AIX Coexistence Label

The AIX Coexistence label resides on the disk, and identifies the disk to the AIX volume manager (LVM) as being controlled by VxVM.

## HP-UX Coexistence Label

The HP-UX Coexistence label resides on the disk, and identifies the disk to the HP volume manager (LVM) as being controlled by VxVM.

## VxVM ID Block

The VxVM ID block resides on the disk, and indicates the disk is under VxVM control. It provides dynamic VxVM private region location and other information.

## CDS Disk Groups

A CDS disk group allows cross-platform data sharing of VxVM objects, so that data written on one of the supported platforms may be accessed on any other supported platform. A CDS disk group is composed only of CDS disks (VM disks with the disk format `cdsdisk`), and is only available for disk group version 110 and greater.

---

**Note** The CDS conversion utility, `vxcdsconvert`, is provided to convert non-CDS VM disk formats to CDS disks, and disk groups with a version number less than 110 to disk groups that support CDS disks. See [“Setting up Your System”](#) on page 11 for more details.

---



All VxVM objects in a CDS disk group are aligned and sized so that any system can access the object using its own representation of an I/O block. The CDS disk group uses a platform-independent alignment value to support system block sizes of up to 8K. See [“Disk Group Alignment”](#) on page 7 for details.

CDS disk groups can be:

- ◆ Initialized on one system and then used “as-is” by VxVM on a system employing a different type of platform.
- ◆ Imported (in a serial fashion) by Linux, Solaris, AIX, and HP-UX systems.
- ◆ Private, shared, or distributed (SAN VM).
- ◆ Shared by CVM.

You cannot include the following disks or volumes in a CDS disk group:

- ◆ Volumes of usage type `root` and `swap`. (This means that you cannot use CDS to share boot devices.)
- ◆ Encapsulated disks.

---

**Note** On Solaris and Linux systems, the process of *disk encapsulation* places the slices or partitions on a disk (which may contain data or file systems) under VxVM control. On AIX and HP-UX systems, LVM volumes may be similarly be converted to VxVM volumes.

---

## Device Quotas

Device quotas limit the number of objects in the disk group which create associated device nodes in the file system. (This is useful for disk groups which share serially between Linux with a pre-2.6 kernel and other supported platforms. Prior to the 2.6 kernel, Linux supported only 256 minor devices per major device.)

You can limit the number of devices that can be created in a given CDS disk group by setting the device quota (refer to [“Setting Device Quotas for CDS Disk Groups”](#) on page 28).

When you create a device, an error is returned if the number of devices would exceed the device quota. You then either need to increase the quota, or remove some objects using device numbers, before the device can be created.

See also [“Displaying Device Quotas for CDS Disk Groups”](#) on page 33 for instructions on displaying the device quota value.



## Minor Device Numbers

Importing a disk group will fail if it will exceed the maximum devices for that platform.

---

**Note** There is a large disparity between the maximum number of devices allowed for devices on the Linux platform with a pre-2.6 kernel, and that for other supported platforms.

---

## Non-CDS Disk Groups

Any version 110 (or greater) disk group (DG) can contain both CDS and non-CDS disks. However, only version 110 (or greater) disk groups composed entirely of CDS disks have the ability to be shared across platforms. Whether or not that ability has been enabled is a license-controlled attribute of the disk group (the *cds* attribute). Enabling that attribute causes a non-CDS disk group to become a CDS disk group.

Although a non-CDS disk group can contain a mixture of CDS and non-CDS disks having dissimilar private region alignment characteristics, its disk group alignment will still direct how all subdisks are created.

## Disk Group Alignment

One of the attributes of the disk group is the block *alignment*, which represents the largest block size supported by the disk group. The alignment constrains attributes of the objects within the disk group, such as:

- ◆ Subdisk; how subdisks are positioned on the drive (*offset*), and their size granularity (*length*)
- ◆ Plex offset
- ◆ Volume length
- ◆ Log length
- ◆ Stripe width

The disk group alignment is assigned at disk group creation time (see “[Maintaining Your System](#)” on page 21).



## Alignment Values

The disk group block alignment has two values: 1 block or 8k (8 kilobytes).

### Alignment Value 1

All disk group versions before version 110 must have a alignment value of 1.

Disk groups have an alignment value of 1 after upgrading from pre-version 110 to version 110 or later.

Encapsulated disks, which require disk cylinder alignment, have an alignment value of 1.

Non-CDS disk groups version 110 and later can have either a value of 1 block or 8k.

### Alignment Value 8K

Non-CDS disk groups version 110 and later can have either a value of 1 block or 8k.

All CDS disk groups have an alignment value of 8k.

The alignment for all newly initialized disk groups in VxVM 4.0 or a later release is 8k. This value (which is used when creating the disk group) cannot be changed. However, the disk group alignment can be subsequently changed (refer to [“Changing the Alignment of a Non-CDS Disk Group”](#) on page 25).

---

**Note** The default usage of `vxassist` is to set the `layout=diskalign` attribute on all platforms. This is ignored on 8K-aligned disk groups, which means that scripts relying on the default may fail.

---

## Dirty Region Log (DRL) Alignment

The location and size of each map within a Dirty Region Log (DRL) must not violate the disk group alignment for the disk group (containing the volume to which the DRL is associated). This means that the region size and alignment of each DRL map must be a multiple of the disk group alignment, which for CDS disk groups is 8K. (Features utilizing the region size can impose additional minimums and size increments over and above this restriction, but cannot violate it.)

In a version 110 disk group, a traditional DRL volume must have:

- ◆ Minimum region size of 512K
- ◆ Incremental region size of 64K

In a version 110 disk group, a version 20 DCO volume must have:

- ◆ Minimum region size of 16K
- ◆ Incremental region size of 8K

---

**Note** The map layout within a Data Change Object (DCO) volume changed with the release of VxVM 4.0 to version 20. This can accommodate both FastResync and DRL maps within the DCO volume. The original version 0 layout for DCO volumes only accommodates FastResync maps.

---





## Setting up Your System

---

This chapter describes how to set up your initial CDS system. It includes information on:

- ◆ [Creating CDS Disks from Uninitialized Disks](#)
- ◆ [Creating CDS Disks from Initialized VxVM Disks](#)
- ◆ [Creating CDS Disk Groups](#)
- ◆ [Converting a Non-CDS Disk to a CDS Disk](#)
- ◆ [Converting a Non-CDS Disk Group to a CDS Disk Group](#)
- ◆ [Licensing](#)
- ◆ [Defaults Files](#)

## Creating CDS Disks from Uninitialized Disks

To create a CDS disk from an uninitialized disk, use one of the following commands:

- ◆ `vxdisksetup`
- ◆ `vxdiskadm`

### Using `vxdisksetup`

To create a CDS disk, use the `vxdisksetup` command:

```
# vxdisksetup -i disk
```

The format defaults to `cdsdisk` unless overridden by the `/etc/default/vxdisk` file (refer to “[Defaults Files](#)” on page 17).

Alternatively, you can override the default format by specifying the `format` attribute:

```
# vxdisksetup -i disk format=cdsdisk
```

See the `vxdisksetup(1M)` manual page for more information.



## Using vxdiskadm

Use the `disk init` option. You are prompted to specify the format.

## Creating CDS Disks from Initialized VxVM Disks

If the disk is already initialized, there are two cases to consider:

- ◆ [Disk Not in a Disk Group](#)
- ◆ [Disk Already in a Disk Group](#)

### Disk Not in a Disk Group

If the disk is not in a disk group:

1. Run the following command to remove the VM disk format for the disk:

```
# vxdiskunsetup disk
```

This is necessary as non-auto types cannot be reinitialized by `vxdisksetup`.

2. If the disk is listed in the `/etc/vx/darecs` file, remove its disk access (DA) record using the command:

```
# vxdisk rm disk
```

(Disk access records that cannot be configured by scanning the disks are stored in an ordinary file, `/etc/vx/darecs`, in the root file system. Refer to the `vxintro(1M)` manual page for more information.)

3. Rescan for the disk using this command:

```
# vxdisk scandisks
```

4. Finally, run this command to set up the disk:

```
# vxdisksetup -i disk
```

### Disk Already in a Disk Group

If the disk is already in a disk group, run `vxcdsconvert`, as described in [“Converting a Non-CDS Disk to a CDS Disk”](#) on page 14.

## Creating CDS Disk Groups

You can create a CDS disk group (DG) in two ways:

- ◆ [Using vx dg init to Create CDS Disk Groups](#)
- ◆ [Using vx diskadm to Create CDS Disk Groups](#)

### Using vx dg init to Create CDS Disk Groups

To create a CDS disk group, use the `vx dg init` command:

```
# vx dg init diskgroup disklist
```

The format defaults to a CDS disk group unless overridden by the `/etc/default/vx dg` file.

Alternatively, you can override the default format by specifying the `cds` attribute:

```
# vx dg init diskgroup disklist cds=on
```

---

**Note** The disk group version must be 110 or greater.

---

See the `vx dg(1M)` manual page for more information.

### Using vx diskadm to Create CDS Disk Groups

You can create a CDS disk group when using `vx diskadm` to initialize a disk. Specify that the disk group should be a CDS disk group when prompted.

---

**Note** You cannot create a CDS disk group using the `vx diskadm` command as part of encapsulating, or LVM conversion.

---

When the `vx diskadm` command initializes a disk into an existing CDS disk group, the disk must have format `cdsdisk`.

The CDS attribute for the disk group remains unchanged (that is, you are not given the option to change it).

---

**Note** When initializing a disk, if the target disk group is an existing CDS disk group, `vx diskadm` will only allow the disk to be initialized as a CDS disk. If the target disk group is a non-CDS disk group, the disk can be initialized as either a CDS disk or a non-CDS disk.

---



## Converting a Non-CDS Disk to a CDS Disk

Use the CDS conversion utility (`vxcdsconvert`) to convert non-CDS disks to CDS disks, to make them portable between different operating systems that are running VxVM with the CDS feature.

```
# vxcdsconvert -g diskgroup [-A] [-d defaults_file] \
  [-o novolstop] disk name [attribute=value] ...

# vxcdsconvert -g diskgroup [-A] [-d defaults_file] \
  [-o novolstop] alldisks [attribute=value] ...
```

---

**Note** The existing DA record needs to have a type of `auto` in order to be re-initialized as a CDS disk.

---

You can use `vxcdsconvert` to convert CDS disks either singly, or by disk group:

- ◆ Use the `disk` keyword to specify a single disk for conversion. You might consider this option, rather than converting the entire disk group, if you do not want a CDS disk group. For example:
  - ◆ Where a disk in the non-CDS disk group has cross-platform exposure, and you want other VxVM nodes to recognize the disk and not assume it is available for initialization.
  - ◆ LVM on HP-UX and AIX, and Windows VxVM nodes need to recognize CDS disks, but not attempt to initialize or manage them.
  - ◆ The intention is to move the disk into an existing CDS disk group.
- ◆ Use the `alldisks` keyword to convert all the non-CDS disks in a disk group.

In addition, you can use the `-o novolstop` option to perform the conversion on-line (that is, while use of the disk continues). However, this may greatly increase the amount of time (and work) required for disk conversion as it may involve evacuating objects from the disk, reinitializing the disk, and relocating objects back to disk.

Alternatively, stop the application, and perform the conversion off-line. This requires minimal offline time.

---

**Note** Disk conversion may take significantly longer if the `-o novolstop` option is specified, depending on the subdisk layout of the disk.

---

Refer to the `vxcdsconvert(1M)` manual page for further information on options, attributes, and keywords.



## Converting a Non-CDS Disk Group to a CDS Disk Group

Use the CDS conversion utility (`vxcdsconvert`) to make a VxVM non-CDS disk group (DG) portable between different operating systems that are running versions of VxVM with the CDS feature. This allows existing data to be made shareable in a CDS environment.

```
# vxcdsconvert -g diskgroup [-A] [-d defaults_file] \
  [-o novolstop] alignment [attribute=value] ...

# vxcdsconvert -g diskgroup [-A] [-d defaults_file] \
  [-o novolstop] group [attribute=value] ...
```

CDS provides the `alignment` and `group` keywords for disk group conversion:

- ◆ Use the `alignment` keyword to specify alignment conversion. In this case, disks are not converted, and an object relay layout is performed on the disk group. A successful completion results in an 8K-aligned disk group. You might consider this option, rather than converting the entire disk group, if you want to reduce the amount of work to be done for a later full conversion to CDS disk group.
- ◆ Use the `group` keyword to specify group conversion. This implies `alldisk`, and will perform that function prior to object relay layout. All the non-CDS disks in the disk group are converted.

In addition, you can use the `-o novolstop` option to perform the conversion on-line (that is, while use of the disk group continues). However, for a group conversion, this may greatly increase the amount of time (and work) required for conversion.

Alternatively, stop the application, and perform the conversion off-line. This requires minimal offline time.

Note the following:

- ◆ The disk group must be in pristine condition. That is:
  - ◆ It has no dissociated or disabled objects.
  - ◆ No sparse plexes are present.
  - ◆ There are no volumes requiring recovery or having pending snapshot operations.
  - ◆ There are no objects in an error state.
- ◆ Stopped (but startable) volumes will be started, for the duration of the conversion only.
- ◆ Conversion has the following side effects:
  - ◆ Any objects created with `layout=diskalign` can no longer be disk aligned.
  - ◆ Encapsulated disks may lose the ability to be unencapsulated.



- ◆ Fine performance tuning may be lost as data may have migrated (and even migrated to different disks).

Please refer to the `vxcdsconvert(1M)` manual page for information on options, attributes, and keywords.

## Notes

- ◆ A non-CDS disk group will be upgraded (using the `vx dg upgrade` command). Also, if the disk group was originally created by the conversion of an LVM Volume Group (VG), then rolling back to the original LVM VG is not an option. If you decide to go through with the conversion, the rollback records for the disk group will be removed, so that an accidental rollback to an LVM VG cannot be done.
- ◆ If the non-CDS disk group has one or more disks that are not CDS disks, these disks are converted to CDS disks. If you do not want one or more of these disks to be converted, you need to move them out of the disk group (for example, using `vx dg move` or `split`) prior to invoking the `vxcdsconvert group` command.
- ◆ If the non-CDS disk group does not have a CDS-compatible disk group alignment, the objects need to go through relayout, so they have a CDS-compatible alignment.
- ◆ Unless the `novolstop` option is specified, applications using disks which require format conversion must be terminated for the duration of the disk conversion process.
- ◆ Use of the `novolstop` option for the disk conversion piece may add a large amount of work, as objects may need to be evacuated and then unrelocated back.

## Examples

- ◆ Check if the disk group `mydg` and all its disks can be made portable, using the command:  
  

```
# vxcdsconvert -g mydg -A group
```
- ◆ Make the disk group `mydg` and all its disks portable while its volumes are still online. You can avoid using the `novolstop` option during disk conversion by using the following sequence of commands:
  - ◆ Stop all applications that are using volumes on the disks requiring format conversion.
  - ◆ Enter the command:  
  

```
# vxcdsconvert -g mydg alldisks
```
  - ◆ Restart applications.

- ◆ Enter the command:

```
# vxcdsconvert -g mydg group
```

- ◆ Make the disk group, anodg, and all its disks, portable while its volumes are still online, and allow subdisks to be evacuated to disks anodg11 through anodg14 if required:

```
# vxcdsconvert -g anodg -o novolstop group \
  move_subdisks_ok=yes evac_subdisks_ok=yes \
  evac_disk_list=anodg11,anodg12,anodg13,anodg14
```

## Licensing

The ability to create or import a CDS disk group is controlled by a CDS license. CDS licenses are included as part of the VERITAS Storage Foundation license.

You can verify the CDS enabling license by running the `vxlicrep` command, and looking for this line in the output:

```
Cross-platform Data Sharing      = Enabled
```

## Defaults Files

Several system defaults files in the `/etc/default` directory are of importance for specifying the alignment of VxVM objects, the initialization or encapsulation of VM disks, the conversion of LVM disks, and the conversion of disk groups and their disks to the CDS-compatible format:

- ◆ `/etc/default/vxassist`

Specifies default values for the following parameters to the `vxcdsconvert` command that have an effect on the alignment of VxVM objects: `dgalignment_checking`, `diskalign`, and `nodiskalign`. See [“Object Alignment During Volume Creation”](#) on page 24, and the `vxassist(1M)` manual page for more information.



◆ `/etc/default/vxcdsconvert`

Specifies default values for the following parameters to the `vxcdsconvert` command: `evac_disk_list`, `evac_subdisks_ok`, `min_split_size`, `move_subdisks_ok`, `privlen`, and `split_subdisks_ok`. The following is a sample `vxcdsconvert` defaults file:

```
evac_subdisks_ok=no
min_split_size=64k
move_subdisks_ok=yes
privlen=2048
split_subdisks_ok=move
```

An alternate defaults file can be specified by using the `-d` option with the `vxcdsconvert` command. See the `vxcdsconvert(1M)` manual page for more information.

◆ `/etc/default/vxdg`

Specifies default values for the `cds`, `default_activation_mode` and `enable_activation` parameters to the `vxdg` command. (The `default_activation_mode` and `enable_activation` parameters are only used with shared disk groups in a cluster.) The following is a sample `vxdg` defaults file:

```
cds=on
```

See the `vxdg(1M)` manual page for more information.

◆ `/etc/default/vxdisk`

Specifies default values for the `format` and `privlen` parameters to the `vxdisk` and `vxdisksetup` commands. These commands are used when disks are initialized by VxVM for the first time. They are also called implicitly by the `vxdiskadm` command and the VERITAS Enterprise Administrator (VEA) GUI. The following is a sample `vxdisk` defaults file:

```
format=cddisk
privlen=2048
```

See the `vxdisk(1M)` and `vxdisksetup(1M)` manual pages for more information.

◆ `/etc/default/vxencap`

Specifies default values for the `format`, `privlen`, `privoffset` and `puboffset` parameters to the `vxencap` command. This command is used when disks with existing partitions or slices are encapsulated, or when LVM disks are converted to VM disks. It is also called implicitly by the `vxdiskadm` and `vxconvert` commands, and by the VEA. The following is a sample `vxencap` defaults file:

```
format=sliced
privlen=4096
privoffset=0
puboffset=1
```

See the `vxencap(1M)` and `vxconvert(1M)` manual pages (where available) for more information.

In the defaults files, a line that is empty, or that begins with a `#` character in the first column, is treated as a comment, and is ignored.

Apart from comment lines, all other lines must define attributes and their values using the format *attribute=value*. Each line starts in the first column, and is terminated by the value. No white space is allowed around the `=` sign.





The section describes the following CDS tasks performed by the system administrator:

- ◆ Disk Tasks
  - ◆ Changing the Default Disk Format Setting
  - ◆ Restoring CDS Disk Labels
- ◆ Disk Group Tasks
  - ◆ Changing the Alignment of a Disk Group by Encapsulation
  - ◆ Object Alignment During Volume Creation
  - ◆ Changing the Alignment of a Non-CDS Disk Group
  - ◆ Determining the Setting of the CDS Attribute
  - ◆ Joining Disk Groups
  - ◆ Moving Objects Between CDS and Non-CDS Disk Groups
  - ◆ Changing Default CDS Setting for Disk Group Creation
  - ◆ Creating Non-CDS Disk Groups
  - ◆ Upgrading an Older Version Non-CDS Disk Group
  - ◆ Replacing a Disk in a CDS Disk Group
  - ◆ Setting Device Quotas for CDS Disk Groups
- ◆ Setting DRL Map Size and Log Size
- ◆ Displaying Information
  - ◆ Displaying the Disk Group Alignment
  - ◆ Displaying Volume Log Map Values
  - ◆ Listing Offset and Length Information
  - ◆ Listing CDS Disk Groups
  - ◆ Listing Disks



- ◆ [Displaying Device Quotas for CDS Disk Groups](#)
- ◆ [Creating CDS Disk Groups Using vx dg split](#)
- ◆ [Default Activation of Shared Disk Groups](#)
- ◆ [Additional Considerations When Importing CDS Disk Groups](#)

## Disk Tasks

The following disk tasks are supported:

- ◆ [Changing the Default Disk Format Setting](#)
- ◆ [Restoring CDS Disk Labels](#)

### Changing the Default Disk Format Setting

When disks are put under VxVM control, they are formatted with the default `cdsdisk` layout. This happens during the following operations:

- ◆ Initialization of disks
- ◆ Encapsulation of disks with existing partitions or slices (Linux and Solaris systems)
- ◆ Conversion of LVM disks (AIX, HP-UX and Linux systems)

You can override this behavior by changing the settings in the system defaults files as described in “[Defaults Files](#)” on page 17. For example, you can change the default format to `sliced` for disk initialization by modifying the definition of the `format` attribute in the `/etc/default/vxdisk` defaults file. To change the default format for disk encapsulation or LVM disk conversion, change the definition of the `format` attribute in the `/etc/default/vxencap` defaults file.

### Restoring CDS Disk Labels

CDS disks have three labels:

- ◆ Platform block
- ◆ AIX coexistence label
- ◆ HP coexistence/VxVM ID block

There are also backup copies of each. If any of the primary labels become corrupted, VxVM will not bring the disk online and user intervention is required.



If two of the three labels are intact, the disk is still recognized as a `cdsdisk` (though in the 'error' state) and `vxdisk flush` can be used to restore the CDS disk labels from their backup copies.

Primary labels are at sectors 0, 7, and 16; and a normal flush will not flush sectors 7 and 16. Also, the private area is not updated as the disk is not in a disk group. There is no means of finding a “good” private region to flush from. In this case, it is possible to restore the CDS disk labels from the existing backups on disk using the flush operation.

If a corruption happened after the labels were read and the disk is still online and part of a disk group, then a flush operation will also flush the private region.

---

**Caution** Caution and knowledge must be employed because the damage could involve more than the CDS disk labels. If the damage is constrained to the first 128K (as would be the case if some entity on the fabric - such as a Windows box - wrote a disk label to a disk which was actually a `cdsdisk` being used in some disk group), then the disk flush would fix it.

---

Use the following command to rewrite the CDS ID information to a specific disk (this rewrites all labels except sectors 7 and 16):

```
# vxdisk flush disk_accessname
```

Use the following command to rewrite all the disks in a CDS disk group (this rewrites all labels except sectors 7 and 16):

```
# vxdg flush dg_name
```

Use the `-f` option with the `vxdisk` command to forcibly rewrite the AIX coexistence label (sector 7), VxVM ID block (sector 16), and HP-UX Coexistence Labels:

```
# vxdisk -f flush disk_accessname
```

This command rewrites all labels if there exists a valid VxVM ID block that points to a valid private region.

---

**Note** Sectors 7 and 16 are only rewritten if the `-f` flag is given. The reason for this is that these sectors lay within the first track of the disk, and Windows systems use the first track for their own purposes. In the example above where the disk was taken offline due to label corruption, if it involved sectors 7 or 16, then `-f` is required to fix the problem.

---



## Disk Group Tasks

The following disk group tasks are supported:

- ◆ [Changing the Alignment of a Disk Group by Encapsulation](#)
- ◆ [Object Alignment During Volume Creation](#)
- ◆ [Changing the Alignment of a Non-CDS Disk Group](#)
- ◆ [Determining the Setting of the CDS Attribute](#)
- ◆ [Joining Disk Groups](#)
- ◆ [Moving Objects Between CDS and Non-CDS Disk Groups](#)
- ◆ [Changing Default CDS Setting for Disk Group Creation](#)
- ◆ [Creating Non-CDS Disk Groups](#)
- ◆ [Upgrading an Older Version Non-CDS Disk Group](#)
- ◆ [Replacing a Disk in a CDS Disk Group](#)
- ◆ [Setting Device Quotas for CDS Disk Groups](#)

### Changing the Alignment of a Disk Group by Encapsulation

If you use the `vxdiskadm` command to encapsulate a disk into a disk group with an alignment of 8K, the disk group alignment must be reduced to 1. As part of the encapsulation process, you are asked to confirm that this reduction of the disk group alignment is acceptable.

---

**Note** If you are using the `vxencap` command to perform the encapsulation, the alignment is carried out automatically without a confirmation prompt.

---

### Object Alignment During Volume Creation

For CDS disk groups, VxVM objects that are used in volume creation are automatically aligned to 8K. For non-CDS disk groups, the `vxassist` attribute, `dgalign_checking`, controls how the command handles attributes that are subject to disk group alignment restrictions. If set to `strict`, the volume length and values of attributes must be integer multiples of the disk group alignment value, or the command fails and an error message is displayed. If set to `round` (default), attribute values are rounded up as required. If this attribute is not specified on the command-line or in a defaults file, the default value of `round` is used.

The `diskalign` and `nodiskalign` attributes of `vxassist`, which control whether subdisks are aligned on cylinder boundaries, is honored only for non-CDS disk groups whose alignment value is set to 1.

## Changing the Alignment of a Non-CDS Disk Group

Use the alignment attribute (`align`) with the `vxdbg set` command to change the alignment value of a version 110 disk group to 1 or to 8KB, as shown here:

```
# vxdbg -g diskgroup set align=1
# vxdbg -g diskgroup set align=8k
```

---

**Note** For a CDS disk group, `alignment` can only take a value of 8k. Attempts to set the alignment of a CDS disk group to 1 fail unless you first change it to a non-CDS disk group.

Increasing the alignment may require `vxcdsconvert` to be run to change the layout of the objects in the disk group.

To display the current alignment value, use the `vxprint` command as described in [“Displaying the Disk Group Alignment”](#) on page 31.

---

The operation to increase the alignment to 8K fails if objects exist in the disk group that do not conform to the new alignment restrictions. In that case, use the `vxcdsconvert alignment` command to change the layout of the objects as described in [“Converting a Non-CDS Disk Group to a CDS Disk Group”](#) on page 15:

```
# vxcdsconvert -g diskgroup [-A] [-d defaults_file] \
  [-o novolstop] alignment [attribute=value] ...
```

This command increases the alignment value of a disk group and its objects to 8K, without converting the disks.

The sequence 8K -> 1 -> 8K is possible only using `vxdbg set` as long as the configuration does not change after the 8K -> 1 transition.



## Determining the Setting of the CDS Attribute

You can use the `vxdbg list` or `vxprint` commands to determine the state of the CDS attribute, as follows:

```
# vxdbg list
NAME          STATE          ID
dgTestSol2    enabled,cds    1063238039.206.vmescl

# vxdbg list dgTestSol2
Group:        dgTestSol2
dgid:         1063238039.206.vmescl
import-id:    1024.205
flags:        cds
version:      110
alignment:    8192 (bytes)
...

# vxprint -F %cds -G -g dgTestSol2
on
```

In these examples, the disk group, `dgTestSol2`, is shown as having the CDS flag set.

## Joining Disk Groups

Use the `vxdbg join` command to join two disk groups. Joining two CDS disk groups or joining two non-CDS disk groups is permitted, but you cannot join a CDS disk group to a non-CDS disk group. If two non-CDS disk groups have different alignment values, the alignment of the resulting joined disk group is set to 1, and an informational message is displayed.

Refer to the *VERITAS Volume Manager Administrator's Guide* for examples of using this command.

## Moving Objects Between CDS and Non-CDS Disk Groups

Use the `vxdbg move` command to move objects between CDS and non-CDS disk groups. The alignment of a source non-CDS disk group must be 8K to allow objects to be moved to a target CDS disk group. If objects are moved from a CDS disk group to a target non-CDS disk group with an alignment of 1, the alignment of the target disk group remains unchanged.

## Moving Objects Between CDS Disk Groups

Use the `vx dg move` command to move objects between CDS disk groups. The disk group alignment does not change.

## Changing Default CDS Setting for Disk Group Creation

You can change the default CDS attribute setting used in creating disk groups by modifying the `/etc/default/vxdg` file.

## Creating Non-CDS Disk Groups

Use the `vx dg` command to create a non-CDS pre-version 110 disk group:

```
# vx dg -T vers init dg disk_name=device_name
```

A pre-version 110 DG is given an alignment value equal to 1.

---

**Note** For a pre-version 110 DG, since the `dg_align` value is not stored in the configuration database, it is set to 1 when the disk group is imported.

---

## Upgrading an Older Version Non-CDS Disk Group

You might want to upgrade a non-CDS pre-version 110 disk group in order to use new features (excluding CDS).

---

**Note** You must always perform a disk group conversion (using the `vx cdsconvert` utility) to use the CDS feature.

---

Use the `vx dg` command to upgrade the non-CDS pre-version 110 disk group:

```
# vx dg upgrade dg
```

---

**Note** After upgrading, the `cds` attribute is set to `off`, and the disk group retains an alignment of 1.

---



## Replacing a Disk in a CDS Disk Group

To replace a disk in a CDS disk group, use the command sequence:

```
# vxdg -k rmdisk disk_name
# vxdg -k adddisk
```

Refer to the *VERITAS Volume Manager Administrator's Guide* for an explanation of the `-k` option.

For example:

```
# vxdg -k rmdisk disk1
# vxdg -k adddisk disk1=c1t0d1
```

where `c1t0d1` is the device name of a Solaris disk that is currently being reassigned to disk `disk1`. For other operating systems, use the appropriate device name format, for example, `hdisk#` on AIX, and `sdX` on Linux.

---

**Note** When replacing a disk in a CDS disk group, you cannot use a non-CDS disk as the replacement.

---

## Setting Device Quotas for CDS Disk Groups

Use the `vxvg set` command to limit the number of devices that can be created in a given CDS disk group:

```
# vxvg set maxdev=max-devices
```

`maxdev` can have any positive integer value greater than the number of devices currently in the disk group. For example:

```
# vxvg -g dg set maxdev=1000
```

## Setting DRL Map Size and Log Size

If DRL is enabled on a newly created volume without specifying a log or map size, default values are used. You can use the command line attributes `logmap_len` and `loglen` in conjunction with the `vxassist`, `vxvol`, and `vxmake` commands to set the DRL map and DRL log sizes. The attributes can be used independently, or they can be combined.

You can change the DRL map size and DRL log size only when the volume is disabled and the DRL maps are not in use. Changes can be made to the DRL map size only for volumes in a CDS disk group.

The `logmap_len` attribute specifies the required size, in bytes, for the DRL log. It cannot be greater than the number of bytes available in the map on the disk.

If you need to change the size after creating, you need to remove and rebuild the logs, using the following commands:

```
# vxassist -g diskgroup remove log volume nlog=0
# vxassist -g diskgroup addlog volume nlog=nlogs logtype=drl \
  logmap_len=len-bytes [drllength=len]
```

Note the following restrictions:

- ◆ If only `logmap_len` is specified, the DRL log size is set to the default value (33\*DG alignment).
- ◆ If `logmap_len` is greater than  $(DRL \ log \ size)/2$ , the command fails, and you need to either provide a sufficiently large `loglen` value or reduce `logmap_len`.
- ◆ For CDS disk groups, the DRL map and log sizes are set to a minimum of 2\*(DG alignment).

## Creating a DRL Log

You can create a DRL log using one of the following commands:

- ◆ `vxassist`
- ◆ `vxvol`

### Using vxassist to Create a DRL Log

When using the `vxassist` command to create a DRL log, it creates logging subdisks equal to the size of the DRL log.

```
# vxassist -g dgTestSol2 make drlvol 100m mirror=2 logtype=drl \
  loglen=264k logmap_len=2048
```

If neither `logmap_len` nor `loglen` is specified, then:

- ◆ `loglen` is set to a default value based on disk group alignment.
- ◆ `maplen` is set to a reasonable value.

If only `loglen` is specified, then:

- ◆ For pre-version 110 disk groups, `maplen` is set to zero.
- ◆ For version 110 and greater disk groups, `maplen` set to use all the bytes available in on-disk map.

If only `logmap_len` is specified (this applies only to disk groups with a version of 110 or greater), `maplen` must be less than number of available bytes in on-disk map for the default log length.



## Using vxvol to Set a DRL Map Length

You can use `vxvol` only if the volume is stopped (that is, the DRL is inactive).

```
# vxvol -g dgTestSol2 set logmap_len=512 drlvol
```

The `vxvol set` command does not change the existing DRL map size.

---

**Note** When specifying a `loglen` attribute, specifying a value less than the minimum required (twice the disk group alignment value) results in an error message.

---

The value of `loglen` is constrained by size of the logging subdisk.

The value of `logmap_len` is constrained by the log length.

If both `logmap_len` and `loglen` are specified, then, if `logmap_len` is greater than `loglen/2`, `vxvol` fails with an error message, and you then have to either increase `loglen` to a sufficiently large value or decrease `logmap_len` to a sufficiently small value.

The value of `logmap_len` is in units of bytes, and the value of `loglen` is in units of blocks. `logmap_len` cannot exceed the number of bytes in the on-disk map. The number of bytes in the on-disk map can be calculated as follows:

$$[\text{ROUND\_DOWN}(\text{loglen}/n\text{maps}) - \text{VOLDRL\_HEADER\_SIZE}]$$

where `nmaps` is 2 for a private disk group or 33 for a shared disk group, `ROUND_DOWN()` represents rounding down to a log map alignment boundary, and the value of `VOLDRL_HEADER_SIZE` is 24.

## Displaying Information

This section describes:

- ◆ [Displaying Traditional DRL Map Size and Log Size](#)
- ◆ [Displaying the Disk Group Alignment](#)
- ◆ [Displaying Volume Log Map Values](#)
- ◆ [Listing Offset and Length Information](#)
- ◆ [Listing CDS Disk Groups](#)
- ◆ [Listing Disks](#)
- ◆ [Displaying Device Quotas for CDS Disk Groups](#)



## Displaying Traditional DRL Map Size and Log Size

Use the `vxprint` command to display the map length and map alignment of traditional DRL logs:

```
# vxprint -g dg1 -vl drlvols
# vxprint -g dg1 -vF '%name %logmap_len %logmap_align' drlvols
```

## Displaying the Disk Group Alignment

To discover the value in blocks of the alignment that is set on a disk group, use this command:

```
# vxprint -g diskgroup -G -F %align
```

Utilities that print information regarding a disk group record (such as, `vxprint` and `vxdbg list`) can also output the disk group alignment. For example, to print information for disk group `dg1`, you can use the command:

```
# vxdbg list dg1
```

## Displaying Volume Log Map Values

Use the `vxprint` command to determine the log map alignment and log map length values:

```
# vxprint [ -g diskgroup ] -lv volname
```

For example, to print information for the volume `vol1` in disk group `dg1`, you can use the command:

```
# vxprint -g dg1 -lv vol1
```

The output is of the form:

```
logging: type=REGION loglen=0 serial=0/0 mapalign=0 maplen=0
(disabled)
```

This indicates a log map alignment (`logmap_align`) value of 0, and a log map length (`logmap_len`) value of 0.

If the log map is set and enabled, the command and results may be of the form:

```
# vxprint -lv drlvol
Disk group: dgTestSol

Volume:    drlvol
info:      len=20480
type:      usetype=fsngen
state:     state=ACTIVE kernel=ENABLED cdsrecovery=0/0 (clean)
```



```
assoc:      plexes=drlvol-01,drlvol-02,drlvol-03
policies:   read=SELECT (round-robin) exceptions=GEN_DET_SPARSE
flags:      closed writecopy writeback
logging:    type=REGION loglen=528 serial=0/0 mapalign=16 maplen=512
            (enabled)
apprecov:   seqno=0/0
recovery:   mode=default
recov_id=0
device:     minor=46000 bdev=212/46000 cdev=212/46000
path=/dev/vx/dsk/dgTestSol/drlvol
perms:      user=root group=root mode=0600
guid: {d968de3e-1dd1-11b2-8fc1-080020d223e5}
```

## Listing Offset and Length Information

Use the `vxprint` command with the `-b` option to output all offset and length information in units of 512 bytes. This enables consistent output from different platforms. Without the `-b` option, this information is output with a value of sectors that may differ between platforms.

When `vxprint -bm` is used, the output also contains the `b` suffix, so that the output can be fed back to `vxmake`.

## Listing CDS Disk Groups

You can use the following commands to list all disk groups; then look for flag `cds` in the listing to locate CDS disk groups.

```
# vx dg list
# vxprint -G1
```

## Listing Disks

Use the `vxdisk list` command with the `-b` option to output all offset and length information in units of 512 bytes. This enables consistent output from different platforms. Without the `-b` option, this information is output with a value of sectors that may differ between platforms.

When `vxdisk -b list` is used, the output also contains the `b` suffix, so that the output can be fed back to `vxmake`.

The disk format is put in the `TYPE` column along with the `auto` type.

## Displaying Device Quotas for CDS Disk Groups

Use the `vxprint -lG` command to display the maximum number of devices in a CDS disk group. For example, to display the maximum number of devices in the CDS disk group `cdsdg1`, you can use a command of the form:

```
# vxprint -lG cdsdg1
```

The output is similar to the following:

```
group:  dg1
info:    dgid=1027365608.1064.vmes6
version: 110
alignment: 8192 (bytes)
activation: read-write
detach-policy: global
copies:  nconfig=default nlog=default
devices: max=32000 cur=0
minors:  >= 16000
cds=on
```

The value of `max` on the `devices` line indicates the maximum number of devices. In this example, the maximum number of devices is 32,000.

You can also use the following command to obtain the maximum number of devices:

```
# vxprint -g dg1 -GF %maxdev
32000
```

## Default Activation of Shared Disk Groups

Default activation of shared disk groups involves a local in-kernel policy that differs between platforms. This means that, regardless of the platform on which the disk group was created, the importing platform will have platform-specific behavior with respect to activation of shared disk groups. Specifically, with the exception of HP-UX, importing a shared disk group results in the volumes being active and enabled for shared-write. In the case of HP-UX, the shared volumes will be inactive and require other actions to activate them for shared-write operations.

## Creating CDS Disk Groups Using `vx dg split`

You can use the `vx dg split` command to create a CDS disk group from an existing CDS disk group. The new (target) and original (source) disk group retain the same CDS attribute; that is, if the source is CDS, then the new disk group is also CDS. The new disk group also has the same alignment as the original disk group.



## Additional Considerations When Importing CDS Disk Groups

Before attempting to use CDS to move disk groups between different operating systems, there are several points that you should consider if the configuration of the disks has changed since the target system was last rebooted:

- ◆ Does the target system know about the disks? For example, the disks may not have been connected to the system either physically (not cabled) or logically (using FC zoning or LUN masking) when the system was booted up, but they have subsequently been connected without rebooting the system. This can happen when bringing new storage on-line, or when adding an additional DMP path to existing storage. On the target system, both the operating system and VxVM must be informed of the existence of the new storage. On Linux, depending on the supported capabilities of the host adapter, you may need to reboot the target system to achieve this. For other operating systems, issue the appropriate command to tell the operating system to look for the storage. Having done this, run either of the following commands on the target system to have VxVM recognize the storage:

```
# vxctl enable
# vxdisk scandisks
```

- ◆ Do the disks contain partitions or slices? Both Solaris and Linux systems maintain information about partitions or slices on disks. If you repartition a disk after the target system was booted, use the appropriate command to instruct the operating system to rescan the disk's TOC or partition table. For example, on a target Linux system, use the following command:

```
# blockdev --rereadpt
```

Having done this, run either of the following commands on the target system to have VxVM recognize the storage:

```
# vxctl enable
# vxdisk scandisks
```

- ◆ Has the format of any of the disks changed since the target system was last booted? For example, if you use the `vxdisksetup -i` command to format a disk for VxVM on one system, the `vxdisk list` command on the target system may still show the format as being `auto:none`. If so, use either of the following commands on the target system to instruct VxVM to rescan the format of the disks:

```
# vxctl enable
# vxdisk scandisks
```

## File System Considerations

---

This chapter describes VERITAS File System (VxFS) support for CDS:

- ◆ [Considerations About Data in the File System](#)
- ◆ [File System Migration](#)
- ◆ [Using the fscdstask Script](#)
- ◆ [One-Time Migration of a File System](#)
- ◆ [Ongoing Migration of a File System](#)
- ◆ [When to Convert a File System](#)
- ◆ [Converting a File System](#)

---

**Note** VERITAS File System 4.1 for Solaris x86 Platform Edition does not support CDS at the file system level.

---



## Considerations About Data in the File System

Data within a file system might not be in the appropriate format to be accessed if moved between different types of systems. For example, files stored in proprietary binary formats often cannot be transferred as is. Files containing databases might not be in a standard format that allows their access when moving a file system between various systems, even if those systems use the same byte order.

Some data is inherently portable, such as plain ASCII files. Other data is designed to be portable and the applications that access such data are able to access it irrespective of the system on which it was created, such as Adobe PDF files.

Note that the CDS facilities do not convert the end user data. The data is uninterpreted by the file system. Only individual applications have knowledge of the data formats, and thus those applications and end users must deal with this issue. This issue is not CDS-specific, but is true whenever data is moved between different types of systems.

Even though a user might have a file system with data that cannot be readily interpreted or manipulated on a different type of system, there still are reasons for moving such data by using CDS mechanisms. For example, if the desire is to bring a file system off line from its primary use location for purposes of backing it up without placing that load on the server or because the system on which it will be backed up is the one that has the tape devices directly attached to it, then using CDS to move the file system is appropriate.

Another example is a principal file server that has various file systems being served by it over the network. If a second file server system with a different operating system was purchased to reduce the load on the original server, CDS can migrate the file system instead of having to move the data to different physical storage over the network, even if the data could not be interpreted or used by either the original or new file server. This is a scenario that often occurs when the data is only accessible or understood by software running on PCs and the file server is UNIX or Linux-based.

## File System Migration

*File system migration* refers to the system management operations related to stopping access to a file system, and then restarting these operations to access the file system from a different computer system. File system migration might be required to be done once, such as when permanently migrating a file system to another system without any future desire to move the file system back to its original system or to other systems. This type of file system migration is referred to as *one-time file system migration*. When ongoing file system migration between multiple systems is desired, this is known as *ongoing file system migration*. Different actions are required depending on the kind of migration, as described in the following sections.

## Using the fscdstask Script

The `fscdstask` script performs several CDS tasks as specified by the use of keywords, summarized below.

---

**Note** The file system migration sections below indicate when each command should be used.

---

To ensure that there are no file system entities with metadata that exceed the limits for a set of operating systems, enter:

```
# fscdstask validate OS_list mount_point
```

When a file system will be migrated on an ongoing basis between multiple systems, the types of systems involved in these migrations is maintained in an `OS_list` file. Knowing what these operating systems are allows VxFS to determine file system limits that are appropriate to all of these operating systems. This `OS_list` is manipulated by several `fscdstask` keywords.

To add to the list of operating systems between which the file system will be moved on an ongoing basis, enter:

```
# fscdstask os add os mount_point
```

To remove from the list of operating systems between which the file system will be moved on an ongoing basis, enter:

```
# fscdstask os remove os mount_point
```

For both the `add` and `remove` keywords, valid values for `os` are AIX, HP-UX, Linux or SunOS.

To clear the list of operating systems between which the file system will be moved, enter:

```
# fscdstask os none mount_point
```

To list the operating systems between which the file system will be moved, enter:

```
# fscdstask os list mount_point
```

The limits implied by `OS_list` are by default ignored. To have the file system enforce or ignore these limits, use the following commands.

To enforce established CDS limits on a file system, enter:

```
# fscdstask limit enforce mount_point
```

To ignore established CDS limits on a file system, enter:

```
# fscdstask limit ignore mount_point
```



Various CDS statuses are maintained on a per-file system basis, including: the `OS_list`, the limits implied by the `OS_list`, and whether the limits are being enforced or ignored.

- ◆ To display the current CDS status of a file system, enter:

```
# fscdstask status mount_point
```

- ◆ For additional help on any of these keywords, enter:

```
# fscdstask help keyword
```

## One-Time Migration of a File System

The following example describes a one-time migration of data from the Solaris OS to Linux. Some of the following steps require a backup of the file system to be created. To simplify the process, you can create one backup before performing any of the steps instead of creating multiple backups as you go.

1. If the underlying Volume Manager storage is not contained in a CDS disk group, it must first be upgraded to be a CDS disk group, and all other physical considerations related to migrating the storage physically between systems must first be addressed as described in [“Setting up Your System”](#) on page 11.
2. If the Solaris file system is using disk layout Version 5 or prior, upgrade the file system to Version 6. See the *Installation Guide* for information on upgrading the file system to disk layout Version 6.
3. Ensure that there are no files in the file system that will be inaccessible after migrating the data due to large file size or user/group id platform differences:

```
# fscdstask validate OS mount_point
```

If such files exist, move the files to another file system or reduce the files' sizes.

---

**Note** There is no output if no limits are exceeded.

---

4. Unmount the file system:

```
# umount mount_point
```

5. Use the `fscdsconv` command to convert the file system to the opposite endian, if required. See [“Converting a File System”](#) on page 41.



6. Make the physical storage and Volume Manager logical storage accessible on the Linux system by exporting the disk group from the source system and importing the disk group on the target system after resolving any other physical storage attachment issues. See “[Maintaining Your System](#)” on page 21 for more information.
7. Mount the file system on the target system.

## Ongoing Migration of a File System

The following example describes how to migrate a file system between Linux and the Solaris OS on an ongoing basis. Some of the following steps require a backup of the file system to be created. To simplify the process, you can create one backup before performing any of the steps instead of creating multiple backups as you go.

1. Ensure that there are no files in the file system that will be inaccessible after migrating the data due to large file size or user/group id platform differences:

```
# fscdstask validate OS mount_point
```

If such files exist, move the files to another file system or reduce the files’ sizes.

---

**Note** There is no output if no limits are exceeded.

---

2. Add SunOS and Linux to the `OS_list` file:

```
# fscdstask os add SunOS /mnt1
# fscdstask os add Linux /mnt1
```

3. Enforce the limits:

```
# fscdstask limit enforce mount_point
```

This is the last of the preparation steps. When the file system is to be migrated, it must be unmounted, and then the storage moved and mounted on the target system:

4. Unmount the file system:

```
# umount mount_point
```



5. Use the `fscdsconv` command to convert the file system to the opposite endian, if required. See “[Converting a File System](#)” on page 41.
6. Make the physical storage and Volume Manager logical storage accessible on the Linux system by exporting the disk group from the source system and importing the disk group on the target system after resolving any other physical storage attachment issues. See “[Maintaining Your System](#)” on page 21 for more information.
7. Mount the file system on the target system.

## Ceasing Ongoing Migration

If in the future you want to stop performing ongoing migrations and want to leave the file system on the current system, the following will stop the usage of CDS mechanisms:

```
# fscdstask limit ignore mount_point
# fscdstask os none mount_point
```

## When to Convert a File System

When moving a file system between different platforms, you might be required to convert the byte order of the file system. Consider the following information to determine if you must convert a file system:

- ◆ You must convert the file system if you are moving the file system between platforms that have different byte orders (little endian versus big endian).
- ◆ Conversion is not required when moving a file system between platforms that have the same byte order.
- ◆ At present, VxFS for the AIX, HP-UX and Solaris OS platforms is available only on big endian systems, while VxFS for Linux is available only on little endian systems (x86). Therefore, byte order conversion is generally only an issue between Linux and these other operating systems. However, some VERITAS OEMs use VxFS on the x86 version of the Solaris OS, and therefore have little endian file systems. In such a case, conversion is required between the Solaris OS on x86 and AIX, HP-UX and the Solaris OS on SPARC, but not between the Solaris OS on x86 and Linux.

---

**Note** VERITAS has never supported moving file systems between different platforms, although in many cases a user might successfully do so and be able to access the data without trouble. You cannot move file systems of disk layouts earlier than Version 6 between different platforms. Doing so can result in data loss or data corruption.

---

# Converting a File System

Use the `fscdsconv` command to convert the byte order of a file system.

1. Determine if you must convert the file system (see “[When to Convert a File System](#)” on page 40).
2. Determine the disk layout version of the file system that you must convert.

```
# fstyp -v /dev/vx/dsk/filesystem | grep version  
magic a501fcf5  version 6  ctime Tue Jul 22 16:16:53 2003
```

Only file systems with Version 6 disk layout can be converted. If the file system has an earlier disk layout version, convert the file system to Version 6 disk layout before proceeding (see the `vxfsconvert(1M)` and `vxupgrade(1M)` manual pages).

3. Perform a full file system back up. Failure to do so could result in data loss or data corruption under some failure scenarios in which restoring from the backup is required.
4. Designate a file system with free space where `fscdsconv` may create a file that will contain recovery information for usage in the event of a failed conversion. Depending on the nature of the file system to be converted, for example if it is mirrored, you may wish to designate the recovery file to reside in a file system with the same level of failure tolerance so as to reduce the number of failure scenarios that would require the use of the backup created in [step 3](#).
5. Unmount the file system to be converted.

```
# umount mount_point
```



6. Use the `fscdsconv` command to convert the file system.

```
# fscdsconv -f recovery_file special
```

*recovery\_file* is the name of the recovery file to be created by the `fscdsconv` command. *special* is the raw device or volume that contains the file system to be converted. Include the file system chosen in [step 4](#) when designating the recovery file. For example, if the file system chosen to contain the recovery file is mounted on `/data/fs3`, the recovery file could be specified as `/data/fs3/jan04recovery`. If there is not enough disk space on the chosen file system for the recovery file to be created, the conversion aborts and the file system to be converted is left intact.

---

**Note** The recovery file is not only used for recovery purposes after a failure, but is also used to perform the conversion. The directory that will contain the recovery file should not allow non-system administrator users to remove or replace the file, as this could lead to data loss or security breaches. The file should be located in a directory that is not subject to system or local scripts that might cause the file's removal after a system reboot, such as that which occurs with the `/tmp` and `/var/tmp` directories on the Solaris OS.

---

If you are converting multiple file systems at the same time, which requires the use of one recovery file per file system, record the names of the recovery files and their corresponding file systems being converted in the event that recovery from failures is required at a later time.

7. After some time, `fscdsconv` creates the recovery file and prompts for confirmation to proceed with the conversion.

```
Do you wish to commit to conversion? (y/n)
```

- a. Inputting **n** causes `fscdsconv` to remove the recovery file and leave the file system intact.
- b. Inputting **y** causes `fscdsconv` to proceed with the conversion. No other interactions are required. The recovery file is not removed after the conversion completes. The recovery file can be used to restore the file system to its original state. Some situations in which the recovery file can be used are if the conversion is done accidentally or if no conversion is necessary. For example, migrating a file system from the Solaris operating system to the AIX operating system does not require a conversion because both operating systems use the big endian format.

8. If a failure occurs during the conversion, the failure could be one of the following cases:

- ◆ System failure.
- ◆ `fscdsconv` failure due to program defect or abnormal termination resulting from user actions.

In either case, the file system being converted is no longer in a state in which it can be mounted or accessed by normal means through other VxFS utilities. To recover the file system, invoke the `fscdsconv` command with the recovery flag, `-r`:

```
# fscdsconv -r -f recovery_file special
```

When the `-r` flag is specified, `fscdsconv` expects the recovery file to exist and that the file system being converted is the same file system specified in this second invocation of `fscdsconv`.

9. After invoking `fscdsconv` with the `-r` flag, the conversion process will restart and complete, given no subsequent failures. In the event of another failure, repeat [step 8](#).

Under some circumstances, you will be required to restore the file system from the backup created in [step 3](#), such as if the disk fails that contains the recovery file. Failure to have created a backup would then result in total data loss in the file system. I/O errors on the device that holds the file system would also require a backup to be restored after the physical device problems are addressed. There may be other causes of failure that would require the use of the backup.





# Cross-Platform Transfer

---

**A**

This appendix contains notes on considerations for data transfer between operating system platforms with different inherent characteristics.

## Alignment Value and Block Size

On the AIX, Linux and Solaris operating systems, an alignment value of 1 is equivalent to a block size of 512 bytes. On the HP-UX operating system, it is equivalent to a block size of 1024 bytes.

The block size on HP-UX is different from that on other supported platforms. Output from commands such as `vxdisk` and `vxprint` looks different on HP-UX for the same disk group if the `-b` option is not specified.

## Default Activation of Shared Disk Groups

This is a local in-kernel policy that differs between platforms. This means that, regardless of the platform on which the disk group was created, the importing platform will have platform-specific behavior with respect to activation of shared disk groups. Specifically, with the exception of HP-UX, importing a shared disk group will result in the volumes being active and enabled for shared-write. In the case of HP-UX, the shared volumes will be inactive and require other actions to activate them for shared-write operations.

## Disk Group Alignment and Encapsulated Disks

On the Solaris OS, all native file systems are cylinder aligned. Encapsulating such a disk results in subdisks that are also cylinder aligned. Such alignment will normally not be 8K aligned, but it will be 1K aligned. For the encapsulation process, there is no flexibility as to where on the disk the subdisks must be since the data location is predefined. If an alignment conflict occurs, user intervention is required. If the disk group alignment is 8K this operation will probably fail because this would require the cylinder to be an even number of 8K blocks in size.



## Importing Disk Groups Between Linux and Non-Linux Machines

A disk group created on non-Linux platforms typically has device numbers above 1000. When that disk group is imported on a Linux machine with a pre-2.6 kernel, the devices are reassigned minor numbers below 256.

If this disk group is now imported to a non-Linux machine, all device numbers will be less than 256. If those devices are available (that is, they do not conflict with devices in an imported boot disk group) they will be used. Otherwise new device numbers will be reassigned.

A single disk group could contain a number of devices exceeding the maximum number of devices for a given platform. In this case, the disk group cannot be imported on that platform because import would exhaust available minor devices for the VxVM driver. Although the case of minor number exhaustion is possible in a homogeneous environment, it will be more pronounced between platforms with different values for the maximum number of devices supported, such as Linux with a pre-2.6 kernel. This difference will render platforms with low maximum devices supported values less useful as heterogeneous disk group failover or recovery candidates.

---

**Note** Using the disk group `maxdev` attribute may reduce the likelihood that a CDS disk group import on Linux with a pre-2.6 kernel will exceed the maximum number of devices.

---



# Error Codes and Error Recovery

## Error Codes and Required Actions

Error Number	Message	Action
329	Cannot join a non-CDS disk group and a CDS disk group	Change the non-CDS disk group into a CDS disk group (or vice versa), then retry the join operation.
330	Disk group is for a different platform	Import the disk group on the correct platform. It cannot be imported on this platform.
331	Volume has a log which is not CDS compatible	To get a log which is CDS compatible; you need to stop the volume, if currently active, then start the volume. After the volume has been successfully started, retry setting the CDS attribute for the disk group.
332	License has expired, or is not available for CDS	Obtain a license from VERITAS that enables the usage of CDS disk groups.
333	Non-CDS disk cannot be placed in a CDS disk group	Do one of the following: <ul style="list-style-type: none"><li>♦ Add the disk to another disk group that is a non-CDS disk group.</li><li>♦ Re-initialize the disk as a CDS disk so that it can be added to the CDS disk group.</li><li>♦ Change the CDS disk group into a non-CDS disk group and then add the disk.</li></ul>
334	Disk group alignment not CDS compatible	Change the alignment of the disk group to 8K and then retry setting the CDS attribute for the disk group.



---

## Error Codes and Required Actions

---

Error Number	Message	Action
335	Subdisk length violates disk group alignment	Ensure that sub-disk length value is a multiple of 8K.
336	Subdisk offset violates disk group alignment	Ensure that sub-disk offset value is a multiple of 8K.
337	Subdisk plex offset violates disk group alignment	Ensure that sub-disk plex offset value is a multiple of 8K.
338	Plex stripe width violates disk group alignment	Ensure that plex stripe width value is a multiple of 8K.
339	Volume or log length violates disk group alignment	<p>Ensure that volume length value is a multiple of 8K.</p> <p>For volume log length; if you are using <code>vxassist</code> to create the volume, then you should set <code>dgalign_checking</code> to <code>round</code>. This will ensure that the log length is silently rounded to a valid value.</p>
340	Last disk media offset violates disk group alignment	Reassociate the DM record prior to upgrading.
341	Too many device nodes in disk group	Increase the number of device nodes allowed in the disk group, if not already at the maximum. Otherwise, you need to remove volumes from the disk group, possibly by splitting the disk group.
342	Map length too large for current log length	Use a smaller map length for the DRL/DCM log, or increase the log length and retry.
343	Volume log map alignment violates disk group alignment	Remove the DRL/DCM log, then add it back after changing the alignment of the disk group.
345	Disk group contains an old-style RVG which cannot be imported on this platform	Import the disk group on the platform that created the RVG. To import the disk group on this platform, first remove the RVG on the creating platform.
346	Cache object autogrow by <code>max_autogrow</code> violates disk group alignment	Ensure that cache attribute value is a multiple of 8K.



---

## Error Codes and Required Actions

---

Error Number	Message	Action
347	User transactions are disabled for the disk group	Retry the command as it was temporarily disallowed by the <code>vxcdsconvert</code> command executing at the same time.
348	Disk is in use	Contact VERITAS Technical Support.

---





# Glossary

---

## **AIX coexistence label**

Data on disk which identifies the disk to the AIX volume manager (LVM) as being controlled by VxVM. The contents has no relation to VxVM ID Blocks.

## **back-rev disk group**

A disk group created using a version of VxVM released prior to the release of CDS. Adding CDS functionality rolls over to the latest disk group version number; see also current-rev disk group.

## **CDS (Cross-platform Data Sharing)**

Sharing data between heterogeneous systems (such as Solaris and HP-UX operating systems), where each system has direct access to the physical devices used to hold the data, and understands the data on the physical device. Sharing in this sense should not be confused with the sharing provided with CVM by means of a shared disk group.

## **CDS disk**

A disk whose contents and attributes are such that the disk can be used for CDS as part of a CDS disk group. In contrast, a non-CDS disk cannot be used for CDS, nor can it be part of a CDS disk group. CDS disk also contains a set of AIX Coexistence Labels, HP-UX Coexistence Labels/VxVM ID Blocks, and Platform Blocks.

## **CDS disk group**

A VxVM disk group whose contents and attributes are such that the disk group can be used to provide CDS. In contrast, a non-CDS disk group (that is, a back-rev disk group or a current-rev disk group) cannot be used for CDS. A CDS disk group is a current-rev disk group with the CDS attribute set for the disk group. A CDS disk group can only contain CDS disks.

## **CFS**

Cluster file system. A VxFS file system mounted on a selected volume in cluster (*shared*) mode.



---

**children**

Objects that belong to an object group.

**cluster**

A set of host machines (*nodes*) that shares a set of disks.

**cluster file system**

See CFS.

**current-rev disk group**

A disk group created using a version of VxVM providing CDS functionality; however, the CDS attribute is not set. If the CDS attribute is set for the disk group, the disk group is called a CDS disk group.

**data change object**

See DCO.

**DCO (Data Change Object)**

A VxVM object that is used to manage information about the FastResync maps in the DCO volume. Both a DCO object and a DCO volume must be associated with a volume to implement Persistent FastResync on that volume.

**DCO volume**

A special volume that is used to hold Persistent FastResync change maps, and dirty region logs (see [dirty region logging](#)). The map layout within the DCO volume changed with the release of VxVM 4.0, although the original format is still available. The old layout is available in DCO Version 0 objects, and the new layout is available in DCO Version 20 objects.

**device name**

The physical disk device name (or *disk access name*).

**dirty region logging**

See DRL.

---

**disk access name**

The device name or address that is used to access a physical disk on an operating system, such as `hdisk1` (AIX), `c0t0d0` (HP-UX), `sda` (Linux), or `c0t0d0s2` (Solaris OS). In a SAN environment, it is more convenient to use *enclosure-based naming*, which forms the device name by concatenating the name of the enclosure (such as `enc0`) with the disk's number within the enclosure, separated by an underscore (for example, `enc0_2`).

**disk group**

A set of disks that are under VxVM control and share a common configuration. A disk group configuration is a set of records containing detailed information on existing VERITAS Volume Manager objects (such as disk and volume attributes) and their relationships. Each disk group has an administrator-assigned name. Volumes can only be created on disks that belong to disk groups.

**disk media name**

A logical or administrative name chosen for a disk that is under the control of VxVM, such as `disk03`. Also referred to a *disk name*.

**DRL (Dirty Region Logging)**

The method by which the VxVM monitors and logs modifications to a plex as a bitmap of changed regions. For a volumes with a new-style DCO volume, the dirty region log is maintained in the DCO volume. Otherwise, the dirty region log is allocated to an associated subdisk called a *log subdisk*.

**encapsulation**

A process that converts existing partitions on a specified disk to volumes. If any partitions contain file systems, `/etc/fstab` entries are modified so that the file systems are mounted on volumes instead. This feature is only supported on the Linux and Solaris operating systems.

**enclosure**

A disk array.

**gap**

A disk region that does not contain VERITAS Volume Manager objects (subdisks).

**HP-UX coexistence label**

Data on disk which identifies the disk to the HP volume manager (LVM) as being controlled by VxVM. The contents of this label are identical to the contents of the VxVM ID block.



---

## mirror

A copy of a volume and its data. There can be several mirrors per volume. The terms *mirror* and *plex* are used synonymously.

## node

In the VxVM tree, a node is an element attached to the tree.

In a cluster environment, a node is a host machine in a cluster.

## object group

A group of objects of the same type. Each object group has a group icon and a group name. In VxVM, object groups include disk groups, disks, volumes, controllers, free disk pool disks, uninitialized disks, and file systems.

## object tree

A dynamic hierarchical display of VERITAS Volume Manager objects and other objects on the system. Each node in the tree represents a group of objects of the same type.

## platform block

Data placed in sector 0, which contains OS-specific data for a variety of platforms that require its presence for proper interaction with each of those platforms. The platform block allows a disk to masquerade as if it was initialized by each of the specific platforms.

## plex

A copy of a volume and its data. There can be several plexes per volume. The terms *mirror* and *plex* are used synonymously.

## 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. The disk header contains a disk ID. All data in the private region is duplicated for extra reliability.

## public region

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

## sector size

Sector size is an attribute of a disk drive (or SCSI LUN for an array-type device), which is set when the drive is formatted. Sectors are the smallest addressable unit of storage on the drive, and are the units in which the device performs I/O.





---

**subdisk**

A set of contiguous disk blocks that form a logical disk segment. Subdisks are associated with plexes (mirrors) to form volumes.

**uninitialized disks**

Disks that are not under VxVM control.

**volume**

A virtual disk or entity that is made up of portions of one or more physical disks.

**VxFS**

VERITAS File System.

**VxVM**

VERITAS Volume Manager.

**VxVM ID block**

Data on disk that indicates the disk is under VxVM control. The VxVM ID Block provides dynamic VxVM private region location, GUID, and other information.





# Index

---

## Symbols

- /etc/default/vxassist defaults file 17
- /etc/default/vxcdsconvert defaults file 18
- /etc/default/vxdg defaults file 18
- /etc/default/vxdisk defaults file 18
- /etc/default/vxencap defaults file 18
- /etc/vx/darecs file 12

## A

- access type 5
- activation
  - default 33
- AIX coexistence label 5
- alignment 7
  - changing 25
- alignment value 8
- attribute
  - CDS 26
- auto disk type 5

## B

- block size 3
- blockdev --rereadpt 34
- byte order 40

## C

- CDS
  - attribute 26
  - changing setting 27
  - creating DGs 13
  - creating disks 12
  - disk group alignment 3
  - disk group device quotas 6
  - disks 3
- CDS disk groups
  - alignment 31
  - joining 26
  - moving 26, 27
  - setting alignment 25
- CDS disks

- creating 11
- changing CDS setting 27
- changing default CDS setting 27
- changing default disk format setting 22
- changing disk format 22
- co-existence label 5
- concepts 1
- converting non-CDS disks to CDS 13
- converting non-CDS disks to CDS disks 14
- creating a DRL log 29
- creating CDS disk groups 13
- creating CDS disks 11, 12
- creating DRL logs 29
- creating non-CDS disk groups 27
- creating pre-version 110 disk groups 27
- cross-platform data sharing 35
  - recovery file 41
- current-rev disk groups 7

## D

- default activation 33
- default CDS setting
  - changing 27
- defaults file 14
- defaults files 17
- device quota 6
- device quotas 6, 33
  - displaying 33
  - setting 28
- disk
  - access type 5
  - change format 22
  - labels 22
  - LVM 22
  - replacing 28
- disk access 3
- disk format 3
- disk group 5
  - alignment 7



- 
- non-CDS 7
  - disk group alignment 25
    - displaying 31
  - disk groups
    - creating 27
    - joining 26
    - upgrading 27
  - disk quotas
    - setting 28
  - disk types 4
  - disks
    - effects of formatting or partitioning 34
    - displaying device quotas 33
    - displaying disk group alignment 31
    - displaying DRL log size 31
    - displaying DRL map size 31
    - displaying log map values 31
    - displaying log size 31
    - displaying v\_logmap values 31
    - displaying volume log map values 31
  - DRL log
    - creating 29
  - DRL log size
    - displaying 31
    - setting 28
  - DRL logs
    - creating 29
  - DRL map length 30
  - DRL map size
    - displaying 31
    - setting 28
  - E**
    - encapsulation 22
  - F**
    - fscdsconv 41
    - fscdstask 37
  - I**
    - I/O block size 3
    - ID block 5
  - J**
    - joining CDS disk groups 26
    - joining disk groups 26
  - L**
    - length listing 32
    - licensing 17
    - listing disk groups 32
    - listing disks 32
    - listing offset and length information 27
    - log size
      - displaying 31
      - setting 28
    - LVM disks 22
  - M**
    - minor device numbers 7
    - moving CDS disk groups 26, 27
    - moving disk group objects 26
  - O**
    - objects
      - moving 26
    - offset
      - listing 32
    - offset information 32
    - operating system data 3
  - P**
    - platform block 5
    - private region 4
    - public region 4
  - R**
    - recovery file, cross-platform data sharing 41
    - replacing disks 28
    - restoring CDS disk labels 22
    - restoring disk labels 22
  - S**
    - setting CDS disk group alignment 25
    - setting device quotas 28
    - setting disk quotas 28
    - setting DRL log size 28
    - setting DRL map length 30
    - setting DRL map size 28
    - setting log size 28
    - system
      - setting up 11
  - T**
    - Technical assistance x
  - U**
    - upgrading disk groups 27
    - upgrading pre-version 110 disk groups 27
  - V**
    - v\_logmap
      - displaying 31
    - vxcdsconvert 14

---

vxctl enable 34  
vxdg init 13  
vxdg split 33  
vxdisk scandisks 34  
vxdiskadm 12, 13

vxdisksetup 11  
VxVM  
    devices 2  
vxvol 30

