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

5.0

N18456C

symantec™

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

Veritas Storage Foundation 5.0

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

Contents

Chapter 1 Overview of CDS

General concepts 9
- Sharing 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

Chapter 2 Setting up your system

Creating CDS disks from uninitialized disks 19
- Using vxdisksetup 19
- Using vxdiskadm 20

Creating CDS disks from initialized VxVM disks 20
- Disk not in a disk group 20
- Disk already in a disk group 20

Creating CDS disk groups 21
- Using vxdg init to create CDS disk groups 21
- Using vxdiskadm to create CDS disk groups 21

Converting a non-CDS disk to a CDS disk 22
Converting a non-CDS disk group to a CDS disk group 23
- Notes 24
- Examples 24

Licensing 25
Defaults files 25

Chapter 3 Maintaining your system

Disk tasks 30
- Changing the default disk format setting 30
- Restoring CDS disk labels 30

Disk group tasks 32

Changing the alignment of a disk group by encapsulation 32
Object alignment during volume creation 32
Changing the alignment of a non-CDS disk group 33
Determining the setting of the CDS attribute 33
Joining disk groups 34
Moving objects between CDS disk groups and non-CDS disk groups 34
Moving objects between CDS disk groups 34
Changing the default CDS setting for disk group creation 35
Creating non-CDS disk groups 35
Upgrading an older version non-CDS disk group 35
Replacing a disk in a CDS disk group 35
Setting device quotas for CDS disk groups 36
Setting DRL map size and log size 36
Creating a DRL log 37
Displaying information 38
Displaying traditional DRL map size and log size 38
Displaying the disk group alignment 38
Displaying volume log map values 39
Listing offset and length information 39
Listing CDS disk groups 40
Listing disks 40
Displaying device quotas for CDS disk groups 40
Default activation mode of shared disk groups 41
Creating CDS disk groups using vxdg split 41
Additional considerations when importing CDS disk groups 41

Chapter 4 File system considerations

Considerations about data in the file system 44
File system migration 45
Specifying the migration target 45
Example target specifications 46
Using the fscdsadm command 47
One-time migration of a file system 48
Ongoing migration of a file system 49
Ceasing ongoing migration 50
When to convert a file system 50
Migrating a file system 50
Importing and mounting a file system from another system 53

Appendix A Cross-platform transfer

Alignment value and block size 55
Default activation of shared disk groups 55

Disk group alignment and encapsulated disks 55
Importing disk groups between Linux and non-Linux machines 56
Data migration example 56

Appendix B Error codes and error recovery

Glossary 63

Index 67

Chapter 1

Overview of CDS

This chapter presents an overview of the Cross-Platform Data Sharing (CDS) feature of Symantec's Veritas Storage Foundation™ software:

- General concepts
- CDS disk access and format
- CDS disk groups

General concepts

CDS provides you with a foundation for moving data between different systems within a heterogeneous environment. The machines may be running HP-UX, AIX, Linux or the Solaris™ operating system (OS), and they may all have direct access to physical devices holding data. CDS allows 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 is a license-enabled feature that is supported at the disk group level by VxVM and at the file system level by VxFS.

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

Sharing 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:

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

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

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

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

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
  - The data may not have been created on a block boundary compatible with that used by the accessing platform.
  - The accessing platform cannot address the start of the data.
- Bleed-over issues
  The size of the data written may not be an exact multiple of the block size used by the accessing platform. Therefore the accessing platform cannot constrain its I/O within the boundaries of the data on disk.

Operating system data

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

CDS disk access and format

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

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

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

Note: The CDS conversion utility, `vxcdsconvert`, is provided to convert non-CDS VM disk formats to CDS disks, and disk groups with a version number less than 110 to disk groups that support CDS disks. See "Setting up your system" on page 19 for more details.

Disk groups with version numbers less than 110 are not supported for the Solaris OS on the x64 platform.

CDS disk types

The CDS disk format, `cdsdisk`, is recognized by all VxVM platforms (including Windows). This is the default disk format for all newly-created VM disks unless overridden in a defaults file (see "Defaults files" on page 25). The `vxcdsconvert` utility is provided to convert other disk formats and types to CDS.

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

Private and public regions

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

The private region is a small area on the disk where VxVM configuration information is stored, such as a disk header label, configuration records for VxVM objects (such as volumes, plexes and subdisks), and an intent log for the configuration database. The default private region size is 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 this attribute must also be 8k to permit the operation of CDS.

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

Disk access type auto

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

Platform block

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

AIX coexistence label

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

HP-UX coexistence label

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

VxVM ID block

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

CDS disk groups

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

Note: The CDS conversion utility, `vxcdsconvert`, is provided to convert non-CDS VM disk formats to CDS disks, and disk groups with a version number less than 110 to disk groups that support CDS disks. See "Setting up your system" on page 19 for more details.

All VxVM objects in a CDS disk group are aligned and sized so that any system can access the object using its own representation of an I/O block. The CDS disk

group uses a platform-independent alignment value to support system block sizes of up to 8K. See "Disk group alignment" on page 15 for details.

CDS disk groups can be:

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

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

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

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

Device quotas

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

You can limit the number of devices that can be created in a given CDS disk group by setting the device quota (refer to "Setting device quotas for CDS disk groups" on page 36).

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

See also "Displaying device quotas for CDS disk groups" on page 40 for instructions on displaying the device quota value.

Minor device numbers

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

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

Non-CDS disk groups

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

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

Disk group alignment

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

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

The disk group alignment is assigned at disk group creation time (see “Maintaining your system” on page 29).

Alignment values

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

Alignment value 1

Note: Disk groups with version numbers less than 110 are not supported for the Solaris OS on the x64 platform.

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

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

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

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

Alignment value 8K

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

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

The alignment for all newly initialized disk groups in VxVM 4.0 or a later release is 8k. This value (which is used when creating the disk group) cannot be changed. However, the disk group alignment can be subsequently changed (refer to "Changing the alignment of a non-CDS disk group" on page 33).

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

Dirty region log alignment

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

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

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

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

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

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

Chapter 2

Setting up your system

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

- Creating CDS disks from uninitialized disks
- Creating CDS disks from initialized VxVM disks
- Creating CDS disk groups
- Converting a non-CDS disk to a CDS disk
- Converting a non-CDS disk group to a CDS disk group
- Licensing
- Defaults files

Creating CDS disks from uninitialized disks

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

- `vxdisksetup`
- `vxdiskadm`

Using vxdisksetup

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

```
# vxdisksetup -i disk
```

The format defaults to `cdsdisk` unless overridden by the `/etc/default/vxdisk` file (refer to “Defaults files” on page 25).

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

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

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

Using vxdiskadm

Use the "Add or initialize one or more disks" option. You are prompted to specify the format.

Caution: 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

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

- Disk not in a disk group
- Disk already in a disk group

Disk not in a disk group

If the disk is not in a disk group:

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

   ```
   # vxdiskunsetup disk
   ```

   This is necessary as non-auto types cannot be reinitialized by vxdisksetup.
2. If the disk is listed in the /etc/vx/darecs file, remove its disk access (DA) record using the command:

   ```
   # vxdisk rm disk
   ```

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

   ```
   # vxdisk scandisks
   ```
4. Finally, run this command to set up the disk:

   ```
   # vxdisksetup -i disk
   ```

Disk already in a disk group

If the disk is already in a disk group, run vxcdsconvert, as described in "Converting a non-CDS disk to a CDS disk" on page 22.

Creating CDS disk groups

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

- Using vxdg init to create CDS disk groups
- Using vxdiskadm to create CDS disk groups

Using vxdg init to create CDS disk groups

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

```
# vxdg init diskgroup disklist
```

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

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

```
# vxdg init diskgroup disklist cds=on
```

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

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

Using vxdiskadm to create CDS disk groups

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

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

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

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

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

Converting a non-CDS disk to a CDS disk

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- Use the `alignment` keyword to specify alignment conversion. In this case, disks are not converted, and an object 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.
- Use the `group` keyword to specify group conversion. This implies `alldisk`, and will perform that function prior to object relayout. All the non-CDS disks in the disk group are converted.

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

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

Note the following:

- The disk group must be in pristine condition. That is:
  - It has no dissociated or disabled objects.
  - No sparse plexes are present.
  - There are no volumes requiring recovery or having pending snapshot operations.
  - There are no objects in an error state.
- Stopped (but startable) volumes will be started, for the duration of the conversion only.
- Conversion has the following side effects:

- Any objects created with `layout=diskalign` can no longer be disk aligned.
- Encapsulated disks may lose the ability to be unencapsulated.
- Fine performance tuning may be lost as data may have migrated (and even migrated to different disks).

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

Notes

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

Examples

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

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

- Make the disk group `mydg` and all its disks portable while its volumes are still online. You can avoid using the `novolstop` option during disk conversion by using the following sequence of commands:

- Stop all applications that are using volumes on the disks requiring format conversion.
- Enter the command:

```
# vxcdsconvert -g mydg alldisks
```

- Restart applications.
- Enter the command:

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

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

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

Licensing

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

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

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

Defaults files

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

`vxassist` Specifies default values for the following parameters to the `vxcdsconvert` command that have an effect on the alignment of VxVM objects: `dgalign_checking`, `diskalign`, and `nodiskalign`. See "Object alignment during volume creation" on page 32, and the `vxassist`(1M) manual page for more information.

`vxcdsconvert` Specifies default values for the following parameters to the `vxcdsconvert` command: `evac_disk_list`, `evac_subdisks_ok`, `min_split_size`,

move_subdisks_ok, privlen, and split_subdisks_ok. The following is a sample vxcdsconvert defaults file:

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

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

vxdg

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

```
cds=on
```

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

vxdisk

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

```
format=cdsdisk
privlen=2048
```

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

vxencap

Specifies default values for the format, privlen, privoffset and puboffset parameters to the vxencap and vxlvmencap 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 vxdiskadm, vxconvert (on AIX) and vxvmconvert (on HP-UX) commands, and by the VEA. The following is a sample vxencap defaults file:

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

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

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

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

Chapter 3

Maintaining your system

This chapter describes the administration tasks that can be performed on CDS.

- Disk tasks
  - Changing the default disk format setting
  - Restoring CDS disk labels
- Disk group tasks
  - Changing the alignment of a disk group by encapsulation
  - Object alignment during volume creation
  - Changing the alignment of a non-CDS disk group
  - Determining the setting of the CDS attribute
  - Joining disk groups
  - Moving objects between CDS disk groups and non-CDS 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 device quotas for CDS disk groups
- Setting DRL map size and log size
- Displaying information
  - Displaying the disk group alignment
  - Displaying volume log map values
  - Listing offset and length information
  - Listing CDS disk groups
  - Listing disks
  - Displaying device quotas for CDS disk groups

- Creating CDS disk groups using vxdg split
- Default activation mode of shared disk groups
- Additional considerations when importing CDS disk groups

Disk tasks

The following disk tasks are supported:

- Changing the default disk format setting
- Restoring CDS disk labels

Changing the default disk format setting

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

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

You can override this behavior by changing the settings in the system defaults files as described in "Defaults files" on page 25. For example, you can change the default format to `sliced` for disk initialization by modifying the definition of the `format` attribute in the `/etc/default/vxdisk` defaults file. To change the default format for disk encapsulation or LVM disk conversion, change the definition of the `format` attribute in the `/etc/default/vxencap` defaults file.

Restoring CDS disk labels

CDS disks have three labels:

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

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

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

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

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

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

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

```
# vxdisk flush disk_accessname
```

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

```
# vxdg flush dg_name
```

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

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

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

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

Disk group tasks

The following disk group tasks are supported:

- Changing the alignment of a disk group by encapsulation
- Object alignment during volume creation
- Changing the alignment of a non-CDS disk group
- Determining the setting of the CDS attribute
- Joining disk groups
- Moving objects between CDS disk groups and non-CDS 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 device quotas for CDS disk groups

Changing the alignment of a disk group by encapsulation

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

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

Object alignment during volume creation

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

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

Changing the alignment of a non-CDS disk group

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

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

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

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

To display the current alignment value, use the vxprint command as described in "Displaying the disk group alignment" on page 38.

The operation to increase the alignment to 8K fails if objects exist in the disk group that do not conform to the new alignment restrictions. In that case, use the vxcdsconvert alignment command to change the layout of the objects as described in "Converting a non-CDS disk group to a CDS disk group" on page 23:

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

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

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

Determining the setting of the CDS attribute

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

```
# vxdg list
NAME            STATE               ID
dgTestSol2      enabled,cds         1063238039.206.vmesc1

# vxdg list dgTestSol2
Group:     dgTestSol2
dgid:      1063238039.206.vmesc1
import-id: 1024.205
flags:     cds
```

```
version:   110
alignment: 8192 (bytes)
...

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

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

Joining disk groups

Use the `vxdg join` command to join two disk groups. Joining two CDS disk groups or joining two non-CDS disk groups is permitted, but you cannot join a CDS disk group to a non-CDS disk group. If two non-CDS disk groups have different alignment values, the alignment of the resulting joined disk group is set to 1, and an informational message is displayed. Refer to the *Veritas Volume Manager Adminstrator's Guide* for examples of using this command.

Note: This command is not supported for the Solaris OS on the x64 platform.

Moving objects between CDS disk groups and non-CDS disk groups

Use the `vxdg move` command to move objects between CDS and non-CDS disk groups. The alignment of a source non-CDS disk group must be 8K to allow objects to be moved to a target CDS disk group. If objects are moved from a CDS disk group to a target non-CDS disk group with an alignment of 1, the alignment of the target disk group remains unchanged. Refer to the *Veritas Volume Manager Adminstrator's Guide* for examples of using this command.

Note: This command is not supported for the Solaris OS on the x64 platform.

Moving objects between CDS disk groups

Use the `vxdg move` command to move objects between CDS disk groups. The disk group alignment does not change. Refer to the *Veritas Volume Manager Adminstrator's Guide* for examples of using this command.

Note: This command is not supported for the Solaris OS on the x64 platform.

Changing the default CDS setting for disk group creation

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

Creating non-CDS disk groups

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

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

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

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

Disk groups with version numbers less than 110 are not supported for the Solaris OS on the x64 platform.

Upgrading an older version non-CDS disk group

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

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

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

```
# vxdg upgrade dg
```

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

Replacing a disk in a CDS disk group

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

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

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

For example:

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

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

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

Setting device quotas for CDS disk groups

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

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

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

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

Setting DRL map size and log size

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

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

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

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

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

Note the following restrictions:

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

Creating a DRL log

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

- vxassist
- vxvol

Using vxassist to create a DRL log

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

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

If neither logmap_len nor loglen is specified, then:

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

If only loglen is specified, then:

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

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

Using vxvol to set a DRL map length

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

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

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

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

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

The value of logmap_len is constrained by the log length.

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

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

```
[ROUND_DOWN(loglen/nmaps) - VOLDRL_HEADER_SIZE]
```

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

Displaying information

This section describes:

- Displaying traditional DRL map size and log size
- Displaying the disk group alignment
- Displaying volume log map values
- Listing offset and length information
- Listing CDS disk groups
- Listing disks
- Displaying device quotas for CDS disk groups

Displaying traditional DRL map size and log size

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

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

Displaying the disk group alignment

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

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

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

```
# vxdg list dg1
```

Displaying volume log map values

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

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

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

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

The output is of the form:

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

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

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

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

Volume:   drlvol
info:     len=20480
type:     usetype=fsgen
state:    state=ACTIVE kernel=ENABLED cdsrecovery=0/0 (clean)
assoc:    plexes=drlvol-01,drlvol-02,drlvol-03
policies: read=SELECT (round-robin) exceptions=GEN_DET_SPARSE
flags:    closed writecopy writeback
logging:  type=REGION loglen=528 serial=0/0 mapalign=16
maplen=512 (enabled)
apprecov: seqno=0/0
recovery: mode=default
recov_id=0
device:   minor=46000 bdev=212/46000 cdev=212/46000
path=/dev/vx/dsk/dgTestSol/drlvol
perms:    user=root group=root mode=0600
guid: {d968de3e-1dd1-11b2-8fc1-080020d223e5}
```

Listing offset and length information

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

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

Listing CDS disk groups

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

```
# vxdg list
# vxprint -Gl
```

Listing disks

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

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

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

Displaying device quotas for CDS disk groups

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

```
# vxprint -lG cdsdg1
```

The output is similar to the following:

```
group:  dg1
info:   dgid=1027365608.1064.vmesc6
version: 110
alignment: 8192 (bytes)
activation: read-write
detach-policy: global
copies: nconfig=default nlog=default
devices: max=32000 cur=0
minors: >= 16000
cds=on
```

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

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

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

Default activation 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.

Creating CDS disk groups using vxdg split

You can use the `vxdg split` command to create a CDS disk group from an existing CDS disk group. The new (target) and original (source) disk group retain the same CDS attribute; that is, if the source is CDS, then the new disk group is also CDS. The new disk group also has the same alignment as the original disk group. Refer to the *Veritas Volume Manager Adminstrator's Guide* for examples of using this command.

Note: This command is not supported for the Solaris OS on the x64 platform.

Additional considerations when importing CDS disk groups

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

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

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

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

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

  ```
  # blockdev --rereadpt
  ```

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

  ```
  # vxdctl 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
  ```

Chapter 4

File system considerations

This chapter describes Veritas File System (VxFS) support for cross-platform data sharing:

- Considerations about data in the file system
- File system migration
- Specifying the migration target
- Using the fscdsadm command
- One-time migration of a file system
- Ongoing migration of a file system
- Migrating a file system

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.

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

File system migration

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

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 target specifiers are described below:

os_name=*os_name*	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 os_name field must be specified if the target is specified.
os_rel=*os_release*	Specifies the operating system release version of the target. For example, specify `5.8`, `5.9`, or `5.10` for SunOS.
arch=*arch*	Specifies the architecture of the target. For example, specify `x86` or `sparc` for SunOS.
vxfs_version=*vxfs_version*	Specifies the VxFS release version that is in use at the target. For example, `4.1` or `5.0`.
bits=*bits*	Specifies the kernel bits of the target. *bits* 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 `fscdsadm` 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, `fscdsadm` 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.

Example target specifications

The following are examples of target specifications.

To specify the target operating system and use defaults for the remainder

- os_name=AIX

To specify the operating system, operating system release version, architecture, VxFS version, and kernel bits of the target

- os_name=HP-UX,os_rel=11.23,arch=ia,vxfs_version=5.0,bits=64

To specify the operating system and architecture of the target

- os_name=SunOS,arch=sparc

To specify the operating system and kernel bits of the target

- os_name=Linux,bits=32

Using the fscdsadm command

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

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

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

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

When a file system will be migrated on an ongoing basis between multiple systems, the types of systems 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 `target_list` file is manipulated by the `fscdsadm` command as follows:

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

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

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

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

  For both the `add` and `remove` commands, the target must be specified.
  See “Specifying the migration target” on page 45.

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

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

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

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

  The limits implied by *target_list* are by default ignored. After manipulating the `target_list` file, use the following commands to have the file system enforce or ignore these limits.

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

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

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

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

- To validate the file system for the targets in the `target_list` file, enter:

  ```
  # fscdsadm -v mount_point
  ```

  Various CDS statuses are maintained on a per-file system basis, including the `target_list` file, the limits implied by the `target_list` file, whether the limits are being enforced or ignored, and whether all files are within the CDS limits for all targets in the `target_list` file.

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

  ```
  # fscdsadm -s mount_point
  ```

One-time migration of a file system

The following 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 as described in "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.

   See the *Veritas Storage Foundation Installation Guide*.
3. Ensure that there are no files in the file system that will be inaccessible after migrating the data due to large file size or user/group id platform differences:

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

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

   ```
   # umount mount_point
   ```
5. Use the `fscdsconv` command to convert the file system to the opposite endian. See "Migrating a file system" on page 50.

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 29 for more information.

7 Mount the file system on the target system.

Ongoing migration of a file system

The following example describes how to migrate a file system between the Solaris OS 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 Ensure that there are no files in the file system that will be inaccessible after migrating the data due to large file size or user/group id platform differences:

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

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

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

```
# fscdsadm -o add -t os_name=SunOS /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 “Migrating a file system” on page 50.

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 29 for more information.

7 Mount the file system on the target system.

Ceasing ongoing migration

To stop performing ongoing migrations and leave the file system on the current system, enter the following commands to stop the usage of CDS mechanisms:

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

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.

Caution: 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.

Migrating 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) and `vxupgrade`(1M) manual pages.

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 so as to reduce the number of failure scenarios that would require the use of the backup created in step 2.

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
```

target specifies the system to which you are migrating the file system. See "Specifying the migration target" on page 45.

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

Note: The recovery file is not only used for recovery purposes after a failure, but is also used to perform the conversion. The directory that will contain the recovery file should not allow non-system administrator users to remove or replace the file, as this could lead to data loss or security breaches. The file should be located in a directory that is not subject to system or local scripts 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.

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.

6 Based on the information provided regarding the migration target, `fscdsconv` constructs and displays the complete migration target and prompts the use to verify all details of the target. If the migration target must be changed, enter **`n`** to exit `fscdsconv` without modifying the file system. At this point in the process, `fscdsconv` has not used the specified recovery file.

7 If the byte order of the file system must be converted to migrate the file system to the specified target, `fscdsconv` 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.

8 The `fscdsconv` 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 `fscdsconv`. At this point in the process, `fscdsconv` has not used the specified recovery file.
If the migration converted the byte order of the file system, `fscdsconv` 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.

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

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

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

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

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

10 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 9.
Under some circumstances, you will be required to restore the file system from the backup created in step 2, 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 also 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
   ```

2. If the byte order of the file system must be converted to migrate the file system to the specified target, `fscdsconv` 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.

   If the migration converted the byte order of the file system, `fscdsconv` 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.

   Once `fscdsconv` completes importing the file system, a message displays indicating that the import is complete.

Appendix A

Cross-platform transfer

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

Alignment value and block size

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

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

Default activation of shared disk groups

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

Disk group alignment and encapsulated disks

On the Solaris OS, all native file systems are cylinder aligned. Encapsulating such a disk results in subdisks that are also cylinder aligned. Such alignment will normally not be 8K aligned, but it will be 1K aligned. For the encapsulation process, there is no flexibility as to where on the disk the subdisks must be since the data location is predefined. If an alignment conflict occurs, user

intervention is required. If the disk group alignment is 8K this operation will probably fail because this would require the cylinder to be an even number of 8K blocks in size.

Importing disk groups between Linux and non-Linux machines

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

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

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

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

Data migration example

The following example demonstrates how to migrate a snapshot volume containing a VxFS file system from a Solaris SPARC system (big endian) to a Linux system (little endian):

1. On the Solaris 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. For example, 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/rdsk/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
```

Note: 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
```

Note: 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
```

Appendix B

Error codes and error recovery

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: ■ 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 volume length value is a multiple of 8K. For volume log length; if you are using `vxassist` to create the volume, then you should set `dgalign_checking` to `round`. This will ensure that the log length is silently rounded to a valid value.
340	Last disk media offset violates disk group alignment	Reassociate the DM record prior to upgrading.

Table B-1 Error codes and required actions

Error number	Message	Action
341	Too many device nodes in disk group	Increase the number of device nodes allowed in the disk group, if not already at the maximum. Otherwise, you need to remove volumes from the disk group, possibly by splitting the disk group.
342	Map length too large for current log length	Use a smaller map length for the DRL/DCM log, or increase the log length and retry.
343	Volume log map alignment violates disk group alignment	Remove the DRL/DCM log, then add it back after changing the alignment of the disk group.
345	Disk group contains an old-style RVG which cannot be imported on this platform	Import the disk group on the platform that created the RVG. To import the disk group on this platform, first remove the RVG on the creating platform.
346	Cache object autogrow by 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 `vxcdsconvert` 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 (see dirty region logging). The map layout within the DCO volume changed with the release of VxVM 4.0, although the original format is still available. The old layout is available in DCO Version 0 objects, and the new layout is available in DCO Version 20 objects.

device name

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

dirty region logging

See DRL.

disk access name

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

disk group

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

disk media name

A logical or administrative name chosen for a disk that is under the control of VxVM, such as `disk03`. Also referred to 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/vxcdsconvert defaults file 25
/etc/default/vxdisk defaults file 26
/etc/vx/darecs file 20

A

access type 13
activation
    default 40
AIX coexistence label 13
alignment 15
    changing 33
alignment value 16
attribute
    CDS 33
auto disk type 13

B

block size 11
blockdev --rereadpt 42

C

CDS
    attribute 33
    changing setting 35
    creating DGs 21
    creating disks 20
    disk group alignment 11
    disk group device quotas 14
    disks 11
CDS disk groups
    alignment 38
    joining 34
    moving 34
    setting alignment 33
CDS disks
    creating 19
changing CDS setting 35
changing default CDS setting 35
changing default disk format setting 30
changing disk format 30
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 37
creating CDS disk groups 21
creating CDS disks 19, 20
creating DRL logs 37
creating non-CDS disk groups 35
creating pre-version 110 disk groups 35
cross-platform data sharing 43
    recovery file 51
current-rev disk groups 15

D

default activation 40
default CDS setting
    changing 35
defaults files 22, 25
device quotas 14, 40
    displaying 40
    setting 36
disk
    access type 13
    change format 30
    labels 30
    LVM 30
    replacing 35
disk access 11
disk format 11
disk group alignment 33
    displaying 38
disk groups 13
    alignment 15
    creating 35
    joining 34
    non-CDS 15
    upgrading 35
disk quotas

setting 36
disk types 12
disks
effects of formatting or partitioning 41
displaying device quotas 40
displaying disk group alignment 38
displaying DRL log size 38
displaying DRL map size 38
displaying log map values 38
displaying log size 38
displaying v_logmap values 38, 39
displaying volume log map values 38
DRL log size
displaying 38
setting 36
DRL logs
creating 37
DRL map length 37
DRL map size
displaying 38
setting 36

E

encapsulation 30

F

fscdsadm 47
fscdsconv 50

I

I/O block size 11
ID block 13

J

joining CDS disk groups 34
joining disk groups 34

L

length listing 39
licensing 25
listing disk groups 39
listing disks 40
listing offset and length information 35
log size
displaying 38
setting 36
LVM disks 30

M

minor device numbers 15
moving CDS disk groups 34
moving disk group objects 34

O

objects
moving 34
offset
listing 39
offset information 39
operating system data 11

P

platform block 13
private region 12
public region 12

R

recovery file, cross-platform data sharing 51
replacing disks 35
restoring CDS disk labels 30
restoring disk labels 30

S

setting CDS disk group alignment 33
setting device quotas 36
setting disk quotas 36
setting DRL log size 36
setting DRL map length 37
setting DRL map size 36
setting log size 36
system
setting up 19

U

upgrading disk groups 35
upgrading pre-version 110 disk groups 35

V

v_logmap
displaying 38, 39
vxcdsconvert 22

vxdctl enable 42
vxdg init 21
vxdg split 41
vxdisk scandisks 42
vxdiskadm 20, 21
vxdisksetup 19
VxVM
    devices 10
vxvol 37