

Veritas Storage Foundation™ Cross-Platform Data Sharing Administrator's Guide

HP-UX 11i v3

5.0



Veritas Storage Foundation Cross-Platform Data Sharing Administrator's Guide

Copyright © 2007 Symantec Corporation. All rights reserved.

Veritas Storage Foundation 5.0

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

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

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

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

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014
www.symantec.com

Third-party legal notices

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

AIX is a registered trademark of IBM Corporation.

HP-UX is a registered trademark of Hewlett-Packard Development Company, L.P.

Linux is a registered trademark of Linus Torvalds.

Solaris is a trademark of Sun Microsystems, Inc.

Licensing and registration

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

Technical support

For technical assistance, visit

http://www.symantec.com/enterprise/support/assistance_care.jsp

and select phone or email support. Use the Knowledge Base search feature to access resources such as TechNotes, product alerts, software downloads, hardware compatibility lists, and our customer email notification service.

Contents

Chapter 1	Overview of CDS	
	General concepts	9
	Shared data across platforms	10
	Disk drive sector size	10
	Block size issues	11
	Operating system data	11
	CDS disk access and format	11
	CDS disk types	12
	CDS disk groups	13
	Non-CDS disk groups	15
	Disk group alignment	15
	Alignment values	16
	Object alignment during volume creation	17
Chapter 2	Setting up your system	
	Creating CDS disks from uninitialized disks	19
	Creating CDS disks by using vxdisksetup	19
	Creating CDS disks by using vxdiskadm	19
	Creating CDS disks from initialized VxVM disks	20
	Creating a CDS disk from a disk that is not in a disk group	20
	Creating a CDS disk from a disk that is already in a disk group	20
	Creating CDS disk groups	20
	Creating a CDS disk group by using vxdg init	21
	Creating a CDS disk group by using vxdiskadm	21
	Converting non-CDS disks to CDS disks	22
	Converting a non-CDS disk group to a CDS disk group	23
	Side effects of conversion	24
	Defaults files	25
Chapter 3	Maintaining your system	
	Disk tasks	27
	Changing the default disk format	27
	Restoring CDS disk labels	27
	Disk group tasks	29
	Changing the alignment of a disk group during disk encapsulation	29

- Changing the alignment of a non-CDS disk group 29
- Splitting a CDS disk group 30
- Moving objects between CDS disk groups and non-CDS disk groups .. 30
- Moving objects between CDS disk groups 31
- Joining disk groups 31
- Changing the default CDS setting for disk group creation 31
- Creating non-CDS disk groups 31
- Upgrading an older version non-CDS disk group 32
- Replacing a disk in a CDS disk group 32
- Setting the maximum number of devices for CDS disk groups 32
- Changing the DRL map and log size 33
- Creating a volume with a DRL log 33
- Setting the DRL map length 34
- Displaying information 35
 - Determining the setting of the CDS attribute on a disk group 35
 - Displaying the maximum number of devices in a CDS disk group 36
 - Displaying map length and map alignment of traditional DRL logs ... 36
 - Displaying the disk group alignment 36
 - Displaying the log map length and alignment 36
 - Displaying offset and length information in units of 512 bytes 37
- Default activation mode of shared disk groups 37
- Additional considerations when importing CDS disk groups 38

Chapter 4 File system data sharing

- Considerations about data in the file system 41
- File system migration 42
- Specifying the migration target 42
 - Examples of target specifications 43
- Using the `fsccsadm` command 44
 - Checking that the metadata limits are not exceeded 44
 - Maintaining the list of target operating systems 44
 - Enforcing the established CDS limits on a file system 45
 - Ignoring the established CDS limits on a file system 45
 - Validating the operating system targets for a file system 46
 - Displaying the CDS status of a file system 46
- Migrating a file system one time 46
- Migrating a file system on an ongoing basis 47
 - Stopping ongoing migration 48
- When to convert a file system 48
- Converting the byte order of a file system 48
 - Importing and mounting a file system from another system 51

Appendix A Transferring data between platforms

	Alignment value and block size	53
	Default activation mode of shared disk groups	53
	Moving disk groups between HP-UX and Linux systems	54
	Migrating a snapshot volume	54
Appendix B	Error codes and error recovery	
Glossary		61
Index		65

Overview of CDS

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 Symantec's 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

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.

Note: 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 such access is supported at all levels that are required.

Shared data across platforms

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 the following functionality applies:

- 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.

An 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. 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.

Disk drive 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 issues

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 | <ul style="list-style-type: none">■ 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)

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 [“Converting non-CDS disks to CDS disks”](#) on page 22.

CDS disk types

The CDS disk format, `cdsdisk`, is recognized by all VxVM platforms (including Windows). The `cdsdisk` disk format is the default for all newly-created VM disks unless overridden in a defaults file. The `vxcdsconvert` utility is provided to convert other disk formats and types to CDS.

See [“Defaults files”](#) on page 25.

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 32MB, which is large enough to record the details of several thousand 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 the disk group alignment 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 logical 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 logical 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 [“Converting non-CDS disks to CDS disks”](#) on page 22.

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 15.

CDS disk groups can be used in the following ways:

- 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.
- Imported as private disk groups, shared disk groups (by CVM), or distributed disk groups (by a volume server).

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

- Volumes of usage type `root` and `swap`. 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 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. Device quotas are useful for disk groups which to be transferred between Linux with a pre-2.6 kernel and other supported platforms such as HP-UX. 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.

See [“Setting the maximum number of devices for CDS disk groups”](#) on page 32.

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 [“Displaying the maximum number of devices in a CDS disk group”](#) on page 36.

Note: The default device quota for HP-UX is 32767.

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 controlled by the `cds` attribute of the disk group. Enabling this 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 the following attributes of the objects within a disk group:

- Subdisk offset
- Subdisk length
- Plex offset
- Volume length
- Log length
- Stripe width

The offset value specifies how an object is positioned on a drive.

The disk group alignment is assigned at disk group creation time.

See [“Disk group tasks”](#) on page 29.

Alignment values

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

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

All disk group versions before version 110 have an alignment value of 1 block, and they retain this value if they are upgraded to version 110 or later.

A disk group that is not a CDS disk group, and which has a version of 110 and later, can have an alignment value of either 1 block or 8k.

The alignment for all newly initialized disk groups in VxVM 4.0 and later releases is 8k. This value, which is used when creating the disk group, cannot be changed. However, the disk group alignment can be subsequently changed.

See [“Changing the alignment of a non-CDS disk group”](#) on page 29.

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

Dirty region log 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 has the following region requirements:

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

In a version 110 disk group, a version 20 DCO volume has the following region requirements:

- 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.

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.

Setting up your system

Creating CDS disks from uninitialized disks

You can create a CDS disk from an uninitialized disk by using one of the following methods:

- [Creating CDS disks by using vxdisksetup](#)
- [Creating CDS disks by using vxdiskadm](#)

Creating CDS disks by using vxdisksetup

To create a CDS disk by using the `vxdisksetup` command

- ◆ Type the following command:

```
# vxdisksetup -i disk [format=disk_format]
```

The format defaults to `cdsdisk` unless this is overridden by the `/etc/default/vxdisk` file, or by specifying the disk format as an argument to the `format` attribute.

See [“Defaults files”](#) on page 25.

See the `vxdisksetup(1M)` manual page.

Creating CDS disks by using vxdiskadm

To create a CDS disk by using the `vxdiskadm` command

- ◆ Run the `vxdiskadm` command, and select the “Add or initialize one or more disks” item from the main menu. You are prompted to specify the format.

Warning: On CDS disks, the CDS information occupies the first sector of that disk, and there is no `fdisk` partition information. Attempting to create an `fdisk` partition (for example, by using the `fdisk` or `format` commands) erases the CDS information, and can cause data corruption..

Creating CDS disks from initialized VxVM disks

How you create a CDS disk depends on the current state of the disk, as follows:

- [Creating a CDS disk from a disk that is not in a disk group](#)
- [Creating a CDS disk from a disk that is already in a disk group](#)

Creating a CDS disk from a disk that is not in a disk group

To create a CDS disk from a disk that 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 Type this command to set up the disk:

```
# vxdisksetup -i disk
```

Creating a CDS disk from a disk that is already in a disk group

To create a CDS disk from a disk that is already in a disk group, run the `vxcdsconvert` command.

See [“Converting non-CDS disks to CDS disks”](#) on page 22.

Creating CDS disk groups

You can create a CDS disk group in the following ways:

- [Creating a CDS disk group by using vxdg init](#)
- [Creating a CDS disk group by using vxdiskadm](#)

Creating a CDS disk group by using vxdg init

To create a CDS disk group by using the vxdg init command

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

- ◆ Type the following command:

```
# vxdg init diskgroup disklist [cds={on|off}]
```

The format defaults to a CDS disk group, unless this is overridden by the `/etc/default/vxdg` file, or by specifying the `cds` argument.

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

Creating a CDS disk group by using vxdiskadm

You cannot create a CDS disk group when encapsulating an existing disk, or when converting an LVM volume.

When initializing a disk, if the target disk group is an existing CDS disk group, `vxdiskadm` 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.

If you use the `vxdiskadm` command to initialize a disk into an existing CDS disk group, the disk must have been added with the `cdsdisk` format.

The CDS attribute for the disk group remains unchanged by this procedure.

To create a CDS disk group by using the vxdiskadm command

- ◆ Run the `vxdiskadm` command, and select the “Add or initialize one or more disks” item from the main menu. Specify that the disk group should be a CDS disk group when prompted.

Converting non-CDS disks to CDS disks

To convert non-CDS disks to CDS disks

Note: The disks must be of type of `auto` in order to be re-initialized as CDS disks.

- 1 If the conversion is not going to be performed on-line (that is, while access to the disk group continues), stop any applications that are accessing the disks.
- 2 Type one of the following forms of the CDS conversion utility (`vxcdsconvert`) to convert non-CDS disks to CDS disks.

```
# 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] ...
```

The `alldisks` and `disk` keywords have the following effect

<code>alldisks</code>	Converts all non-CDS disks in the disk group into CDS disks.
<code>disk</code>	Specifies a single disk for conversion. You would use this option under the following circumstances: <ul style="list-style-type: none">■ If a disk in the non-CDS disk group has cross-platform exposure, you may want other VxVM nodes to recognize the disk, but not to assume that it is available for initialization.■ If the native Logical Volume Manager (LVM) that is provided by the operating system needs to recognize CDS disks, but it is not required to initialize or manage these disks.■ Your intention is to move the disk into an existing CDS disk group.

The conversion involves evacuating objects from the disk, reinitializing the disk, and relocating objects back to disk. You can specify the `-o novolstop` option to perform the conversion on-line (that is, while access to the disk continues). If the `-o novolstop` option is not specified, stop any applications that are accessing the disks, and perform the conversion off-line.

Warning: Specifying the `-o novolstop` option can greatly increase the amount of time that is required to perform conversion.

Before you use the `vxcdsconvert` command, make sure you understand its options, attributes, and keywords.

See the `vxcdsconvert(1M)` manual page.

Converting a non-CDS disk group to a CDS disk group

To convert a non-CDS disk group to a CDS disk group

- 1 If the disk group contains one or more disks that you do not want to convert to CDS disks, use the `vx dg move` or `vx dg split` command to move the disks out of the disk group.
- 2 The disk group to be converted must have the following characteristics:
 - No dissociated or disabled objects.
 - No sparse plexes.
 - No volumes requiring recovery.
 - No volumes with pending snapshot operations.
 - No objects in an error state.

To verify whether a non-CDS disk group can be converted to a CDS disk group, type the following command:

```
# vxcdsconvert -g diskgroup -A group
```

- 3 If the disk group does not have a CDS-compatible disk group alignment, the objects in the disk group must be relayed out with a CDS-compatible alignment.
- 4 If the conversion is not going to be performed on-line (that is, while access to the disk group continues), stop any applications that are accessing the disks.
- 5 Type one of the following forms of the CDS conversion utility (`vxcdsconvert`) to convert a non-CDS disk group to a CDS disk group.

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

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

The `alignment` and `group` keywords have the following effect

<code>alignment</code>	Specifies alignment conversion where disks are not converted, and an object relayout 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.
<code>group</code>	Specifies group conversion of all non-CDS disks in the disk group before realying out objects in the disk group.

The conversion involves evacuating objects from the disk, reinitializing the disk, and relocating objects back to disk. You can specify the `-o novolstop` option to perform the conversion on-line (that is, while access to the disk group continues). If the `-o novolstop` option is not specified, stop any applications that are accessing the disks, and perform the conversion off-line.

Warning: Specifying the `-o novolstop` option can greatly increase the amount of time that is required to perform conversion.

In the following example, the disk group, `anodg`, and all its disks are converted to CDS while keeping its volumes are still online:

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

The `evac_disk_list` attribute specifies a list of disks (`nodg11` through `anodg14`) to which subdisks can be evacuated to disks if required.

Before you use the `vxcdsconvert` command, make sure you understand its options, attributes, and keywords.

See the `vxcdsconvert(1M)` manual page.

Side effects of conversion

Conversion has the following side effects:

- Non-CDS disk group are upgraded by using the `vxvg upgrade` command. If the disk group was originally created by the conversion of an LVM volume group (VG), rolling back to the original LVM VG is not possible. 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.

- Stopped, but startable, volumes are started for the duration of the conversion.
- Any volumes or other objects in the disk group that were created with the `layout=diskalign` attribute specified can no longer be disk aligned.
- Encapsulated disks may lose the ability to be unencapsulated.
- Performance may be degraded because data may have migrated to different regions of a disk, or to different disks.

Defaults files

The following system defaults files in the `/etc/default` directory are used to specify 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

<code>vxassist</code>	<p>Specifies default values for the following parameters to the <code>vxcdsconvert</code> command that have an effect on the alignment of VxVM objects: <code>dgalign_checking</code>, <code>diskalign</code>, and <code>nodiskalign</code>.</p> <p>See “Object alignment during volume creation” on page 17. See the <code>vxassist(1M)</code> manual page.</p>
<code>vxcdsconvert</code>	<p>Specifies default values for the following parameters to the <code>vxcdsconvert</code> command: <code>evac_disk_list</code>, <code>evac_subdisks_ok</code>, <code>min_split_size</code>, <code>move_subdisks_ok</code>, <code>privlen</code>, and <code>split_subdisks_ok</code>.</p> <p>The following is a sample <code>vxcdsconvert</code> defaults file:</p> <pre>evac_subdisks_ok=no min_split_size=64k move_subdisks_ok=yes privlen=2048 split_subdisks_ok=move</pre> <p>An alternate defaults file can be specified by using the <code>-d</code> option with the <code>vxcdsconvert</code> command. See the <code>vxcdsconvert(1M)</code> manual page.</p>

<code>vx dg</code>	<p>Specifies default values for the <code>cds</code>, <code>default_activation_mode</code> and <code>enable_activation</code> parameters to the <code>vx dg</code> command. The <code>default_activation_mode</code> and <code>enable_activation</code> parameters are only used with shared disk groups in a cluster.</p> <p>The following is a sample <code>vx dg</code> defaults file:</p> <pre>cds=on</pre> <p>See the <code>vx dg(1M)</code> manual page.</p>
<code>vx disk</code>	<p>Specifies default values for the <code>format</code> and <code>privlen</code> parameters to the <code>vx disk</code> and <code>vx disksetup</code> commands. These commands are used when disks are initialized by VxVM for the first time. They are also called implicitly by the <code>vx diskadm</code> command and the Veritas Enterprise Administrator (VEA) GUI.</p> <p>The following is a sample <code>vx disk</code> defaults file:</p> <pre>format=cdsdisk privlen=2048</pre> <p>See the <code>vx disk(1M)</code> manual page.</p> <p>See the <code>vx disksetup(1M)</code> manual page.</p>
<code>vx encap</code>	<p>Specifies default values for the <code>format</code>, <code>privlen</code>, <code>privoffset</code> and <code>puboffset</code> parameters to the <code>vx encap</code> and <code>vx lvmencap</code> commands. These commands are 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 <code>vx diskadm</code>, <code>vx convert</code> (on AIX) and <code>vx vmconvert</code> (on HP-UX) commands, and by the VEA.</p> <p>The following is a sample <code>vx encap</code> defaults file:</p> <pre>format=sliced privlen=4096 privoffset=0 puboffset=1</pre> <p>See the <code>vx encap(1M)</code> manual page.</p> <p>See the <code>vx convert(1M)</code> manual page.</p> <p>See the <code>vx vmconvert(1M)</code> manual page.</p>

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.

Maintaining your system

Disk tasks

The following disk tasks are supported:

- [Changing the default disk format](#)
- [Restoring CDS disk labels](#)

Changing the default disk format

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. 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

- ◆ Edit the `/etc/default/vxencap` defaults file, and change the definition of the `format` attribute.

See [“Defaults files”](#) on page 25.

Restoring CDS disk labels

CDS disks have the following labels:

- Platform block

- AIX coexistence label
- HP-UX coexistence or 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 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.

Warning: 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, the disk flush would fix it. This could happen if another system on the fabric wrote a disk label to a disk that was actually a CDS disk in some disk group.

To rewrite the CDS ID information on a specific disk

- ◆ Type the following command:

```
# vxdisk flush disk_access_name
```

This rewrites all labels except sectors 7 and 16

To rewrite all the disks in a CDS disk group

- ◆ Type the following command:

```
# vxdg flush diskgroup
```

This rewrites all labels except sectors 7 and 16

To forcibly rewrite the AIX coexistence label in sector 7 and the HP-UX coexistence label or VxVM ID block in sector 16

- ◆ Type the following command:

```
# vxdisk -f flush disk_access_name
```

This command rewrites all labels if there exists a valid VxVM ID block that points to a valid private region. The `-f` option is required to rewrite sectors

7 and 16 when a disk is taken offline due to label corruption (possibly by a Windows system on the same fabric).

Disk group tasks

The following disk group tasks are supported:

- [Changing the alignment of a disk group during disk encapsulation](#)
- [Changing the alignment of a non-CDS disk group](#)
- [Determining the setting of the CDS attribute on a disk group](#)
- [Splitting a CDS disk group](#)
- [Moving objects between CDS disk groups and non-CDS disk groups](#)
- [Moving objects between CDS disk groups](#)
- [Joining disk groups](#)
- [Changing the 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 the maximum number of devices for CDS disk groups](#)

Changing the alignment of a disk group during disk 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.

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

To change the alignment of a disk group during disk encapsulation

- ◆ Run the `vxdiskadm` command, and select the “Add or initialize one or more disks” item from the main menu. As part of the encapsulation process, you are asked to confirm that a reduction of the disk group alignment from 8K to 1 is acceptable.

Changing the alignment of a non-CDS disk group

The alignment value can only be changed for disk groups with version 110 or greater.

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 of a disk group, use the `vxpriint` command.

See “[Displaying the disk group alignment](#)” on page 36.

To change the alignment value of a disk group

- ◆ Type the `vxdg set` command:

```
# vxdg -g diskgroup set align={1|8k}
```

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:

```
# 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 to 1 to 8K is possible only using `vxdg set` as long as the configuration does not change after the 8K to 1 transition.

See “[Converting a non-CDS disk group to a CDS disk group](#)” on page 23.

Splitting a CDS disk group

You can use the `vxdg split` command to create a CDS disk group from an existing CDS disk group. The new (target) disk group preserves the setting of the CDS attribute and alignment in the original (source) disk group.

To split a CDS disk group

- ◆ Use the `vxdg split` command to split CDS disk groups.
See the *Veritas Volume Manager Administrator's Guide*.

Moving objects between CDS disk groups 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.

To move objects between a CDS disk group and a non-CDS disk group

- ◆ Use the `vxchg move` command to move objects between a CDS disk group and a non-CDS disk groups.
See the *Veritas Volume Manager Administrator's Guide*.

Moving objects between CDS disk groups

The disk group alignment does not change as a result of moving objects between CDS disk groups.

To move objects between CDS disk groups

- ◆ Use the `vxchg move` command to move objects between CDS disk groups.
See the *Veritas Volume Manager Administrator's Guide*.

Joining 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.

To join two disk groups

- ◆ Use the `vxchg join` command to join two disk groups.
See the *Veritas Volume Manager Administrator's Guide*.

Changing the default CDS setting for disk group creation

To change the default CDS setting for disk group creation

- ◆ Edit the `/etc/default/vxchg` file, and change the setting for the `cds` attribute.

Creating non-CDS disk groups

A disk group with a version lower than 110 is given an alignment value equal to 1 when it is imported. This is because the `dg_align` value is not stored in the configuration database for such disk groups.

To create a non-CDS disk group with a version lower than 110

- ◆ Type the following `vxchg` command:

```
# vxchg -T version init diskgroup disk_name=disk_access_name
```

Upgrading an older version non-CDS disk group

You may want to upgrade a non-CDS disk group with a version lower than 110 in order to use new features other than CDS. After upgrading the disk group, the `cds` attribute is set to `off`, and the disk group has an alignment of 1.

Note: You must also perform a disk group conversion (using the `vxcdsconvert` utility) to use the CDS feature.

To upgrade the non-CDS pre-version 110 disk group

- ◆ Type the following `vx dg` command:

```
# vx dg upgrade diskgroup
```

Replacing a disk in a CDS disk group

To replace a disk in a CDS disk group

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

- ◆ Type the following commands:

```
# vx dg -k rmdisk disk_name  
# vx dg -k adddisk disk_name=disk_access_name
```

The `-k` option retains and reuses the disk media record for the disk that is being replaced. The following example shows a disk device `disk21` being reassigned to disk `mydg01`.

```
# vx dg -k rmdisk mydg01  
# vx dg -k adddisk mydg01=disk21
```

For other operating systems, use the appropriate device name format.

Setting the maximum number of devices for CDS disk groups

To set the maximum number of devices that can be created in a CDS disk group

- ◆ Type the following `vx dg set` command:

```
# vx dg set maxdev=max-devices
```

The `maxdev` attribute can take any positive integer value that is greater than the number of devices that are currently in the disk group.

Changing the DRL map 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.

To change the DRL map and log size

- ◆ Use the following commands to remove and rebuild the logs:

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

Note the following restrictions

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

Creating a volume with a DRL log

To create a volume with a traditional DRL log by using the `vxassist` command

- ◆ Type the following command:

```
# vxassist -g diskgroup make volume length mirror=2 \
  logtype=drl [loglen=len-blocks] [logmap_len=len-bytes]
```

This command creates log subdisks that are each equal to the size of the DRL log.

Note the following restrictions

- | | |
|---|---|
| If neither <code>logmap_len</code> nor <code>loglen</code> is specified | <ul style="list-style-type: none">■ <code>loglen</code> is set to a default value that is based on disk group alignment.■ <code>maplen</code> is set to a reasonable value. |
| If only <code>loglen</code> is specified | <ul style="list-style-type: none">■ For pre-version 110 disk groups, <code>maplen</code> is set to zero.■ For version 110 and greater disk groups, <code>maplen</code> is set to use all the bytes available in the on-disk map. |
| If only <code>logmap_len</code> is specified | <ul style="list-style-type: none">■ For pre-version 110 disk groups, <code>logmap_len</code> is not applicable.■ For version 110 and greater disk groups, <code>maplen</code> must be less than the number of available bytes in the on-disk map for the default log length. |

Setting the DRL map length

To set a DRL map length

- 1 Stop the volume to make the DRL inactive.
- 2 Type the following command:

```
# vxvol -g dgTestSol2 set [loglen=len-blocks] \  
[logmap_len=len-bytes] volume
```

This command does not change the existing DRL map size.

Note the following restrictions

- | | |
|---|--|
| If both <code>logmap_len</code> and <code>loglen</code> are specified | <ul style="list-style-type: none">■ if <code>logmap_len</code> is greater than <code>loglen/2</code>, <code>vxvol</code> fails with an error message. Either increase <code>loglen</code> to a sufficiently large value, or decrease <code>logmap_len</code> to a sufficiently small value.■ The value of <code>logmap_len</code> cannot exceed the number of bytes in the on-disk map. |
|---|--|

- If `logmap_len` is specified
- The value is constrained by size of the log, and cannot exceed the size of the on-disk map. The size of the on-disk map in blocks can be calculated from the following formula:

$$\text{round}(\text{loglen}/\text{nmaps}) - 24$$
 where `nmaps` is 2 for a private disk group, or 33 for a shared disk group.
 - The value of `logmap_len` cannot exceed the number of bytes in the on-disk map.
- If `loglen` is specified
- Specifying a value that is less than twice the disk group alignment value results in an error message.
 - The value is constrained by size of the logging subdisk.

Displaying information

This section describes the following tasks:

- [Determining the setting of the CDS attribute on a disk group](#)
- [Displaying the maximum number of devices in a CDS disk group](#)
- [Displaying map length and map alignment of traditional DRL logs](#)
- [Displaying the disk group alignment](#)
- [Displaying the log map length and alignment](#)
- [Displaying offset and length information in units of 512 bytes](#)

Determining the setting of the CDS attribute on a disk group

To determine the setting of the CDS attribute on a disk group

- ◆ Use the `vx dg list` command or the `vx print` command to determine the setting of the CDS attribute, as shown in the following examples:

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

# vx dg 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
```

The disk group, `dgTestSol2`, is shown as having the CDS flag set.

Displaying the maximum number of devices in a CDS disk group

To display the maximum number of devices in a CDS disk group

Type the following command:

```
# vxprint -g diskgroup -GF %maxdev
```

Displaying map length and map alignment of traditional DRL logs

To display the map length and map alignment of traditional DRL logs

- ◆ Type the following commands

```
# vxprint -g dg1 -v1 volume
# vxprint -g dg1 -vF '%name %logmap_len %logmap_align' \
volume
```

Displaying the disk group alignment

To display the disk group alignment

- ◆ Type the following command:

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

Utilities such as `vxprint` and `vxdg list` that print information about disk group records also output the disk group alignment.

Displaying the log map length and alignment

To display the log map length and alignment

- ◆ Type the following command:

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

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

```
# 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 in the following form:

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

Volume:  drlvol
info:    len=20480
type:    usetype=fsgen
state:   state=ACTIVE kernel=ENABLED cdsrecovery=0/0 (clean)
assoc:   plexes=dr1vol-01,dr1vol-02,dr1vol-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}
```

Displaying offset and length information in units of 512 bytes

To display offset and length information in units of 512 bytes

- ◆ Specify the `-b` option to the `vxprint` and `vxdisk` commands, as shown in these examples:

```
# vxprint -bm
# vxdisk -b list
```

Specifying the `-b` option enables consistent output to be obtained on different platforms. Without the `-b` option, the information is output in units of sectors. The number of bytes per sector differs between platforms. When the `vxprint -bm` or `vxdisk -b list` command is used, the output also contains the `b` suffix, so that the output can be fed back to `vxmake`.

Default activation mode of shared disk groups

The default activation mode 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.

Additional considerations when importing CDS disk groups

Before you attempt to use CDS to move disk groups between different operating systems, and if the configuration of the disks has changed since the target system was last rebooted, you should consider the following points

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. Issue the appropriate command to tell the operating system to look for the storage. (On Linux, depending on the supported capabilities of the host adapter, you may need to reboot the target system to achieve this.) 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 the Solaris and Linux operating 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.

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:

```
# vxdctl enable
# vxdisk scandisks
```


File system data sharing

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 require conversion for use on the target platform. 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.

An 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.

Specifying the migration target

Most of the operations performed by the CDS commands require the target to which the file system is to be migrated to be specified by target specifiers in the following format:

```
os_name=os_name [, os_rel=os_release] [, arch=arch]
[, vxfs_version=vxfs_ver] [bits=bits]
```

The CDS commands require the following target specifiers

<code>os_name=<i>os_name</i></code>	Specifies the name of the target operating system to which the file system is planned to be migrated. Possible values are HP-UX, AIX, SunOS, or Linux. The <code>os_name</code> field must be specified if the target is specified.
<code>os_rel=<i>os_release</i></code>	Specifies the operating system release version of the target. For example, 11.31.
<code>arch=<i>arch</i></code>	Specifies the architecture of the target. For example, specify <code>ia</code> or <code>pa</code> .
<code>vxfs_version=<i>vxfs_version</i></code>	Specifies the VxFS release version that is in use at the target. For example, 4.1 or 5.0.
<code>bits=<i>bits</i></code>	Specifies the kernel bits of the target. <i>bits</i> can have a value of 32 or 64 to indicate whether the target is running a 32-bit kernel or 64-bit kernel.

While *os_name* must be specified for all `fsccdsadm` invocations that permit the target to be specified, all other target specifiers are optional and are available for the user to fine tune the migration target specification.

The CDS commands use the limits information available in the default CDS limits file, `/etc/vx/cdslimitstab`. If the values for the optional target specifiers are not specified, `fsccdsadm` will choose the defaults for the specified target based on the information available in the limits file that best fits the specified target, and proceed with the CDS operation. The chosen defaults are displayed to the user before proceeding with the migration.

Note: The default CDS limits information file, `/etc/vx/cdslimitstab`, is installed as part of the VxFS package. The contents of this file are used by the VxFS CDS commands and should not be altered.

Examples of target specifications

The following are examples of target specifications

<code>os_name=HP-UX</code>	Specifies the target operating system and use defaults for the remainder.
<code>os_name=HP-UX, os_rel=11.23,arch=pa, vxfs_version=5.0,bits=64</code>	Specifies the operating system, operating system release version, architecture, VxFS version, and kernel bits of the target.
<code>os_name=HP-UX,arch=ia</code>	Specifies the operating system and architecture of the target.
<code>os_name=HP-UX,bits=64</code>	Specifies the operating system and kernel bits of the target.

Using the `fscdsadm` command

The `fscdsadm` command can be used to perform the following CDS tasks:

- [Checking that the metadata limits are not exceeded](#)
- [Maintaining the list of target operating systems](#)
- [Enforcing the established CDS limits on a file system](#)
- [Ignoring the established CDS limits on a file system](#)
- [Validating the operating system targets for a file system](#)
- [Displaying the CDS status of a file system](#)

Checking that the metadata limits are not exceeded

To check that the metadata limits are not exceeded

- ◆ Type the following command to check whether there are any file system entities with metadata that exceed the limits for the specified target operating system:

```
# fscdsadm -v -t target mount_point
```

Maintaining the list of target operating systems

When a file system will be migrated on an ongoing basis between multiple systems, the types of operating systems that are involved in these migrations are maintained in a `target_list` file. Knowing what these targets are allows VxFS to determine file system limits that are appropriate to all of these targets. The file system limits that are enforced are file size, user ID, and group ID. The contents of the `target_list` file are manipulated by using the `fscdsadm` command.

Adding an entry to the list of target operating systems

To add an entry to the list of target operating systems

- ◆ Type the following command:

```
# fscdsadm -o add -t target mount_point
```

See [“Specifying the migration target”](#) on page 42.

Removing an entry from the list of target operating systems

To remove an entry from the list of target operating systems

- ◆ Type the following command:

```
# fscdsadm -o remove -t target mount_point
```

See “[Specifying the migration target](#)” on page 42.

Removing all entries from the list of target operating systems

To remove all entries from the list of target operating systems

- ◆ Type the following command:

```
# fscdsadm -o none mount_point
```

Displaying the list of target operating systems

To display a list of all target operating systems

- ◆ Type the following command:

```
# fscdsadm -o list mount_point
```

Enforcing the established CDS limits on a file system

By default, CDS ignores the limits that are implied by the operating system targets that are listed in the `target_list` file.

To enforce the established CDS limits on a file system

- ◆ Type the following command:

```
# fscdsadm -l enforce mount_point
```

Ignoring the established CDS limits on a file system

By default, CDS ignores the limits that are implied by the operating system targets that are listed in the `target_list` file.

To ignore the established CDS limits on a file system

- ◆ Type the following command:

```
# fscdsadm -l ignore mount_point
```

Validating the operating system targets for a file system

To validate the operating system targets for a file system

- ◆ Type the following command:

```
# fscdsadm -v mount_point
```

Displaying the CDS status of a file system

The CDS status that is maintained for a file system includes the following information:

- the `target_list` file
- the limits implied by the `target_list` file
- whether the limits are being enforced or ignored
- whether all files are within the CDS limits for all operating system targets that are listed in the `target_list` file

To display the CDS status of a file system

- ◆ Type the following command:

```
# fscdsadm -s mount_point
```

Migrating a file system one time

This example describes a one-time migration of data between two operating systems. 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.

To perform a one-time migration

- 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.
See [“Setting up your system”](#) on page 19.
- 2 If the file system is using a disk layout version prior to 7, upgrade the file system to Version 7.
- 3 Use the following command to ensure that there are no files in the file system that will be inaccessible after migrating the data due to large file size or to differences in user or group ID between platforms:

```
# fscdsadm -v -t target mount_point
```

If such files exist, move the files to another file system or reduce the size of the files.

- 4 Unmount the file system:

```
# umount mount_point
```

- 5 Use the `fscdsconv` command to convert the file system to the opposite endian.

See “[Converting the byte order of a file system](#)” on page 48.

- 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 27.

- 7 Mount the file system on the target system.

Migrating a file system on an ongoing basis

This example describes how to migrate a file system between HP-UX and Linux 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.

To perform an ongoing migration

- 1 Use the following command to ensure that there are no files in the file system that will be inaccessible after migrating the data due to large file size or to differences in user or group ID between platforms:

```
# fscdsadm -v -t target mount_point
```

If such files exist, move the files to another file system or reduce the size of the files.

- 2 Add HP-UX and Linux to the `target_list` file:

```
# fscdsadm -o add -t os_name=HP-UX /mnt1  
# fscdsadm -o add -t os_name=Linux /mnt1
```

- 3 Enforce the limits:

```
# fscdsadm -l 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 Make the file system suitable for use on the specified target.
See “[Converting the byte order of a file system](#)” on page 48.
- 6 Make the physical storage and Volume Manager logical storage accessible on the target 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 27.
- 7 Mount the file system on the target system.

Stopping ongoing migration

To stop performing ongoing migration

- ◆ Type the following commands:

```
# fscdsadm -l ignore mount_point
# fscdsadm -o none mount_point
```

The file system is left on the current system.

When to convert a file system

When moving a file system between two systems, it is essential to run the `fscdsconv` command to perform all of the file system migration tasks. The `fscdsconv` command validates the file system to ensure that it does not exceed any of the established CDS limits on the target, and converts the byte order of the file system if the byte order of the target is opposite to that of the current system.

Warning: Prior to VxFS 4.0 and disk layout Version 6, VxFS did not officially support moving file systems between different platforms, although in many cases a user may have successfully done so. Do not move file systems between platforms when using versions of VxFS prior to Version 4, or when using disk layouts earlier than Version 6. Instead, upgrade to VxFS 4.0 or higher, and disk layout Version 6 or later. Failure to upgrade before performing cross-platform movement can result in data loss or data corruption.

Converting the byte order of a file system

Use the `fscdsconv` command to migrate a file system from one system to another.

To convert the byte order of a file system

- 1 Determine the disk layout version of the file system that you will migrate:

```
# fstyp -v /dev/vx/dsk/filesystem | grep version  
magic a501fcf5 version 7 ctime Thu Jun 1 16:16:53 2006
```

Only file systems with Version 6 or later disk layout can be converted. If the file system has an earlier disk layout version, convert the file system to Version 6 or Version 7 disk layout before proceeding.

See the `vxfsconvert(1M)` manual page.

See the `vxupgrade(1M)` manual page.

- 2 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.
- 3 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. Having the same level of failure tolerance reduces the number of failure scenarios that would require restoration from the backup.

- 4 Unmount the file system to be converted:

```
# umount mount_point
```

- 5 Use the `fscdsconv` command to export the file system to the required target:

```
# fscdsconv -f recovery_file -t target -e special_device  
target specifies the system to which you are migrating the file system.
```

See “[Specifying the migration target](#)” on page 42.

recovery_file is the name of the recovery file to be created by the `fscdsconv` command. *special_device* is the raw device or volume that contains the file system to be converted.

Include the file system that you chose in [step 3](#) 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.

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 will remove the file after a system reboot, such as that which occurs with the `/tmp` and `/var/tmp` directories on the Solaris operating system.

The recovery file is almost always a sparse file. The disk utilization of this file can best be determined by using the following command:

```
# du -sk filename
```

The recovery file is used only when the byte order of the file system must be converted to suit the specified migration target.

- 6 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 Based on the information provided regarding the migration target, `fscdscnv` constructs and displays the complete migration target and prompts the user to verify all details of the target. If the migration target must be changed, enter `n` to exit `fscdscnv` without modifying the file system. At this point in the process, `fscdscnv` has not used the specified recovery file.
- 8 If the byte order of the file system must be converted to migrate the file system to the specified target, `fscdscnv` prompts you to confirm the migration. Enter `y` to convert the byte order of the file system. If the byte order does not need to be converted, a message displays indicating this fact.
- 9 The `fscdscnv` command indicates if any files are violating the maximum file size, maximum UID, or maximum GID limits on the specified target and prompts you if it should continue. If you must take corrective action to ensure that no files violate the limits on the migration target, enter `n` to exit `fscdscnv`. At this point in the process, `fscdscnv` has not used the specified recovery file.
If the migration converted the byte order of the file system, `fscdscnv` created a recovery file. The recovery file is not removed after the migration completes, and can be used to restore the file system to its original state if required at a later time.
- 10 If a failure occurs during the conversion, the failure could be one of the following cases:
 - System failure.
 - `fscdscnv` 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_device
```

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`.

- 11 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 10](#).

Under some circumstances, you will be required to restore the file system from the backup, 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.

Importing and mounting a file system from another system

The `fscdsconv` command can be used to import and mount a file system that was previously used on another system.

To import and mount a file system from another system

- 1 Convert the file system:

```
# fscdsconv -f recovery_file -i special_device
```

If the byte order of the file system needs to be converted

Enter **y** to convert the byte order of the file system when prompted by `fscdsconv`. If the migration converted the byte order of the file system, `fscdsconv` creates a recovery file that persists after the migration completes. If required, you can use this file to restore the file system to its original state at a later time.

If the byte order of the file system does not need to be converted

Enter **n** to convert the byte order of the file system when prompted by `fscdsconv`.

Transferring data between platforms

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 mode of shared disk groups

The default activation mode of shared disk groups is a local in-kernel policy that differs between platforms. 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.

Moving disk groups between HP-UX and Linux systems

A disk group created on an HP-UX system typically has device numbers greater than 1000. When that disk group is imported on a Linux machine with a pre-2.6 kernel, the devices are reassigned minor numbers less than 256.

If a disk group on a Linux system is imported to an HP-UX system, 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.

Migrating a snapshot volume

This example demonstrates how to migrate a snapshot volume containing a VxFS file system from an HP-UX system (big endian) to a Linux system (little endian).

To migrate a snapshot volume

- 1 On the HP-UX system, create the instant snapshot volume, `snapvol`, from an existing plex in the volume, `vol`, in the CDS disk group, `datadg`:

```
# vxsnap -g datadg make source=vol/newvol=snapvol/nmirror=1
```
- 2 Quiesce any applications that are accessing the volume. For example, suspend updates to the volume that contains the database tables. The database may have a hot backup mode that allows you to do this by temporarily suspending writes to its tables.
- 3 Refresh the plexes of the snapshot volume using the following command:

```
# vxsnap -g datadg refresh snapvol source=yes syncing=yes
```

- 4 The applications can now be unquiesced.

If you temporarily suspended updates to the volume by a database in [step 2](#), release all the tables from hot backup mode.

- 5 Use the `vxsnap syncwait` command to wait for the synchronization to complete:

```
# vxsnap -g datadg syncwait snapvol
```

- 6 Check the integrity of the file system, and then mount it on a suitable mount point:

```
# fsck -F vxfs /dev/vx/rdisk/datadg/snapvol  
# mount -F vxfs /dev/vx/dsk/datadg/snapvol /mnt
```

- 7 Confirm whether the file system can be converted to the target operating system:

```
# fscdstask validate Linux /mnt
```

- 8 Unmount the snapshot:

```
# umount /mnt
```

- 9 Convert the file system to the opposite endian:

```
# fscdsconv -f /tmp/fs_recov/recov.file /dev/vx/dsk/datadg/  
snapvol
```

This step is only required if the source and target systems have the opposite endian configuration.

- 10 Split the snapshot volume into a new disk group, `migdg`, and deport that disk group:

```
# vxdg split datadg migdg snapvol  
# vxdg deport migdg
```

- 11 Import the disk group, `migdg`, on the Linux system:

```
# vxdg import migdg
```

It may be necessary to reboot the Linux system so that it can detect the disks.

- 12 Use the following commands to recover and restart the snapshot volume:

```
# vxrecover -g migdg -m snapvol  
# vxvol -g migdg start snapvol
```

- 13 Check the integrity of the file system, and then mount it on a suitable mount point:

```
# fsck -t vxfs /dev/vx/dsk/migdg/snapvol  
# mount -t vxfs /dev/vx/dsk/migdg/snapvol /mnt
```


Error codes and error recovery

[Table B-1](#) lists the CDS error codes and the action that is required.

Table B-1 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 Symantec that enables the usage of CDS disk groups.

Table B-1 Error codes and required actions

Error number	Message	Action
333	Non-CDS disk cannot be placed in a CDS disk group	Do one of the following: <ul style="list-style-type: none"> ■ Add the disk to another disk group that is a non-CDS disk group. ■ Re-initialize the disk as a CDS disk so that it can be added to the CDS disk group. ■ Change the CDS disk group into a non-CDS disk group and then add the disk.
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.
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	Ensure that the length of the volume is a multiple of 8K. For a log, set the value of the <code>dgalign_checking</code> attribute to <code>round</code> . This ensures that the length of the log is silently rounded to a valid value.
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.

Table B-1 Error codes and required actions

Error number	Message	Action
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 max_autogrow violates disk group alignment	Ensure that cache attribute value is a multiple of 8K.
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 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. 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), `disk11` (HP-UX 11i v3 onwards), `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 as 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 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 25
- /etc/default/vxcdsconvert defaults file 25
- /etc/default/vxdg defaults file 26
- /etc/default/vxdisk defaults file 26
- /etc/default/vxencap defaults file 26
- /etc/vx/darecs file 20

A

- access type 12
- activation
 - default 36
- AIX coexistence label 13
- alignment 15
 - changing 29
- attribute
 - CDS 35
- auto disk type 12

B

- block size 11

C

- CDS
 - attribute 35
 - changing setting 31
 - creating DGs 20
 - creating disks 20
 - disk group alignment 11
 - disk group device quotas 14
 - disks 11
- CDS disk groups
 - alignment 36
 - joining 31
 - moving 30, 31
 - setting alignment 29
- CDS disks
 - creating 19
- changing CDS setting 31

- changing default CDS setting 31
- changing disk format 27
- co-existence label 13
- concepts 9
- converting non-CDS disks to CDS 21
- converting non-CDS disks to CDS disks 22
- creating a DRL log 33
- creating CDS disk groups 20
- creating CDS disks 19, 20
- creating DRL logs 33
- creating non-CDS disk groups 31
- creating pre-version 110 disk groups 31
- cross-platform data sharing
 - recovery file 49
- current-rev disk groups 15

D

- default activation 36
- default CDS setting
 - changing 31
- defaults files 22, 25
- device quotas 14, 36
 - displaying 36
 - setting 32
- disk
 - access type 12
 - change format 27
 - labels 27
 - LVM 27
 - replacing 32
- disk access 11
- disk format 11
- disk group alignment 29
 - displaying 36
- disk groups 13
 - alignment 15
 - creating 31
 - joining 31
 - non-CDS 15
 - upgrading 32
- disk quotas

- setting 32
- disk types 12
- disks
 - effects of formatting or partitioning 38
- displaying device quotas 36
- displaying disk group alignment 36
- displaying DRL log size 36
- displaying DRL map size 36
- displaying log map values 36
- displaying log size 36
- displaying v_logmap values 36
- displaying volume log map values 36
- DRL log size
 - displaying 36
 - setting 33
- DRL logs
 - creating 33
- DRL map length 34
- DRL map size
 - displaying 36
 - setting 33

E

- encapsulation 27

F

- fscdsadm 44
- fscdsconv 48

I

- I/O block size 11
- ID block 13

J

- joining CDS disk groups 31
- joining disk groups 31

L

- length listing 37
- listing disk groups 37
- listing disks 37
- listing offset and length information 31
- log size
 - displaying 36
 - setting 33
- LVM disks 27

M

- minor device numbers 15
- moving CDS disk groups 30, 31
- moving disk group objects 30

O

- objects
 - moving 30
- offset
 - listing 37
- offset information 37
- operating system data 11

P

- platform block 13
- private region 12
- public region 12

R

- recovery file, cross-platform data sharing 49
- replacing disks 32
- restoring CDS disk labels 27
- restoring disk labels 27

S

- setting CDS disk group alignment 29
- setting device quotas 32
- setting disk quotas 32
- setting DRL log size 33
- setting DRL map length 34
- setting DRL map size 33
- setting log size 33

U

- upgrading disk groups 32
- upgrading pre-version 110 disk groups 32

V

- v_logmap
 - displaying 36
- vxcdsconvert 22
- vxctl enable 38
- vx dg init 21
- vx dg split 30
- vx disk scandisks 38

vxdiskadm 19, 21
vxdisksetup 19
VxVM
 devices 10
vxvol 34

