VERITAS FlashSnap™ Point-In-Time Copy Solutions

Administrator's Guide 1.1

July 2002
N08839F

VERITAS

Disclaimer

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

Copyright

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

Contents

Introduction vii
Audience and Scope vii
Organization viii
Related Documents ix
Conventions x
Getting Help xi

Chapter 1. Point-In-Time Copy Solutions 1
Applications of Point-in-Time Copy Solutions 2
Point-In-Time Copy Scenarios 3
VERITAS Software Used in Point-In-Time Copy Scenarios 4
FastResync of Volume Snapshots 7
Disk Group Split/Join 8
Storage Checkpoints 8
Implementing Point-In Time Copy Solutions on a Primary Host 9
Implementing Off-Host Point-In-Time Copy Solutions 11
Choices for Snapshot Resynchronization 15

Chapter 2. Setting up Volume Snapshot Mirrors 17
Preparing Volumes for Taking Snapshots 17
Adding a DCO and DCO Volume 18
Creating a Snapshot Mirror of a Volume 20
Considerations for Placing DCO Plexes 21

Chapter 3. Online Database Backup 23
Making a Backup of an Online Database on the Same Host 24
Making an Off-Host Backup of an Online Database 26

Chapter 4. Off-Host Cluster File System Backup 31
Mounting a File System for Shared Access 33
Using Off-Host Processing to Back Up Cluster File Systems 33

Chapter 5. Decision Support 37
Creating a Replica Database on the Same Host 38
Resynchronizing the Data 41
Creating up an Off-Host Replica Database 42
Resynchronizing the Data with the Primary Host 46
Updating a Warm Standby Sybase ASE 12.5 Database 47

Chapter 6. Oracle Database Recovery 49
Creating Storage Checkpoints 50
Rolling Back a Database 50
Script to Initiate Online Off-Host Oracle Database Backup 55
Script to Put Oracle Database into Hot Backup Mode 57
Script to Quiesce Sybase ASE Database 58
Script to Suspend I/O for a DB2 Database 59
Script to End Oracle Database Hot Backup Mode 60
Script to Release Sybase ASE Database from Quiesce Mode 61
Script to Resume I/O for a DB2 Database 62
Script to Perform Off-Host Backup 63
Script to Create an Off-Host Replica Oracle Database 64
Script to Complete, Recover and Start Replica Oracle Database 66
Script to Start Replica Sybase ASE Database 68
Text Control File for Original Production Database 72
SQL Script to Create Control File 73

Initialization File for Original Production Database 74
Initialization File for Replica Oracle Database 75

Index 77

Preface

Introduction

The purpose of this guide is to demonstrate how to use VERITAS FlashSnap™ to implement point-in-time copy solutions on enterprise systems. FlashSnap offers you flexible solutions for the efficient management of multiple point-in-time copies of your data, and for reducing resource contention on your business-critical servers.

Note This guide supersedes the *VERITAS Off-Host Processing Using FastResync Administrator's Guide.*

Audience and Scope

The *VERITAS® Point-In-Time Copy Solutions Administrator's Guide* provides information about how to implement solutions for online backup of databases and cluster-shareable file systems, for decision support on enterprise systems, and for Storage Rollback of databases to implement fast database recovery.

This guide is intended for experienced system administrators responsible for installing, configuring, and maintaining high-availability clustered systems under the control of VERITAS software.

This guide assumes that you have a good understanding of the following topics:

- The UNIX (AIX, HP-UX, Linux or Solaris) operating system.
- UNIX (AIX, HP-UX, Linux or Solaris) system administration.
- Cluster hardware and its configuration in enterprise installations (for scenarios that use clusters).
- Volume management.
- Configuration and administration of DB2, Oracle or Sybase databases (for operating systems and scenarios that use these databases).

Organization

This guide is organized as follows:

- Point-In-Time Copy Solutions
- Setting up Volume Snapshot Mirrors
- Online Database Backup
- Off-Host Cluster File System Backup
- Decision Support
- Oracle Database Recovery
- Files and Scripts for Sample Scenarios
- Preparing a Replica Oracle Database

Related Documents

The following documents provide more information related to the installation, configuration and administration of the products described in this guide:

- *VERITAS NetBackup BusinesServer Installation Guide*
- *VERITAS NetBackup BusinesServer System Administrator's Guide*
- *VERITAS Cluster File System Installation and Configuration Guide*
- *VERITAS Database Edition for DB2 Database Administrator's Guide*
- *VERITAS Database Edition for DB2 Installation and Configuration Guide*
- *VERITAS Database Edition for Oracle Database Administrator's Guide*
- *VERITAS Database Edition for Oracle Installation and Configuration Guide*
- *VERITAS Database Edition for Sybase Database Administrator's Guide*
- *VERITAS Database Edition for Sybase Installation and Configuration Guide*
- *VERITAS File System Administrator's Guide*
- *VERITAS File System Installation Guide*
- *VERITAS NetBackup DataCenter Installation Guide*
- *VERITAS NetBackup DataCenter System Administrator's Guide*
- *VERITAS NetBackup for Oracle ServerFree Agent System Administrator's Guide*
- *VERITAS NetBackup ServerFree Agent System Administrator's Guide*
- *VERITAS Volume Manager Administrator's Guide*
- *VERITAS Volume Manager Installation Guide*

Conventions

The following table describes the typographic conventions used in this guide.

Typeface	Usage	Examples
`monospace`	Computer output, file contents, files, directories, software elements such as command options, function names, and parameters	Read tunables from the `/etc/vx/tunefstab` file. See the `ls`(1) manual page for more information.
italic	New terms, book titles, emphasis, variables to be replaced by a name or value	See the *User's Guide* for details. The variable *ncsize* determines the value of...
`monospace` (**`bold`**)	User input; the "#" symbol indicates a command prompt	# **`mount -F vxfs /h/filesys`**
`monospace` ***(bold and italic)***	Variables to be replaced by a name or value in user input	# **`mount -F`** ***`fstype mount_point`***

Symbol	Usage	Examples
%	C shell prompt	
$	Bourne / Korn / Bash shell prompt	
#	Superuser prompt (all shells)	
\	Continued input on the following line	# **`mount -F vxfs`** \ **`/h/filesys`**
[]	In a command synopsis, brackets indicates an optional argument	`ls [ -a ]`
|	In a command synopsis, a vertical bar separates mutually exclusive arguments	`mount [suid | nosuid ]`

Getting Help

If you have any comments or problems with the VERITAS products, contact the VERITAS Technical Support:

- U.S. and Canadian Customers: 1-800-342-0652
- International Customers: +1 (650) 527-8555
- Email: `support@veritas.com`

For license information (U.S. and Canadian Customers):

- Phone: 1-925-931-2464
- Email: license@veritas.com
- Fax: 1-925-931-2487

For software updates:

- Email: swupdate@veritas.com

For additional technical support information, such as TechNotes, product alerts, and hardware compatibility lists, visit the VERITAS Technical Support Web site at:

- http://support.veritas.com

For information about VERITAS products, VERITAS Education Services and VPro Consulting Services, visit the VERITAS Web site at:

- http://www.veritas.com

Point-In-Time Copy Solutions 1

This chapter introduces the point-in-time copy solutions that you can implement using the VERITAS FlashSnap™ technology.

Note To implement the Point-In-Time Copy solutions presented in this document, a valid license for VERITAS FlashSnap must be present on all the systems to which the solutions are applied.

FlashSnap offers a flexible and efficient means of managing business critical data. It allows you to capture an online image of actively changing data at a given instant: a *point-in-time copy*. You can perform system backup, upgrade and other maintenance tasks on point-in-time copies while providing continuous availability of your critical data. If required, you can offload processing of the point-in-time copies onto another host to avoid contention for system resources on your production server.

Two kinds of point-in-time copy solution are supported by the FlashSnap license:

- Volume-level solutions are made possible by the Persistent FastResync and Disk Group Split/Join features of VERITAS Volume Manager™ 3.5 . These features are suitable for implementing solutions where the I/O performance of the production server is critical.

  The Persistent FastResync and Disk Group Split/Join features are described in "FastResync of Volume Snapshots" and "Disk Group Split/Join" on page 8.

- File system-level solutions use the Storage Checkpoints™ feature of VERITAS File System™ 3.5. Storage Checkpoints are suitable for implementing solutions where storage space is critical for:

  - File systems that contain a small number of mostly large files.
  - Application workloads that change a relatively small proportion of file system data blocks (for example, web server content and some databases).
  - Applications where multiple writable copies of a file system are required for testing or versioning.

  The Storage Checkpoints feature is described in "Storage Checkpoints" on page 8.

Applications of Point-in-Time Copy Solutions

The following typical activities are suitable for Point-In Time Copy solutions implemented using VERITAS FlashSnap:

- *Data Backup*—Many enterprises require 24 x 7 data availability. They cannot afford the downtime involved in backing up critical data offline. By taking snapshots of your data, and backing up from these snapshots, your business-critical applications can continue to run without extended down time or impacted performance.
- *Decision Support Analysis and Reporting*—Operations such as decision support analysis and business reporting may not require access to real-time information. You can direct such operations to use a replica database that you have created from snapshots, rather than allow them to compete for access to the primary database. When required, you can quickly resynchronize the database copy with the data in the primary database.
- *Testing and Training*—Development or service groups can use snapshots as test data for new applications. Snapshot data provides developers, system testers and QA groups with a realistic basis for testing the robustness, integrity and performance of new applications.
- *Database Error Recovery*—Logic errors caused by an administrator or an application program can compromise the integrity of a database. You can recover a database more quickly by restoring the database files by using Storage Checkpoints or a snapshot copy than by full restoration from tape or other backup media.

Note To provide continuity of service in the event of hardware failure in a cluster environment, you can use most point-in-time copy solutions in conjunction with the high availability cluster functionality of VERITAS SANPoint Foundation Suite™ HA or VERITAS Database Edition™ / HA. An exception is Storage Checkpoints, which cannot be used with VERITAS Cluster File System™ 3.5 (CFS).

Point-In-Time Copy Scenarios

Point-in-time copies of volumes allow you to capture an image of a database or file system at a selected instant for use in applications such as backups, decision support, reporting, and development testing.

Point-in-time copy solutions may additionally be configured to use off-host processing to remove much of the performance overhead on a production system.

The following chapters describe how you can use FlashSnap to implement regular online backup of database and cluster file system volumes, to set up a replica of a production database for decision support:

- Online Database Backup
- Off-Host Cluster File System Backup
- Decision Support

Three types of point-in-time copy solution are considered in this document:

- Primary host solutions where the copy is processed on the same system as the active data. See “Implementing Point-In Time Copy Solutions on a Primary Host” on page 9 for more information.
- Off-host solutions where the copy is processed on a different system from the active data. If implemented correctly, such solutions have almost no impact on the performance of the primary production system. See “Implementing Off-Host Point-In-Time Copy Solutions” on page 11 for more information.
- Using Storage Checkpoints to quickly roll back a database instance to an earlier point in time. See “Oracle Database Recovery” on page 49 for more information.

VERITAS Software Used in Point-In-Time Copy Scenarios

This guide provides a number of example scenarios that illustrate how to implement point-in-time copy solutions. The following table shows the VERITAS products that may be used with a VERITAS FlashSnap license to provide the required functionality in different environments:

Environment	Database server	Other applications
Standalone primary host	VERITAS Database Edition	VERITAS Foundation Suite™
Cluster without automatic failover	VERITAS Database Edition	VERITAS SANPoint Foundation Suite
Cluster with automatic failover	VERITAS Database Edition / HA	VERITAS SANPoint Foundation Suite HA

VERITAS Database Edition is required if you want to use features such as VERITAS Quick I/O™, VERITAS Extension for Oracle Disk Manager (ODM), VERITAS QuickLog™, Storage Checkpoints and management interfaces to enhance database performance and manageability.

The VERITAS Clustering Functionality for VxVM (CVM) and VERITAS Cluster File System 3.5 features of VERITAS SANPoint Foundation Suite 3.5 and VERITAS SANPoint Foundation Suite 3.5 HA allow you to share data within a cluster.

The HA versions of VERITAS SANPoint Foundation Suite and VERITAS Database Edition use VERITAS Cluster Server™ 3.5 to allow you to configure automated application and storage failover to provide continuous availability of service.

Note If you use both VERITAS SANPoint Foundation Suite and VERITAS Database Edition (HA or non-HA versions) with a cluster, be aware that Storage Checkpoints, Cached Quick I/O and QuickLog are not supported for use with VERITAS Cluster File System 3.5. However, Quick I/O is supported for use with VERITAS Cluster File System 3.5.

The following non-cluster specific components are used in the sample scenarios:

- VERITAS Volume Manager 3.5 (VxVM) is a disk management subsystem that supports disk striping, disk mirroring, and simplified disk management for improved data availability and superior performance. The FlashSnap license enables the use of the Persistent FastResync and Disk Group Split/Join features of VxVM.
- VERITAS File System 3.5 (VxFS) is a high-performance, fast-recovery file system that is optimized for business-critical database applications and data-intensive workloads. VxFS offers online administration, letting you perform most frequently scheduled maintenance tasks (including online backup, resizing, and file system changes) without interrupting data or system availability. The FlashSnap license enables the use of the Storage Checkpoints feature of VxFS.

You can also use the following cluster-specific components with the sample scenarios where required:

- VERITAS Cluster Server™ 3.5(VCS) is a high-availability (HA) solution for cluster configurations. VCS monitors systems and application services, and restarts services on a different cluster node (failover) in the event of either hardware or software failure. It also allows you to perform general administration tasks such as making nodes join or leave a cluster.

Note On HP-UX systems, MC/ServiceGuard may be configured as the cluster monitor.

- VERITAS Clustering Functionality for VxVM (CVM) allows multiple hosts to simultaneously access and manage a given set of disks that are under the control of VERITAS Volume Manager.
- VERITAS Cluster File System 3.5 (CFS) allows cluster nodes to share access to the same VxFS file system. CFS is especially useful for sharing read-intensive data between cluster nodes.

If you require a backup solution, the following VERITAS software is recommended:

- VERITAS NetBackup™ DataCenter provides mainframe-class data protection for corporate data centers. NetBackup DataCenter allows you to manage all aspects of backup and recovery, and allows consistent backup policies to be enforced across your organization. Optional NetBackup ServerFree Agents enhance NetBackup DataCenter to provide data protection for frozen image data such as VxVM snapshot mirrors. They can also be used to offload backup processing to NetBackup media servers or third-party copy devices over Fibre Channel networks.
- VERITAS NetBackup BusinesServer™ provides protection for small to medium-size server installations. It does not provide integrated support for backing up VxVM snapshot mirrors or for offloading backup processing. However, you can use it to back up snapshot volumes that have been created from snapshot mirrors.

For more information about installing and configuring these products see the following documentation:

- *VERITAS Cluster File System Installation and Configuration Guide*
- *VERITAS Database Edition for DB2 Installation and Configuration Guide*
- *VERITAS Database Edition for Oracle Installation and Configuration Guide*
- *VERITAS Database Edition for Sybase Installation and Configuration Guide*
- *VERITAS NetBackup DataCenter Installation Guide*
- *VERITAS NetBackup DataCenter System Administrator's Guide*
- *VERITAS NetBackup ServerFree Agent System Administrator's Guide*
- *VERITAS NetBackup for Oracle ServerFree Agent System Administrator's Guide*
- *VERITAS NetBackup BusinesServer Installation Guide*
- *VERITAS NetBackup BusinesServer System Administrator's Guide*
- *VERITAS File System Administrator's Guide*
- *VERITAS File System Installation Guide*
- *VERITAS Volume Manager Administrator's Guide*
- *VERITAS Volume Manager Installation Guide*

FastResync of Volume Snapshots

VERITAS Volume Manager allows you to take multiple snapshots of your data at the level of a volume. A snapshot volume contains a stable copy of a volume's data at a given moment in time that you can use for online backup or decision support. If FastResync is enabled on a volume, VxVM uses a *FastResync map* to keep track of which blocks are updated in the volume and in the snapshot. If the data in one mirror is not updated for some reason, it becomes out-of-date, or *stale*, with respect to the other mirrors in the volume. The presence of the FastResync map means that only those updates that the mirror has missed need be reapplied to resynchronize it with the volume. A full, and thereby much slower, resynchronization of the mirror from the volume is unnecessary.

Two forms or FastResync may be configured on a volume: Persistent FastResync and Non-Persistent FastResync.

- Non-Persistent FastResync stores its change maps in memory. If a system is rebooted, the information in the map is not preserved, so a full resynchronization is required on snapback. This limitation can be overcome for volumes in cluster-shareable disk groups, provided that at least one of the nodes in the cluster remained running to preserve the FastResync map in its memory.
- Persistent FastResync uses disk storage to ensure that FastResync maps survive both system and cluster crashes. Persistent FastResync is required to implement the solutions described in this document.

When snapshot volumes are reattached to their original volumes, FastResync allows the snapshot data to be quickly refreshed and re-used. If Persistent FastResync is enabled on a volume in a private disk group, such incremental resynchronization can happen even if the host is rebooted.

Persistent FastResync can track the association between volumes and their snapshot volumes after they are moved into different disk groups. After the disk groups are rejoined, Persistent FastResync allows the snapshot plexes to be quickly resynchronized. Non-Persistent FastResync cannot be used for this purpose.

For more information, see the *VERITAS Volume Manager Administrator's Guide*.

Disk Group Split/Join

One or more volumes, such as snapshot volumes, can be split off into a separate disk group and deported. They are then ready for importing on another host that is dedicated to off-host processing. This host need not be a member of a cluster but it must have access to the disks on which the volumes are configured. At a later stage, the disk group can be deported, re-imported, and joined with the original disk group, or with a different disk group.

For more information, see the *VERITAS Volume Manager Administrator's Guide.*

Storage Checkpoints

A Storage Checkpoint is a persistent image of a file system at a given instance in time. Storage Checkpoints use a *copy-on-write* technique to reduce I/O overhead by identifying and maintaining only those file system blocks that have changed since a previous Storage Checkpoint was taken. Storage Checkpoints have the following important features:

- Storage Checkpoints persist across system reboots and crashes.
- A Storage Checkpoint can preserve not only file system metadata and the directory hierarchy of the file system, but also user data as it existed when the Storage Checkpoint was taken.
- After creating a Storage Checkpoint of a mounted file system, you can continue to create, remove, and update files on the file system without affecting the image of the Storage Checkpoint.
- Unlike file system snapshots, Storage Checkpoints are writable.
- To minimize disk space usage, Storage Checkpoints use available free space in the file system.

Storage Checkpoints and the Storage Rollback feature of VERITAS Database Edition *for Oracle* enable rapid recovery of databases from logical errors such as database corruption, missing files and dropped table spaces. You can mount successive Storage Checkpoints of a database to locate the error, and then roll back the database to a Storage Checkpoint before the problem occurred. For more information, see "Oracle Database Recovery" on page 49 and the *VERITAS Database Edition Database Administrator's Guide.*

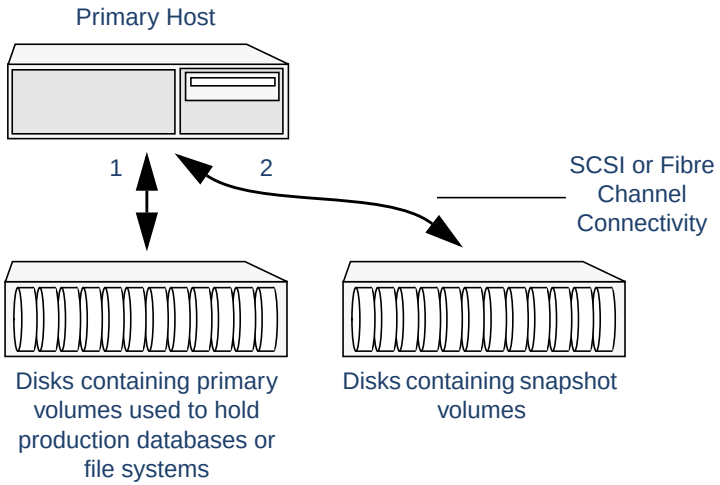
VERITAS NetBackup for Oracle Advanced BLI Agent uses Storage Checkpoints to enhance the speed of backing up Oracle databases. For more information, see the *VERITAS NetBackup for Oracle Advance BLI Agent System Administrator's Guide.*

For more information about the implementation of Storage Checkpoints, see the *VERITAS File System Administrator's Guide.*

Implementing Point-In Time Copy Solutions on a Primary Host

The figure "Example Point-In-Time Copy Solution on a Primary Host" shows the suggested arrangement for implementing solutions where the primary host is used.

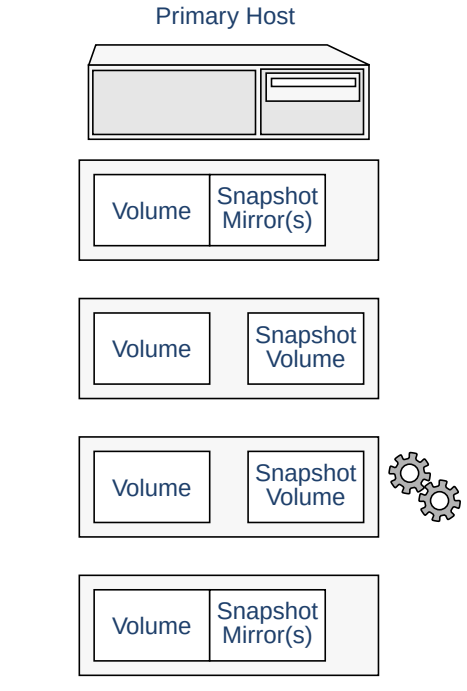
Example Point-In-Time Copy Solution on a Primary Host



In this setup, it is recommended that separate paths (shown as 1 and 2) from separate controllers be configured to the disks containing the primary volumes and the snapshot volumes. This avoids contention for disk access but the primary host's CPU, memory and I/O resources are more heavily utilized when the processing application is run.

The figure "Using Snapshots and FastResync to Implement Point-In-Time Copy Solutions on a Primary Host" illustrates the steps that are needed to set up the processing solution on the primary host. Note that the Disk Group Split/Join functionality is not used.

Using Snapshots and FastResync to Implement Point-In-Time Copy Solutions on a Primary Host

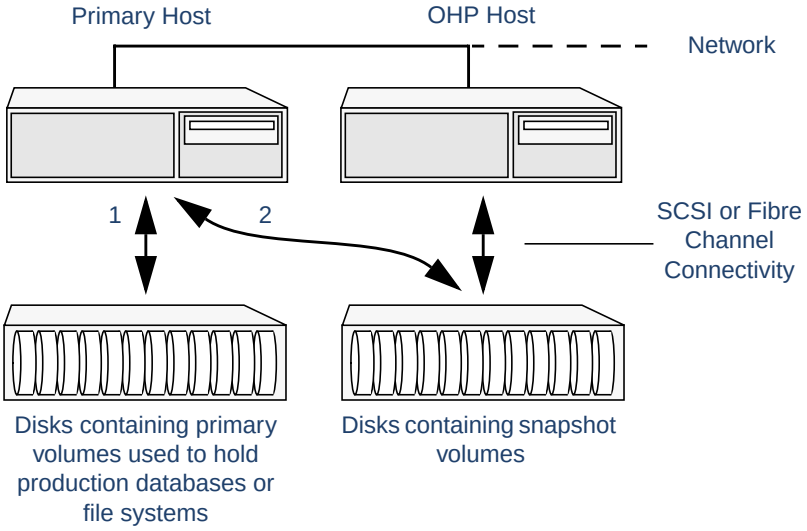


Note To back up a file system that does not contain an active database, you can use VERITAS NetBackup ServerFree Agent to combine steps 2 through 4 into a single operation. This additionally provides the ability to offload backup processing onto NetBackup media servers or third-party copy devices. An alternative approach for backing up Oracle databases is to use the VERITAS NetBackup for Oracle Advanced BLI Agent, which uses the Storage Checkpoint feature of VxFS 3.5. See the *System Administrator's Guides* that accompany these products for full details.

Implementing Off-Host Point-In-Time Copy Solutions

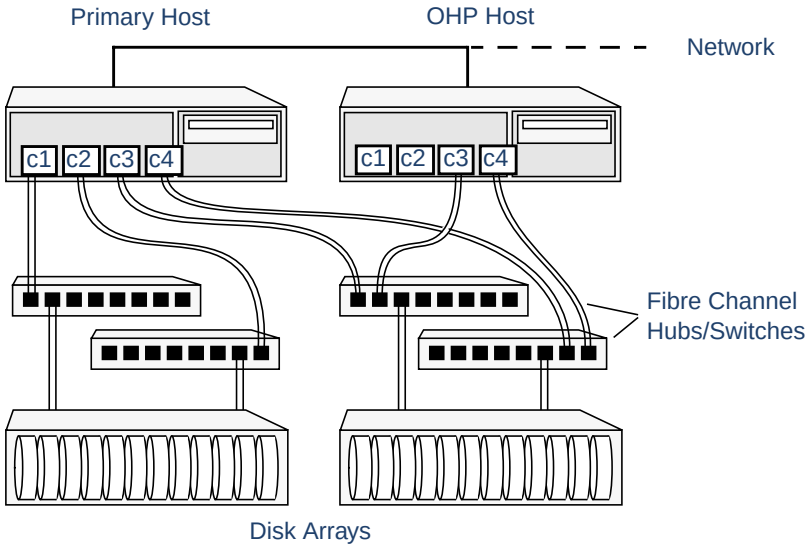
As shown in "Example Implementation of an Off-Host Point-In-Time Copy Solution" on page 11, by accessing snapshot volumes from a lightly loaded host (shown here as the *OHP host*), CPU- and I/O-intensive operations for online backup and decision support do not degrade the performance of the primary host that is performing the main production activity (such as running a database). Also, if you place the snapshot volumes on disks that are attached to host controllers other than those for the disks in the primary volumes, it is possible to avoid contending with the primary host for I/O resources. To implement this, paths 1 and 2 shown in the following figures should be connected to different controllers.

Example Implementation of an Off-Host Point-In-Time Copy Solution



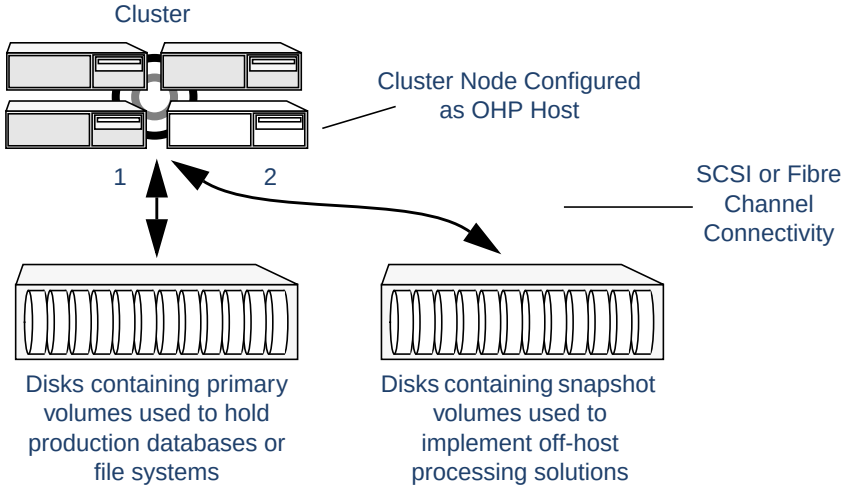
The figure "Example Connectivity for Off-Host Solution Using Redundant-Loop Access" on page 12 gives an example of how you might achieve such connectivity using Fibre Channel technology with 4 Fibre Channel controllers in the primary host. This layout uses redundant-loop access to deal with the potential failure of any single component in the path between a system and a disk array.

Example Connectivity for Off-Host Solution Using Redundant-Loop Access



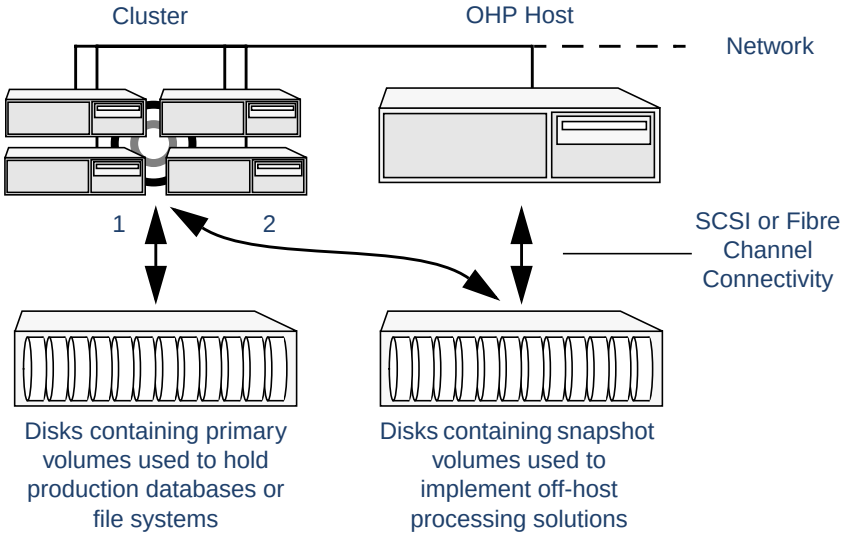
"Example Implementation of an Off-Host Point-In-Time Copy Solution Using a Cluster Node" shows how off-host processing might be implemented in a cluster by configuring one of the cluster nodes as the OHP node.

Example Implementation of an Off-Host Point-In-Time Copy Solution Using a Cluster Node



Alternatively, the OHP node could be a separate system that has a network connection to the cluster, but which is not is cluster node and which is not connected to the cluster's private network. This arrangement is illustrated in "Example implementation of an Off-Host Point-In-Time Copy Solution Using a Separate OHP Host."

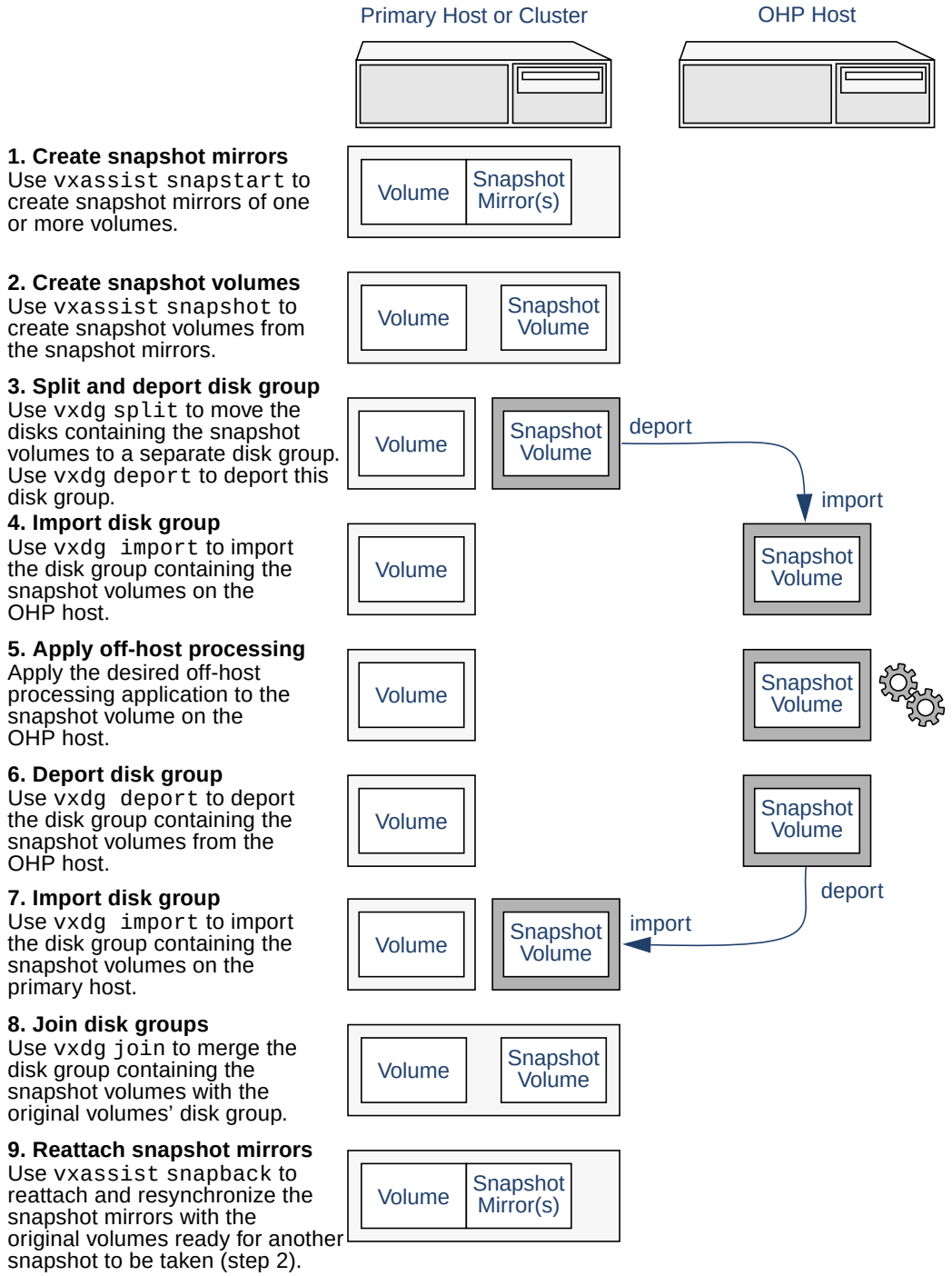
Example implementation of an Off-Host Point-In-Time Copy Solution Using a Separate OHP Host



Note For off-host processing, the example scenarios in this document assume that a separate OHP host is dedicated to the backup or decision support role. For clusters, it may be simpler to configure an OHP host that is not a member of the cluster.

"Using VERITAS FlashSnap to implement Off-Host Processing Solutions" on page 14 illustrates the steps that are needed to set up the processing solution on the primary host. Disk Group Split / Join is used to split off snapshot volumes into a separate disk group that is imported on the OHP host.

Using VERITAS FlashSnap to implement Off-Host Processing Solutions



Choices for Snapshot Resynchronization

When a snapshot volume is reattached to its original volume within a shared disk group, there are two choices for resynchronizing the data in the volume:

- *Resynchronize the snapshot from the original volume*—updates the snapshot with data from the primary volume that has changed since the snapshot was taken. The snapshot is then again ready to be taken for the purposes of backup or decision support.
- *Resynchronize the original volume from the snapshot*—updates the original volume with data from the snapshot volume that has changed since the snapshot was taken. This may be necessary to restore the state of a corrupted database or file system, or to implement upgrades to production software, and is usually much quicker than using alternative approaches such as full restoration from backup media.

Setting up Volume Snapshot Mirrors 2

This chapter describes how to prepare snapshots of the volumes that you want to back up, or that you want to use for decision support or reporting.

For more information about volume snapshots and FastResync, see the *VERITAS Volume Manager Administrator's Guide.*

Preparing Volumes for Taking Snapshots

To prepare volumes for use in implementing point-in-time copy solutions, create snapshot mirrors of them as described in "Creating a Snapshot Mirror of a Volume" on page 20. If either Non-Persistent FastResync is currently configured on the volume, or a data change object (DCO) and DCO volume are not associated with the volume, add a data change object (DCO) and DCO volume to the volume as described in the section "Adding a DCO and DCO Volume" on page 18.

To check whether a DCO and DCO volume are associated with a volume, use the following command:

```
# vxprint [-g diskgroup] -F%hasdcolog volume
```

This returns `on` if a DCO and DCO volume are associated with a volume; otherwise, it returns `off`.

To list all volumes on which Non-Persistent FastResync is enabled, use the following command:

```
# vxprint [-g diskgroup] -F "%name" \
    -e "v_fastresync=on && !v_hasdcolog"
```

To list all volumes on which Persistent FastResync is enabled, use the following command:

```
# vxprint [-g diskgroup] -F "%name" -e "v_fastresync=on \
  && v_hasdcolog"
```


The following table summarizes which volumes require the creation of snapshot mirrors for backup, decision support, and database error recovery.

Creation of Snapshot Mirrors

Point-In-Time Copy Application	Create Snapshot Mirrors for Volumes containing...
Online Database Backup	VxFS file systems for database datafiles to be backed up.
Off-Host Cluster File System Backup	VxFS cluster file systems to be backed up.
Decision Support	VxFS file systems for database datafiles to be replicated.

Caution To avoid data inconsistencies, do not use the same snapshot with different point-in-time copy applications. If you require snapshot mirrors for more than one application, configure at least one snapshot mirror that is dedicated to each application.

Adding a DCO and DCO Volume

Caution If the existing volume was created before release 3.2 of VxVM, and it has any attached snapshot plexes or it is associated with any snapshot volumes, follow the procedure given in the section “Enabling Persistent FastResync on Existing Volumes with Associated Snapshots” in the “Administering Volumes” chapter of the *VERITAS Volume Manager Administrator’s Guide*. The procedure given in this section assumes that neither snapshot plexes nor snapshot volumes are associated with the volumes.

To put Persistent FastResync into effect for a volume, a Data Change Object (DCO) and DCO volume must first be associated with that volume.

To add a DCO object and DCO volume to an existing volume (which may already have dirty region logging (DRL) enabled), use the following procedure:

1. Ensure that the disk group containing the existing volume has been upgraded to at least version 90. Use the following command to check the version of a disk group:

   # **vxprint -l *diskgroup* | egrep ’version:’**

To upgrade a disk group to version 90, use the following command:

```
# vxdg -T 90 upgrade diskgroup
```

2. Use the following command to turn off Non-Persistent FastResync on the original volume if it is currently enabled:

```
# vxvol [-g diskgroup] set fastresync=off volume
```

If you are uncertain about which volumes have Non-Persistent FastResync enabled, use the following command to obtain a listing of such volumes:

```
# vxprint [-g diskgroup] -F "%name" \
   -e "v_fastresync=on && !v_hasdcolog"
```

3. Use the following command to add a DCO and DCO volume to the existing volume:

```
# vxassist [-g diskgroup[ addlog volume logtype=dco \
  [ndcolog=number] [dcologlen=size] [storage_attributes]
```

The default number of plexes in the mirrored DCO volume is 2. You can use the `ndcolog` attribute to specify a different *number*. It is recommended that you configure as many DCO plexes as there are existing data and snapshot plexes in the volume. For example, specify `ndcolog=3` when adding a DCO to a 3-way mirrored volume.

The default size of each plex is 132 blocks. You can use the `dcologlen` attribute to specify a different *size*. If specified, the size of the plex must be a integer multiple of 33 blocks from 33 up to a maximum of 2112 blocks.

If the disks that contain volumes and their snapshots are to be moved into different disk groups, you must ensure that the disks that contain their DCO plexes can accompany them. You can use storage attributes to specify which disks to use for the DCO plexes. If possible, specify the same disks as those on which the volume is configured. For example, to add a DCO object and DCO volume with plexes on `disk5` and `disk6`, and a plex size of 264 blocks to the volume, `myvol`, use the following command:

```
# vxassist -g mydg addlog myvol logtype=dco dcologlen=264 \
  disk5 disk6
```

If required, you can use the `vxassist move` command to relocate DCO plexes to different disks. For example, the following command moves the plexes of the DCO volume for volume `vol1` from `disk3` and `disk4` to `disk7` and `disk8`:

```
# vxassist -g mydg move vol1_dcl !disk3 !disk4 disk7 disk8
```

When you have added a DCO object and DCO volume to a volume, you can enable Persistent FastResync on the volume as described in "Creating a Snapshot Mirror of a Volume" on page 20.

To view the details of the DCO object and DCO volume that are associated with a volume, use the `vxprint` command. The following is example `vxprint` output for the volume named `zoo` (the TUTIL0 and PUTIL0 columns are omitted for clarity):

```
TY  NAME         ASSOC      KSTATE    LENGTH    PLOFFS  STATE   ...
v   zoo          fsgen      ENABLED   1024      -       ACTIVE
pl  zoo-01       zoo        ENABLED   1024      -       ACTIVE
sd  c1t66d0s2-02 zoo-01     ENABLED   1024      0       -
pl  foo-02       zoo        ENABLED   1024      -       ACTIVE
sd  c1t67d0s2-02 zoo-02     ENABLED   1024      0       -
dc  zoo_dco      zoo        -         -         -       -
v   zoo_dcl      gen        ENABLED   132       -       ACTIVE
pl  zoo_dcl-01   zoo_dcl    ENABLED   132       -       ACTIVE
sd  c1t66d0s2-01 zoo_dcl-01 ENABLED   132       0       -
pl  zoo_dcl-02   zoo_dcl    ENABLED   132       -       ACTIVE
sd  c1t67d0s2-01 zoo_dcl-02 ENABLED   132       0       -
```

In this output, the DCO object is shown as `zoo_dco`, and the DCO volume as `zoo_dcl` with 2 plexes, `zoo_dcl-01` and `zoo_dcl-02`.

For more information, see "Considerations for Placing DCO Plexes" on page 21 and the `vxassist`(1M) manual page.

Creating a Snapshot Mirror of a Volume

To create a snapshot mirror of a volume, perform the following steps on the primary host (in a cluster, this is the master node):

1. Turn on FastResync tracking for the volume using the following command:

   # **`vxvol -g`** ***`diskgroup`*** **`set fastresync=on`** ***`volume`***

   For example, to turn on FastResync for the volume `dbase_vol` in the disk group `dbasedg`:

   # **`vxvol -g dbasedg set fastresync=on dbase_vol`**

2. Add a snapshot mirror to the volume on specified storage using the following command:

   # **`vxassist -g`** ***`diskgroup`*** **`-b snapstart [nmirror=`*****`N`*****`]`** ***`volume`*** **`\`**
   `storage_attributes`

   Use the `nmirror` attribute to create as many snapshot mirrors as you need for the snapshot volume. For a backup, you should usually only require the default of one.

If the disks that contain volumes and their snapshots are to be moved into different disk groups, you must ensure that the disks that contain their DCO plexes can accompany them. Use storage attributes to prevent the snapshot mirror being created on disks that are in use by the volume or its DCO volume. For example, to take a snapshot of the volume `dbase_vol` using disks attached to controller `c3` and excluding `disk3` and `disk4`:

```
# vxassist -g dbasedg -b snapstart dbase_vol ctlr:c3 !disk3 !disk4
```

3. Wait for the snapshot mirror to be created. The `-b` option to `vxassist` runs this process in the background. If you leave out this option, the `vxassist` command will block until the snapshot mirror is complete. If you run the command in the background, either use the `vxprint` command to check when the status of the snapshot changes to SNAPDONE, or use `vxassist snapwait` to wait for the snapshot to complete as shown here:

   ```
   # vxassist -g diskgroup snapwait volume
   ```

   For example, to wait for a snapshot to be taken of the volume `dbase_vol`:

   ```
   # vxassist -g dbasedg snapwait dbase_vol
   ```

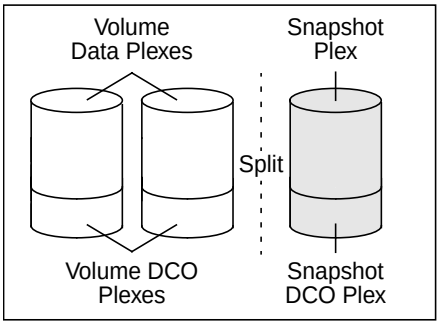
Considerations for Placing DCO Plexes

If you use the `vxassist` command or the graphical user interface to create a volume, or to enable Persistent FastResync on a volume, the DCO plexes are automatically placed on the same disks as the data plexes of the parent volume. When you move a snapshot volume to a different disk group, this ensures that its DCO volume automatically accompanies it. If you use the `vxassist addlog` command to set up a DCO, you must ensure that the disks that contain the plexes of the DCO volume accompany their parent volume during the move. Use the `vxprint` command on a volume to examine the configuration of its associated DCO volume.

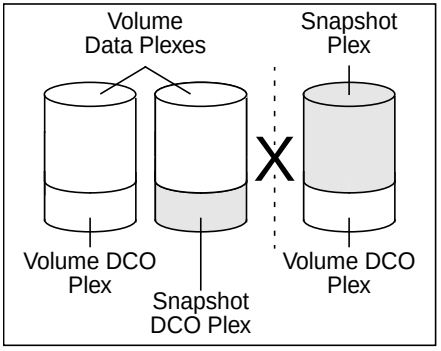
"Examples of Disk Groups That Can and Cannot be Split" on page 22 illustrates some instances in which it is not be possible to split a disk group because of the location of the DCO plexes.

For more information about relocating DCO plexes, see "Adding a DCO and DCO Volume" on page 18.

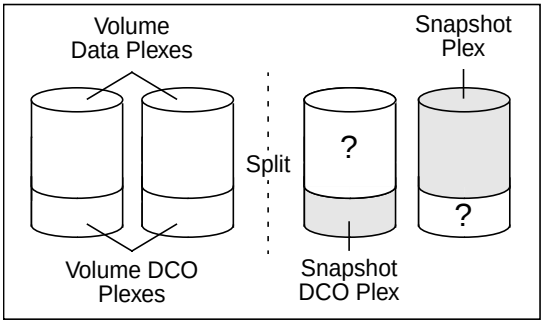
Examples of Disk Groups That Can and Cannot be Split



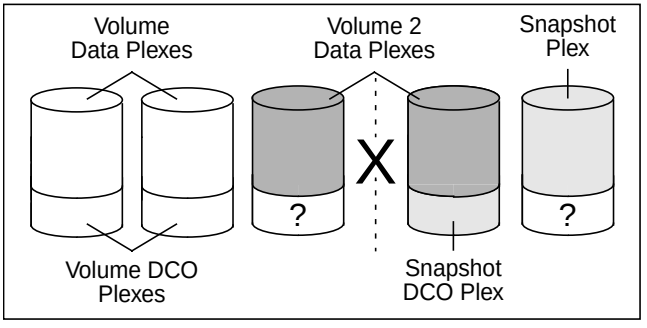
The disk group can be split as the DCO plexes are on the same disks as the data plexes and can therefore accompany their volumes.



The disk group cannot be split as the DCO plexes have been separated from their data plexes and so cannot accompany their volumes. One solution is to relocate the DCO plexes. In this example, it may be necessary to use an additional disk in the disk group as an intermediary to swap the plexes.



The disk group can be split as the DCO plexes can accompany their volumes even though they are on different disks. However, you may not wish the data in the portions of the disks marked “?” to be moved as well.



The disk group cannot be split as this would separate the disks that contain the data plexes of Volume 2. Possible solutions are to relocate the snapshot DCO plex to the disk containing the snapshot plex, or to another suitable disk that can be moved.

Online Database Backup 3

Online backup of a database can be implemented by configuring either the primary host or a dedicated separate host to perform the backup operation on snapshot mirrors of the primary host's database.

Two backup methods are described in the following sections:

- Making a Backup of an Online Database on the Same Host
- Making an Off-Host Backup of an Online Database

Note All commands require superuser (`root`) or equivalent privileges, except where it is explicitly stated that a command must be run by the database administrator.

For more information about using snapshots to back up DB2, Oracle and Sybase databases, see the chapter "Using Snapshots for Database Backup" in the *VERITAS Database Edition Database Administrator's Guide*.

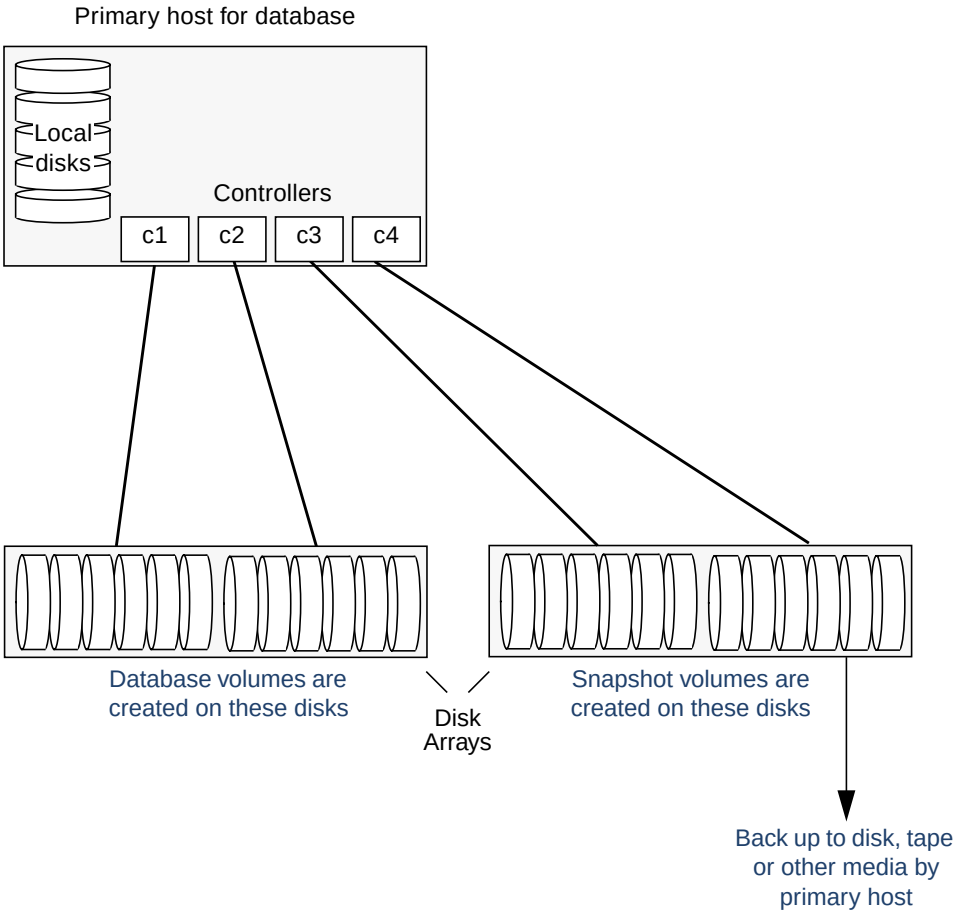
The following sections include sample scripts:

- "Script to Initiate Online Off-Host Oracle Database Backup" on page 55
- "Script to Put Oracle Database into Hot Backup Mode" on page 57
- "Script to Quiesce Sybase ASE Database" on page 58
- "Script to Suspend I/O for a DB2 Database" on page 59
- "Script to End Oracle Database Hot Backup Mode" on page 60
- "Script to Release Sybase ASE Database from Quiesce Mode" on page 61
- "Script to Resume I/O for a DB2 Database" on page 62
- "Script to Perform Off-Host Backup" on page 63

Making a Backup of an Online Database on the Same Host

As illustrated in "Example System Configuration for Off-Host Database Backup" on page 27, the primary database volumes to be backed up, `dbase_vol` and `dbase_logs`, are configured on disks attached to controllers `c1` and `c2`, and the snapshots are created on disks attached to controllers `c3` and `c4`.

Example System Configuration for Database Backup on the Primary Host



Note It is assumed that you have already configured snapshot mirrors for the volumes containing the file systems for the datafiles to be backed up as described in "Setting up Volume Snapshot Mirrors" on page 17. For an Oracle database, it is not necessary to create snapshots of the volumes containing the file systems for the redo log volumes or archived logs.

To make a backup of an online database on the same host:

1. If the volumes to be backed up contain database tables in file systems, suspend updates to the volumes:
   - DB2 provides the `write suspend` command to temporarily suspend I/O activity for a database. As the DB2 database administrator, use a script such as that shown in "Script to Suspend I/O for a DB2 Database" on page 59. Note that to allow recovery from any backups taken from snapshots, the database must be in LOGRETAIN RECOVERY mode.
   - Oracle supports online backup by temporarily suspending updates to the datafiles of the tablespaces, provided that the database is running in archive mode and the tablespaces are online. As the Oracle database administrator, put each tablespace into hot backup mode using a script such as that shown in "Script to Put Oracle Database into Hot Backup Mode" on page 57.
   - Sybase ASE from version 12.0 onward provides the Quiesce feature to allow temporary suspension of writes to a database. As the Sybase database administrator, put the database in quiesce mode by using a script such as that shown in "Script to Quiesce Sybase ASE Database" on page 58.
2. Make a snapshot volume, *snapvol*, using the following command:

   # **`vxassist -g` *`volumedg`* `snapshot [nmirror=`*`N`*`]` *`volume snapvol`* `...`**

   If required, use the `nmirror` attribute to specify the number of mirrors, *N*. in the snapshot volume. By default, a snapshot volume only contains a single plex.

   If a database spans more than one volume, specify all the volumes and their snapshot volumes on the same line, for example:

   # **`vxassist -g dbasedg snapshot vol1 svol1 vol2 svol2 vol3 svol3`**
3. If you temporarily suspended updates to volumes in step 1, release all the tablespaces or databases from suspend, hot backup or quiesce mode:
   - As the DB2 database administrator, use a script such as that shown in "Script to Resume I/O for a DB2 Database" on page 62.
   - As the Oracle database administrator, release all the tablespaces from hot backup mode using a script such as that shown in "Script to End Oracle Database Hot Backup Mode" on page 60.
   - As the Sybase database administrator, release the database from quiesce mode using a script such as that shown in "Script to Release Sybase ASE Database from Quiesce Mode" on page 61.

4. Back up the snapshot volume. If you need to remount the file system in the volume to back it up, first run `fsck` on the volume. The following are sample commands for checking and mounting a file system:

   ```
   # fsck -F vxfs /dev/vx/rdsk/snapvoldg/snapvol
   # mount -F vxfs /dev/vx/dsk/snapvoldg/snapvol mount_point
   ```

   Back up the file system at this point using a command such as `bpbackup` in VERITAS NetBackup. After the backup is complete, use the following command to unmount the file system.

   ```
   # unmount mount_point
   ```

5. Reattach the plexes of each snapshot volume to their original volume, and resynchronize their contents using the following command:

   ```
   # vxassist -g volumedg snapback [nmirror=N] snapvol
   ```

   Here the `nmirror` attribute specifies the number of mirrors in the snapshot volume to be re-attached.

   For example, to reattach `snap1_dbase_vol`:

   ```
   # vxassist -g dbasedg snapback snap1_dbase_vol
   ```

Repeat steps 1 through 5 each time that you need to back up the volume.

In some instances, such as recovering the contents of a corrupted volume, it may be useful to resynchronize a volume from its snapshot volume (which is used as a hot standby):

```
# vxassist -g diskgroup -o resyncfromreplica snapback snapvol
```

For example, to resynchronize the volume `dbase_vol` from its snapshot volume `snap2_dbase_vol`:

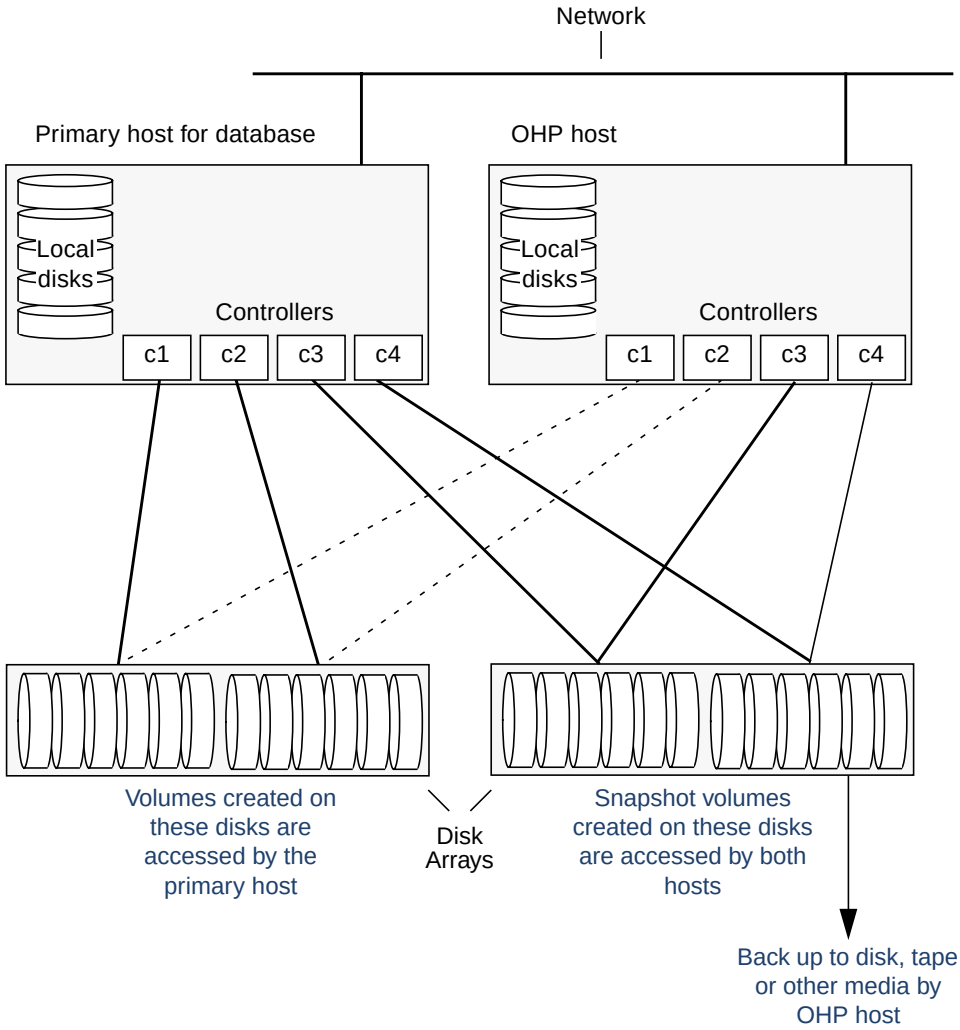
```
# vxassist -g dbasedg -o resyncfromreplica snapback \
  snap2_dbase_vol
```

Note You must shut down the database and unmount the file system that is configured on the original volume before attempting to resynchronize its contents from a snapshot.

Making an Off-Host Backup of an Online Database

As illustrated in “Example System Configuration for Off-Host Database Backup” on page 27, the primary database volumes to be backed up, `dbase_vol` and `dbase_logs`, are configured on disks attached to controllers `c1` and `c2`, and the snapshots are created on disks attached to controllers `c3` and `c4`. There is no requirement for the OHP host to have access to the disks that contain the primary database volumes.

Example System Configuration for Off-Host Database Backup



Note It is assumed that you have already configured snapshot mirrors for the volumes containing the file systems for the datafiles to be backed up as described in "Setting up Volume Snapshot Mirrors" on page 17. For an Oracle database, it is not necessary to create snapshots of the volumes containing the file systems for the redo log volumes or archived logs.

Note If the database is configured on volumes in a cluster-shareable disk group, it is assumed that the primary host for the database is the master node for the cluster. If the primary host is not also the master node, all VxVM operations on shared disk groups must be performed on the master node.

To make an off-host backup of an online database:

1. If the volumes to be backed up contain database tables in file systems, suspend updates to the volumes:
   - DB2 provides the `write suspend` command to temporarily suspend I/O activity for a database. As the DB2 database administrator, use a script such as that shown in "Script to Suspend I/O for a DB2 Database" on page 59. Note that to allow recovery from any backups taken from snapshots, the database must be in LOGRETAIN RECOVERY mode.
   - Oracle supports online backup by temporarily suspending updates to the datafiles of the tablespaces, provided that the database is running in archive mode and the tablespaces are online. As the Oracle database administrator, put each tablespace into hot backup mode using a script such as that shown in "Script to Put Oracle Database into Hot Backup Mode" on page 57.
   - Sybase ASE from version 12.0 onward provides the Quiesce feature to allow temporary suspension of writes to a database. As the Sybase database administrator, put the database in quiesce mode by using a script such as that shown in "Script to Quiesce Sybase ASE Database" on page 58.
2. On the primary host, take snapshots of the tablespace volumes using the following command:

   # **vxassist -g *volumedg* snapshot [nmirror=*N*] *volume snapvol* ...**

   If required, use the `nmirror` attribute to specify the number of mirrors, *N*. in the snapshot volume. By default, a snapshot volume only contains a single plex.

   If a database spans more than one volume, specify all the volumes and their snapshot volumes on the same line, for example:

   # **vxassist -g dbasedg snapshot vol1 svol1 vol2 svol2 vol3 svol3**
3. If you temporarily suspended updates to volumes in step 1, release all the tablespaces or databases from suspend, hot backup or quiesce mode:
   - As the DB2 database administrator, use a script such as that shown in "Script to Resume I/O for a DB2 Database" on page 62.
   - As the Oracle database administrator, release all the tablespaces from hot backup mode using a script such as that shown in "Script to End Oracle Database Hot Backup Mode" on page 60.

- As the Sybase database administrator, release the database from quiesce mode using a script such as that shown in "Script to Release Sybase ASE Database from Quiesce Mode" on page 61.

4. On the primary host, use the following command to split the disks containing the snapshot volumes into a separate disk group, *snapvoldg*, from the original disk group, *volumedg*:

   # **vxdg split *volumedg snapvoldg snapvol* ...**

5. On the primary host, deport the snapshot volumes' disk group using the following command:

   # **vxdg deport *snapvoldg***

6. On the OHP host where the backup is to be performed, use the following command to import the snapshot volumes' disk group:

   # **vxdg import *snapvoldg***

7. The snapshot volumes are initially disabled following the split. Use the following commands on the OHP host to recover and restart the snapshot volumes:

   # **vxrecover -g *snapvoldg* -m *snapvol* ...**
   # **vxvol -g *snapvoldg* start *snapvol* ...**

8. On the OHP host, back up the snapshot volumes. If you need to remount the file system in the volume to back it up, first run `fsck` on the volumes. The following are sample commands for checking and mounting a file system:

   # **fsck -F vxfs /dev/vx/rdsk/*snapvoldg*/*snapvol***
   # **mount -F vxfs /dev/vx/dsk/*snapvoldg*/*snapvol* *mount_point***

   Back up the file system at this point using a command such as `bpbackup` in VERITAS NetBackup. After the backup is complete, use the following command to unmount the file system.

   # **unmount *mount_point***

9. On the OHP host, use the following command to deport the snapshot volumes' disk group:

   # **vxdg deport *snapvoldg***

10. On the primary host, re-import the snapshot volumes' disk group using the following command:

    # **vxdg [-s] import *snapvoldg***

Note Specify the `-s` option if you are reimporting the disk group to be rejoined with a shared disk group in a cluster.

11. On the primary host, use the following command to rejoin the snapshot volumes's disk group with the original volumes' disk group:

    ```
    # vxdg join snapvoldg volumedg
    ```

12. The snapshot volume is initially disabled following the join. Use the following commands on the primary host to recover and restart a snapshot volume:

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

13. On the primary host, reattach the plexes of each snapshot volume to their original volume, and resynchronize their contents using the following command:

    ```
    # vxassist -g volumedg snapback [nmirror=N] snapvol
    ```

    Here the `nmirror` attribute specifies the number of mirrors in the snapshot volume to be re-attached.

    For example, to reattach `snap1_dbase_vol`:

    ```
    # vxassist -g dbasedg snapback snap1_dbase_vol
    ```

Repeat steps 1 through 13 each time that you need to back up the volume.

For an example of a script that uses this method, see "Script to Initiate Online Off-Host Oracle Database Backup" on page 55.

In some instances, such as recovering the contents of a corrupted volume, it may be useful to resynchronize a volume from its snapshot volume (which is used as a hot standby):

```
# vxassist -g diskgroup -o resyncfromreplica snapback \
  [nmirror=N] snapvol
```

For example, to resynchronize the volume `dbase_vol` from its snapshot volume `snap2_dbase_vol`:

```
# vxassist -g dbasedg -o resyncfromreplica snapback \
  snap2_dbase_vol
```

Note You must shut down the database and unmount the file system that is configured on the original volume before attempting to resynchronize its contents from a snapshot.

Off-Host Cluster File System Backup 4

VERITAS Cluster File System (CFS) allows cluster nodes to share access to the same file system. CFS is especially useful for sharing read-intensive data between cluster nodes.

Off-host backup of cluster file systems may be implemented by taking a snapshot of the volume containing the file system and performing the backup operation on a separate host.

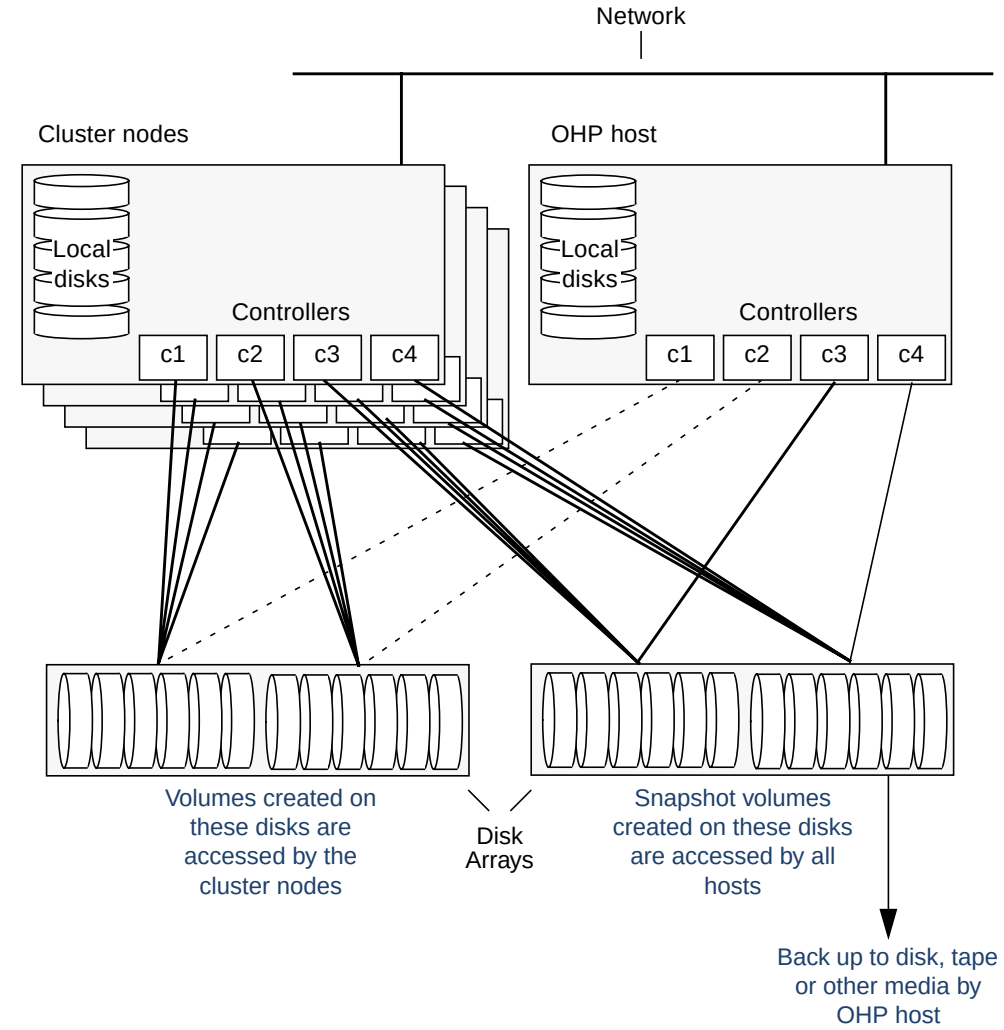
As illustrated in "System Configuration for Off-Host File System Backup Scenarios" on page 32, the primary volume that contains the file system to be backed up is configured on disks attached to controllers `c1` and `c2`, and the snapshots are created on disks attached to controllers `c3` and `c4`.

See "Mounting a File System for Shared Access" on page 33 for a description of how to mount a VxFS file system for shared access by the nodes of a cluster.

See "Using Off-Host Processing to Back Up Cluster File Systems" on page 33 for a description of how to perform off-host backups of cluster-shared file systems.

Note All commands require superuser (`root`) or equivalent privileges.

System Configuration for Off-Host File System Backup Scenarios



Mounting a File System for Shared Access

To mount a VxFS file system for shared access, use the following command on each cluster node where required:

```
# mount -F vxfs -o cluster /dev/vx/dsk/diskgroup/volume mount_point
```

For example, to mount the volume `cfs_vol` in the disk group `exampledg` for shared access on the mount point, `/mnt_pnt`:

```
# mount -F vxfs -o cluster /dev/vx/dsk/exampledg/cfs_vol /mnt_pnt
```

Using Off-Host Processing to Back Up Cluster File Systems

Note It is assumed that you have already configured snapshot mirrors for the volumes containing the file systems that are to be backed up as described in "Preparing Volumes for Taking Snapshots" on page 17.

To back up a snapshot of a mounted file system which has shared access, perform the following steps on the master node of the cluster:

1. Take a snapshot of the volume containing the file system using the following command:

   ```
   # vxassist -g diskgroup snapshot [nmirror=N] volume snapvol
   ```

   If required, use the `nmirror` attribute to specify the number of mirrors, *N*. in the snapshot volume. By default, a snapshot volume only contains a single plex.

   For example, to take a snapshot of the volume `cfs_vol` in the shared disk group `exampledg`:

   ```
   # vxassist -g exampledg snapshot cfs_vol snap_cfs_vol
   ```

2. On the master node, use the following command to split the snapshot volume into a separate disk group, *snapvoldg*, from the original disk group, *volumedg*:

   ```
   # vxdg split volumedg snapvoldg snapvol
   ```

   For example, to place the snapshot of the volume `cfs_vol` into the shared disk group `splitdg`:

   ```
   # vxdg split exampledg splitdg snap_cfs_vol
   ```

3. On the master node, deport the snapshot volume's disk group using the following command:

   ```
   # vxdg deport snapvoldg
   ```


For example, to deport the disk group `splitdg`:

```
# vxdg deport splitdg
```

4. On the OHP host where the backup is to be performed, use the following command to import the snapshot volume's disk group:

   ```
   # vxdg import snapvoldg
   ```

   For example, to import the disk group `splitdg`:

   ```
   # vxdg import splitdg
   ```

5. The snapshot volume is initially disabled following the split. Use the following commands on the OHP host to recover and restart the snapshot volume:

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

6. On the OHP host, use the following commands to check and *locally* mount the snapshot volume:

   ```
   # fsck -F vxfs /dev/vx/rdsk/diskgroup/volume
   # mount -F vxfs /dev/vx/dsk/diskgroup/volume mount_point
   ```

   For example, to check and mount the volume `snap_cfs_vol` in the disk group `exampledg` for shared access on the mount point, `/bak/mnt_pnt`:

   ```
   # fsck -F vxfs /dev/vx/rdsk/exampledg/snap_cfs_vol
   # mount -F vxfs /dev/vx/dsk/exampledg/snap_cfs_vol /bak/mnt_pnt
   ```

7. Back up the file system at this point using a command such as `bpbackup` in VERITAS NetBackup. After the backup is complete, use the following command to unmount the file system.

   ```
   # unmount mount_point
   ```

8. On the OHP host, use the following command to deport the snapshot volume's disk group:

   ```
   # vxdg deport snapvoldg
   ```

9. On the master node, re-import the snapshot volume's disk group as a shared disk group using the following command:

   ```
   # vxdg -s import snapvoldg
   ```

10. On the master node, use the following command to rejoin the snapshot volume's disk group with the original volume's disk group:

    ```
    # vxdg join snapvoldg volumedg
    ```

For example, to join disk group `splitdg` with `exampledg`:

```
# vxdg join splitdg exampledg
```

11. The snapshot volume is initially disabled following the join. Use the following commands on the primary host to recover and restart the snapshot volume:

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

12. When the backup is complete, use the following commands to unmount the snapshot volume and resynchronize it from the primary volume:

```
# umount mount_point
# vxassist -g diskgroup snapback [nmirror=N] snapvol
```

Here the `nmirror` attribute specifies the number of mirrors in the snapshot volume to be re-attached.

For example, to unmount and resynchronize `snap_cfs_vol`:

```
# umount /bak/mnt_pnt
# vxassist -g exampledg snapback snap_cfs_vol
```

The snapshot is now ready to be re-used for backup.

Caution Before attempting to resynchronize the snapshot from the original volume, shut down all applications that access a file system in the snapshot volume, and also unmount any such file system.

Repeat steps 1 through 12 each time that you need to back up the volume.

In some instances, such as recovering the contents of a corrupted volume, it may be useful to resynchronize a volume from its snapshot volume (which is used as a hot standby):

```
# vxassist -g diskgroup -o resyncfromreplica snapback \
  [nmirror=N] snapvol
```

For example, to resynchronize the volume `cfs_vol` from its snapshot volume `snap1_cfs_vol`:

```
# vxassist -g dbasedg -o resyncfromreplica snapback \
  snap1_cfs_vol
```

Note You must unmount the file system that is configured on the original volume before attempting to resynchronize its contents from a snapshot.

Decision Support 5

You can use snapshots of a primary database to create a replica of the database at a given moment in time. You can then implement decision support analysis and report generation operations that take their data from the database copy rather than from the primary database. The FastResync functionality of VERITAS Volume Manager (VxVM) allows you to quickly refresh the database copy with up-to-date information from the primary database. Reducing the time taken to update decision support data also lets you generate analysis reports more frequently.

Two methods are described for setting up a replica database for decision support:

- Creating a Replica Database on the Same Host
- Creating up an Off-Host Replica Database

Note All commands require superuser (`root`) or equivalent privileges, except where it is explicitly stated that a command must be run by the database administrator.

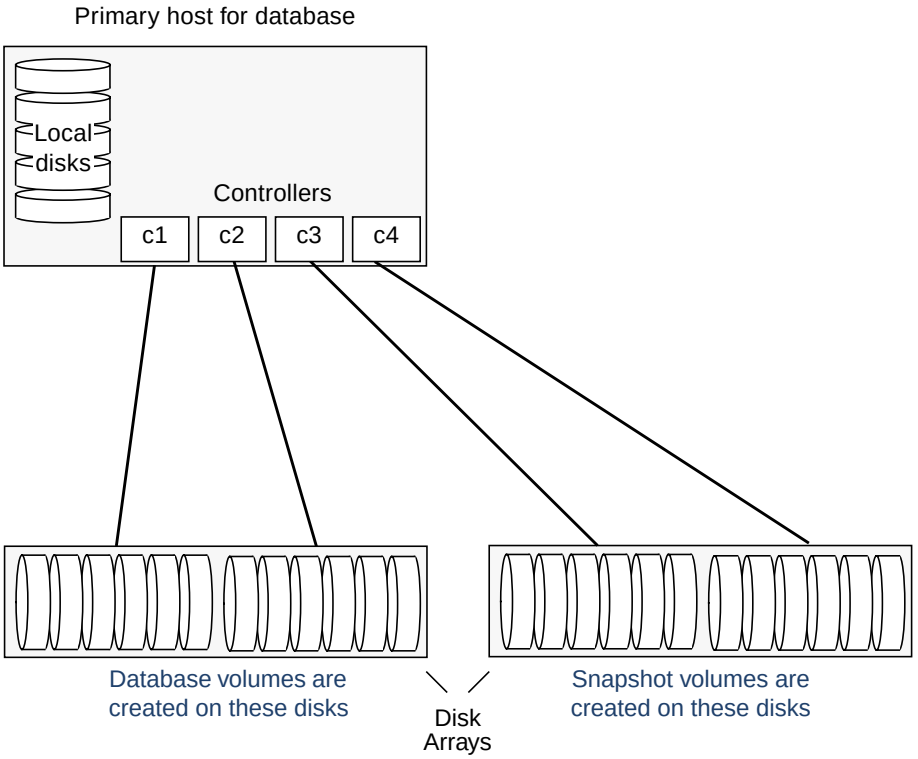
The following sections include sample scripts:

- "Script to Put Oracle Database into Hot Backup Mode" on page 57
- "Script to Quiesce Sybase ASE Database" on page 58
- "Script to Suspend I/O for a DB2 Database" on page 59
- "Script to End Oracle Database Hot Backup Mode" on page 60
- "Script to Release Sybase ASE Database from Quiesce Mode" on page 61
- "Script to Resume I/O for a DB2 Database" on page 62
- "Script to Create an Off-Host Replica Oracle Database" on page 64
- "Script to Complete, Recover and Start Replica Oracle Database" on page 66
- "Script to Start Replica Sybase ASE Database" on page 68

Creating a Replica Database on the Same Host

As illustrated in "Example System Configuration for Decision Support on the Primary Host", the primary database volumes to be backed up, `dbase_vol` and `dbase_logs`, are configured on disks attached to controllers `c1` and `c2`, and the snapshots are created on disks attached to controllers `c3` and `c4`.

Example System Configuration for Decision Support on the Primary Host



Note It is assumed that you have already configured snapshot mirrors for the database volumes to be replicated as described in "Preparing Volumes for Taking Snapshots" on page 17.

To set up a replica database to be used for decision support on the primary host:

1. If not already done, prepare the host to use the snapshot volume that contains the copy of the database tables. Set up any new database logs and configuration files that are required to initialize the database.

2. If the volumes to be backed up contain database tables in file systems, suspend updates to the volumes:

   - Oracle supports online backup by temporarily suspending updates to the datafiles of the tablespaces, provided that the database is running in archive mode and the tablespaces are online. As the Oracle database administrator, put each tablespace into hot backup mode using a script such as that shown in "Script to Put Oracle Database into Hot Backup Mode" on page 57.
   - Sybase ASE from version 12.0 onward provides the Quiesce feature to allow temporary suspension of writes to a database. As the Sybase database administrator, put the database in quiesce mode by using a script such as that shown in "Script to Quiesce Sybase ASE Database" on page 58.

     If you are using Sybase ASE 12.5, you can specify the `for external dump` clause to the `quiesce` command. This *warm standby* method allows you to update a replica database using transaction logs dumped from the primary database. See "Updating a Warm Standby Sybase ASE 12.5 Database" on page 47 for more information.

3. Take snapshots of the tablespace volumes using the following command:

   # **vxassist -g *volumedg* snapshot [nmirror=*N*] *volume snapvol* ...**

   If required, use the `nmirror` attribute to specify the number of mirrors in the snapshot volume. Note that there must be sufficient snapshot mirrors available in the volume to do this.

   For example, the following command takes a snapshot with 2 mirrors of the volume `dbase_vol` in the shared disk group, `dbasedg`:

   ```
   # vxassist -g dbasedg snapshot nmirror=2 dbase_vol \
     snap1_dbase_vol
   ```

   If a database spans more than one volume, specify all the volumes and their snapshot volumes on the same line, for example:

   ```
   # vxassist -g dbasedg snapshot vol1 svol1 vol2 svol2 vol3 svol3
   ```

4. If you temporarily suspended updates to volumes in step 2, release all the tablespaces or databases from suspend, hot backup or quiesce mode:

   - As the Oracle database administrator, release all the tablespaces from hot backup mode using a script such as that shown in "Script to End Oracle Database Hot Backup Mode" on page 60.
   - As the Sybase database administrator, release the database from quiesce mode using a script such as that shown in "Script to Release Sybase ASE Database from Quiesce Mode" on page 61.

5. For each snapshot volume containing tablespaces, check the file system that it contains, and mount the volume using the following commands:

   ```
   # fsck -F vxfs /dev/vx/rdsk/diskgroup/snapvol
   # mount -F vxfs /dev/vx/dsk/diskgroup/snapvol mount_point
   ```

   For example, to check the file system in the snapshot volume `snap1_dbase_vol`, and mount it on `/rep_dbase_vol`:

   ```
   # fsck -F vxfs /dev/vx/rdsk/dbasedg/snap1_dbase_vol
   # mount -F vxfs /dev/vx/dsk/dbasedg/snap1_dbase_vol \
     /rep_dbase_vol
   ```

6. Copy any required log files from the primary database to the replica database.

   - For an Oracle database, copy the archived log files that were generated while the database was in hot backup mode to the new database's archived log directory (for example, `/rep_archlog`).
   - For a Sybase ASE database, if you specified the `for external dump` clause when you quiesced the database, use the following `isql` command as the database administrator to dump the transaction log for the database:

     ```
     dump transaction to dump_device with standby_access
     ```

     Then copy the dumped transaction log to the appropriate replica database directory.

7. As the database administrator, start the new database:

   - For an Oracle database, use a script such as that shown in "Script to Complete, Recover and Start Replica Oracle Database" on page 66.
   - For a Sybase ASE database, use a script such as that shown in "Script to Start Replica Sybase ASE Database" on page 68.

     If you are using the warm standby method, specify the `-q` option to the `dataserver` command. Use the following `isql` commands to load the dump of the transaction log and put the database online:

     ```
     load transaction from dump_device with standby_access
     online database database_name for standby_access
     ```

     If you are not using the warm standby method, use the following `isql` command to recover the database, roll back any uncommitted transactions to the time that the quiesce command was issued, and put the database online:

     ```
     online database database_name
     ```

Resynchronizing the Data

When you no longer need the replica database, or you want to resynchronize its data with the primary database, you can reattach the snapshot plexes to the original volume as described below:

1. Shut down the replica database, and use the following command to unmount the snapshot volume:

   # **unmount *mount_point***

2. Reattach the plexes of each snapshot volume to their original volume, and resynchronize their contents using the following command:

   # **vxassist -g *volumedg* snapback [nmirror=*N*] *snapvol***

   Here the `nmirror` attribute specifies the number of mirrors in the snapshot volume to be re-attached.

   For example, to reattach `snap1_dbase_vol`:

   # **vxassist -g dbasedg snapback nmirror=2 snap1_dbase_vol**

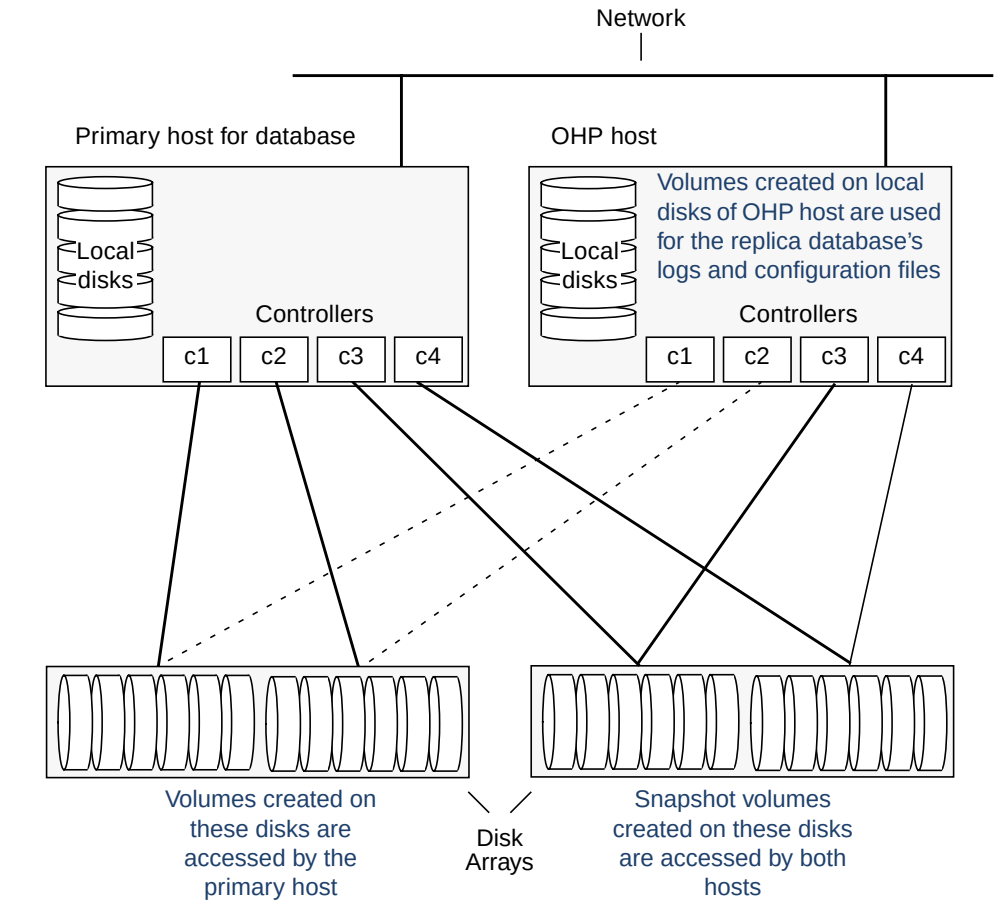
The snapshots are now ready to be re-used for backup or for other decision support applications.

Creating up an Off-Host Replica Database

As illustrated in "Example System Configuration for Off-Host Decision Support", the primary database volumes to be backed up, `dbase_vol` and `dbase_logs`, are configured on disks attached to controllers `c1` and `c2`, and the snapshots are created on disks attached to controllers `c3` and `c4`. There is no requirement for the OHP host to have access to the disks that contain the primary database volumes.

Note If the database is configured on volumes in a cluster-shareable disk group, it is assumed that the primary host for the database is the master node for the cluster. If the primary host is not also the master node, all VxVM operations on shared disk groups must be performed on the master node.

Example System Configuration for Off-Host Decision Support



Note It is assumed that you have already configured snapshot mirrors for the database volumes to be replicated as described in "Preparing Volumes for Taking Snapshots" on page 17.

To set up a replica database to be used for decision support on an OHP host:

1. If not already done, prepare the OHP host to use the snapshot volume that contains the copy of the database tables. Set up any new database logs and configuration files that are required to initialize the database. See "Preparing a Replica Oracle Database" on page 69 for details of this procedure for an Oracle database.

2. If the volumes to be backed up contain database tables in file systems, suspend updates to the volumes:

   - DB2 provides the `write suspend` command to temporarily suspend I/O activity for a database. As the DB2 database administrator, use a script such as that shown in "Script to Suspend I/O for a DB2 Database" on page 59. Note that if the replica database must be able to be rolled forward (for example, it is to be used a standby database), the primary database must be in LOGRETAIN RECOVERY mode.

   - Oracle supports online backup by temporarily suspending updates to the datafiles of the tablespaces, provided that the database is running in archive mode and the tablespaces are online. As the Oracle database administrator, put each tablespace into hot backup mode using a script such as that shown in "Script to Put Oracle Database into Hot Backup Mode" on page 57.

   - Sybase ASE from version 12.0 onward provides the Quiesce feature to allow temporary suspension of writes to a database. As the Sybase database administrator, put the database in quiesce mode by using a script such as that shown in "Script to Quiesce Sybase ASE Database" on page 58.

     If you are using Sybase ASE 12.5, you can specify the `for external dump` clause to the `quiesce` command. This *warm standby* method allows you to update a replica database using transaction logs dumped from the primary database. See "Updating a Warm Standby Sybase ASE 12.5 Database" on page 47 for more information.

3. On the primary host, take snapshots of the tablespace volumes using the following command:

   # **`vxassist -g` *`volumedg`* `snapshot [nmirror=`*`N`*`]` *`volume snapvol`* `...`**

   If required, use the `nmirror` attribute to specify the number of mirrors in the snapshot volume. Note that there must be sufficient snapshot mirrors available in the volume to do this.

For example, the following command takes a snapshot with 2 mirrors of the volume `dbase_vol` in the shared disk group, `dbasedg`:

```
# vxassist -g dbasedg snapshot nmirror=2 dbase_vol \
  snap1_dbase_vol
```

If a database spans more than one volume, specify all the volumes and their snapshot volumes on the same line, for example:

```
# vxassist -g dbasedg snapshot vol1 svol1 vol2 svol2 vol3 svol3
```

4. If you temporarily suspended updates to volumes in step 2, release all the tablespaces or databases from suspend, hot backup or quiesce mode:
   - As the DB2 database administrator, use a script such as that shown in "Script to Resume I/O for a DB2 Database" on page 62.
   - As the Oracle database administrator, release all the tablespaces from hot backup mode using a script such as that shown in "Script to End Oracle Database Hot Backup Mode" on page 60.
   - As the Sybase database administrator, release the database from quiesce mode using a script such as that shown in "Script to Release Sybase ASE Database from Quiesce Mode" on page 61.

5. On the primary host, use the following command to split the disks containing the snapshot volumes into a separate disk group, *snapvoldg*, from the original disk group, *volumedg*:

   ```
   # vxdg split volumedg snapvoldg snapvol ...
   ```

6. On the primary host, deport the snapshot volumes' disk group using the following command:

   ```
   # vxdg deport snapvoldg
   ```

7. On the OHP host where the replica database is to be set up, use the following command to import the snapshot volumes' disk group:

   ```
   # vxdg import snapvoldg
   ```

8. The snapshot volumes are initially disabled following the split. Use the following commands on the OHP host to recover and restart the snapshot volumes:

   ```
   # vxrecover -g snapvoldg -m snapvol ...
   # vxvol -g snapvoldg start snapvol ...
   ```

9. On the OHP host, for each snapshot volume containing tablespaces, check the file system that it contains, and mount the volume using the following commands:

```
# fsck -F vxfs /dev/vx/rdsk/diskgroup/snapvol
# mount -F vxfs /dev/vx/dsk/diskgroup/snapvol mount_point
```

For example, to check the file system in the snapshot volume `snap1_dbase_vol`, and mount it on `/rep/dbase_vol`:

```
# fsck -F vxfs /dev/vx/rdsk/dbasedg/snap1_dbase_vol
# mount -F vxfs /dev/vx/dsk/dbasedg/snap1_dbase_vol \
  /rep/dbase_vol
```

Note For a replica DB2 database, the database volume must be mounted in the same location as on the primary host.

10. Copy any required log files from the primary host to the OHP host:

- For an Oracle database on the OHP host, copy the archived log files that were generated while the database was in hot backup mode to the new database's archived log directory (for example, `/rep/archlog`).
- For a Sybase ASE database on the primary host, if you specified the `for external dump` clause when you quiesced the database, use the following `isql` command as the database administrator to dump the transaction log for the database:

  ```
  dump transaction to dump_device with standby_access
  ```

  Then copy the dumped transaction log to the appropriate database directory on the OHP host.

11. As the database administrator, start the new database:

- If the replica DB2 database is not to be rolled forward, use the following commands to start and recover it:

  ```
  db2start
  db2inidb database as snapshot
  ```

  If the replica DB2 database is to be rolled forward (the primary must have been placed in LOGRETAIN RECOVERY mode before the snapshot was taken), use the following commands to start it, and put it in roll-forward pending state:

  ```
  db2start
  db2inidb database as standby
  ```

  Obtain the latest log files from the primary database, and use the following command to roll the replica database forward to the end of the logs:

  ```
  db2 rollforward db database to end of logs
  ```


- For an Oracle database, use a script such as that shown in "Script to Complete, Recover and Start Replica Oracle Database" on page 66. (This script also creates the control file for the new database by executing the SQL script that you created using the procedure in "Preparing a Replica Oracle Database" on page 69.)
- For a Sybase ASE database, use a script such as that shown in "Script to Start Replica Sybase ASE Database" on page 68.

  If you are using the warm standby method, specify the `-q` option to the `dataserver` command. Use the following `isql` commands to load the dump of the transaction log and put the database online:

  `load transaction from` ***`dump_device`*** **`with standby_access`**
  `online database` ***`database_name`*** **`for standby_access`**

  If you are not using the warm standby method, use the following `isql` command to recover the database, roll back any uncommitted transactions to the time that the quiesce command was issued, and put the database online:

  `online database` ***`database_name`***

Resynchronizing the Data with the Primary Host

When you no longer need the replica database, or you want to resynchronize its data with the primary database, you can reattach snapshot plexes with their original volume as described below:

1. On the OHP host, shut down the replica database, and use the following command to unmount each of the snapshot volumes:

   # **`unmount`** ***`mount_point`***

2. On the OHP host, use the following command to deport the snapshot volumes' disk group:

   # **`vxdg deport`** ***`snapvoldg`***

3. On the primary host, re-import the snapshot volumes' disk group using the following command:

   # **`vxdg [-s] import`** ***`snapvoldg`***

Note Specify the `-s` option if you are reimporting the disk group to be rejoined with a shared disk group in a cluster.

4. On the primary host, use the following command to rejoin the snapshot volumes' disk group with the original volumes' disk group:

   # **vxdg join *snapvoldg volumedg***

5. The snapshot volumes are initially disabled following the join. Use the following commands on the primary host to recover and restart a snapshot volume:

   # **vxrecover -g *volumedg* -m *snapvol***
   # **vxvol -g *volumedg* start *snapvol***

6. On the primary host, reattach the plexes of each snapshot volume to their original volume, and resynchronize their contents using the following command:

   # **vxassist -g *volumedg* snapback [nmirror=*N*] *snapvol***

   Here the `nmirror` attribute specifies the number of mirrors in the snapshot volume to be re-attached.

   For example, to reattach `snap1_dbase_vol`:

   # **vxassist -g dbasedg snapback nmirror=2 snap1_dbase_vol**

The snapshots are now ready to be re-used for backup or for other decision support applications.

Updating a Warm Standby Sybase ASE 12.5 Database

If you specified the `for external dump` clause when you quiesced the primary database, and you started the replica database by specifying the `-q` option to the `dataserver` command, you can use transaction logs to update the replica database. As the database administrator, perform the following steps each time that you want to update the replica database:

1. On the primary host, use the following `isql` command to dump the transaction log for the database:

   dump transaction to *dump_device* with standby_access

   Copy the transaction log dump to the appropriate database directory on the OHP host.

2. On the OHP host, use the following `isql` command to load the new transaction log:

   load transaction from *dump_device* with standby_access

3. On the OHP host, use the following `isql` command to put the database online:

   online database *database_name* for standby_access

Oracle Database Recovery 6

You can use VERITAS Storage Checkpoints to implement efficient backup and recovery of Oracle databases that have been laid out on VxFS file systems. A Storage Checkpoint allows you to roll back an entire database, a tablespace, or a single database file to the time that the Storage Checkpoint was taken. Rolling back to or restoring from any Storage Checkpoint is generally very fast because only the changed data blocks need to be restored.

Storage Checkpoints can also be mounted, allowing regular file system operations to be performed or secondary databases to be started.

This chapter provides an introduction to using Storage Checkpoints for Storage Rollback.

Note Storage Checkpoints can only be used to restore from logical errors such as human mistakes or software faults. You cannot use them to restore files after a disk failure because all the data blocks are on the same physical device. Disk failure requires restoration of a database from a backup copy of the database files kept on a separate medium. Combining data redundancy (for example, disk mirroring) with Storage Checkpoints is recommended for highly critical data to protect against both physical media failure and logical errors.

Storage Checkpoints require space in the file systems where they are created, and the space required grows over time as copies of changed file system blocks are made. If a file system runs out of space, and there is no disk space into which the file system and any underlying volume can expand, VxFS automatically removes the oldest Storage Checkpoints if they were created with the removable attribute.

Note It is recommended that you use the VxDBA utility to administer Storage Checkpoints when they are applied to database applications.

For full information on how to administer the Storage Checkpoints feature using the VxDBA utility, see the *VERITAS Database Edition for Oracle Database Administrator's Guide*.

For full information on how Storage Checkpoints work, see the *VERITAS File System Administrator's Guide*.

Creating Storage Checkpoints

To create storage checkpoints, select 3 Storage Checkpoint Administration > Create New Storage Checkpoints in the VxDBA utility. This can be done with a database either online or offline.

Note To create a Storage Checkpoint while the database is online, ARCHIVELOG mode must be enabled in Oracle. During the creation of the Storage Checkpoint, the tablespaces are placed in backup mode. Because it only takes a few seconds to take a Storage Checkpoint, the extra redo logs generated while the tablespaces are in online backup mode are very small. For best recoverability, always keep ARCHIVELOG mode enabled.

Caution Changes to the structure of a database, such as the addition or removal of datafiles, make Storage Rollback impossible if they are made after a Storage Checkpoint was taken. A backup copy of the control file for the database is saved under the /etc/vx/vxdba/*ORACLE_SID*/checkpoint_dir directory immediately after a Storage Checkpoint is created. If necessary, you can use this file to assist with database recovery. If possible, both an ASCII and binary copy of the control file are made, with the binary version being compressed to conserve space. Use extreme caution if you attempt to recover your database using these control files. It is recommended that you remove old Storage Checkpoints and create new ones whenever you restructure a database.

Rolling Back a Database

To roll back a database, for example, after a logical error has occurred:

1. Ensure that the database is offline. You can use the VxDBA utility to display the status of the database and its tablespaces, and to shut down the database:
   - Select 2 Display Database/VxDBA Information to access the menus that display status information.
   - Select 1 Database Administration > Shutdown Database Instance to shut down a database.
2. Select 4 Storage Rollback Administration > Roll Back the Database to a Storage Checkpoint in the VxDBA utility, and choose the appropriate Storage Checkpoint. This restores all data files used by the database, except redo logs and control files, to their state at the time that the Storage Checkpoint was made.

3. Start up, *but do not open*, the database instance by selecting `1 Database Administration > Startup Database Instance` in the VxDBA utility.

4. Use one of the following commands to perform an incomplete media recovery of the database:

   - Recover the database until you stop the recovery:

     ```
     recover database until cancel;
     ...
     alter database [database] recover cancel;
     ```

   - Recover the database to the point just before a specified system change number, *scn*:

     ```
     recover database until change scn;
     ```

   - Recover the database to the specified time:

     ```
     recover database until time 'yyyy-mm-dd:hh:mm:ss';
     ```

   - Recover the database to the specified time using a backup control file:

     ```
     recover database until time 'yyyy-mm-dd:hh:mm:ss' using \
     backup controlfile;
     ```

Note To find out when an error occurred, check the `../bdump/alert*.log` file.

See the Oracle documentation for complete and detailed information on database recovery.

5. To open the database after an incomplete media recovery, use the following command:

   ```
   alter database open resetlogs;
   ```

Note The `resetlogs` option is required after an incomplete media recovery to reset the log sequence. Remember to perform a full database backup and create another Storage Checkpoint after log reset.

6. Perform a full database backup, and use the VxDBA utility to remove any existing Storage Checkpoints that were taken before the one to which you just rolled back the database. These Storage Checkpoints can no longer be used for Storage Rollback. If required, use the VxDBA utility to delete create a new Storage Checkpoint.

Files and Scripts for Sample Scenarios A

Note These scripts are not supported by VERITAS, and are provided for informational use only. You can purchase customization of the environment through VERITAS Vpro Consulting Services. For contact details, see "Getting Help" in the Preface.

This appendix contains the following configuration files and scripts for the sample point-in-time copy processing scenarios described in this guide:

File or Script	Used for...
"Script to Initiate Online Off-Host Oracle Database Backup" on page 55	- Online off-host backup; see "Making an Off-Host Backup of an Online Database" on page 26.
"Script to Put Oracle Database into Hot Backup Mode" on page 57,	- Online backup; see "Online Database Backup" on page 23
"Script to Quiesce Sybase ASE Database" on page 58 or	- Decision support; see "Decision Support" on page 37.
"Script to Suspend I/O for a DB2 Database" on page 59	
"Script to End Oracle Database Hot Backup Mode" on page 60,	- Online backup; see "Online Database Backup" on page 23
"Script to Release Sybase ASE Database from Quiesce Mode" on page 61 or	- Decision support; see "Decision Support" on page 37.
"Script to Resume I/O for a DB2 Database" on page 62	
"Script to Perform Off-Host Backup" on page 63	- Online off-host backup; see "Making an Off-Host Backup of an Online Database" on page 26.
"Script to Create an Off-Host Replica Oracle Database" on page 64	- Decision support; see "Creating up an Off-Host Replica Database" on page 42.

File or Script	Used for...
"Script to Complete, Recover and Start Replica Oracle Database" on page 66 or "Script to Start Replica Sybase ASE Database" on page 68	- Decision support; see "Creating up an Off-Host Replica Database" on page 42

Script to Initiate Online Off-Host Oracle Database Backup

```
#!/bin/ksh
#
# script: backup_online.sh <dbnode>
#
# Sample script for online, off-host backup
# This is not a production level script, its intention is to help
# you understand the procedure and commands for implementing
# an off-host point-in-time copy solution.

export ORACLE_SID=dbase
export ORACLE_HOME=/oracle/816
export PATH=$ORACLE_HOME/bin:$PATH

dbnode=$1
dbasedg=dbasedg
snapvoldg=snapdbdg
vollist="dbase_vol"
exit_cnt=0
arch_loc=/archlog

# Check that snapshot mirrors have been created.
for i in `echo $vollist`
do
  vxprint -ht $i | grep SNAPDONE > /dev/null 2>&1
  if [ "$?" != "0" ]; then
     echo "Snapshot mirror not found for volume $i"
     exit_cnt=1
  fi
done
[ ! "$exit_cnt" ] && exit 1

# Put the Oracle database in hot-backup mode;
# see the backup_start.sh script for information.
su oracle -c backup_start.sh

# Make sure FastResync is on for the volumes.
for i in `echo $vollist'
do
  fastresync=`vxprint -g $dbasedg -F "%name" -e "v_fastresync=on $i`
  if [ -z "$fastresync" ]; then
     echo "Turn FastResync on for volume "$i
     vxvol -g $dbasedg set fastresync=on $i
  fi
done

# Take snapshots of the volumes.
#
# Note: If the volume is not mounted, you can safely ignore the
# following message that is output by the snapshot operation:
#
# ERROR: Volume dbase_vol: No entry in /etc/mnttab for volume
snapvollist=""
```


```
allvollist=""
for i in `echo $vollist`
do
  snapvollist=$snapvollist" "snap_$i
  allvollist=$allvollist" "$i" "snap_$i
done

vxassist -g $dbasedg snapshot $allvollist

# Take the database out of hot-backup mode;
# see the backup_end.sh script for information.
su oracle -c backup_end.sh

# Back up the archive logs that were generated while the database
# was in hot backup mode (as reported by the Oracle Server Manager).

# Move the snapshot volumes into a separate disk group
vxdg split $dbasedg $snapdg $vollist

# Deport the snapshot disk group
vxdg deport $snapdg

# The snapshots of the database can be imported and backed up
# on the OHP node and then deported.
rsh $dbnode -c "do_backup.sh $vollist"

# Import the snapshot disk group -- if the database disk group is
# cluster-shareable, you must also specify the -s option
vxdg import $snapdg

# Join the snapshot disk group to the original volume disk group
vxdg join $snapdg $dbasedg

# Restart the snapshot volumes and reattach them to their original
# volumes ready for the next backup cycle.
for i in 'echo $snapvollist'
do
  vxrecover -g $dbasedg -m $i
  vxvol -g $dbasedg start $i
  vxassist -g $dbasedg snapback $i
done
```

Script to Put Oracle Database into Hot Backup Mode

```
#!/bin/ksh
#
# script: backup_start.sh
#
# Sample script to put example Oracle database into hot backup mode.

export ORACLE_SID=dbase
export ORACLE_HOME=/oracle/816
export PATH=$ORACLE_HOME/bin:$PATH

svrmgrl <<!
connect internal
archive log list;
alter tablespace ts1 begin backup;
# .
# . Put all required tablespaces into hot backup mode
# .
alter tablespace ts<i>N</i> begin backup;
quit
!
```


Script to Quiesce Sybase ASE Database

```
#!/bin/ksh
#
# script: backup_start.sh
#
# Sample script to quiesce example Sybase ASE database.
#
# Note: The “for external dump” clause was introduced in Sybase
# ASE 12.5 to allow a snapshot database to be rolled forward.
# See the Sybase ASE 12.5 documentation for more information.

isql -Usa -Ppassword -SFMR <<!
quiesce database tag hold database1[, database2]... [for external dump]
go
quit
!
```

Script to Suspend I/O for a DB2 Database

```
#!/bin/ksh
#
# script: backup_start.sh
#
# Sample script to suspend I/O for a DB2 database.
#
# To recover a database using backups of snapshots, the database
# must be in LOGRETAIN mode.

db2 <<!
connect to database
set write suspend for database
quit
!
```


Script to End Oracle Database Hot Backup Mode

```
#!/bin/ksh
#
# script: backup_end.sh
#
# Sample script to end hot backup mode for example Oracle database.

export ORACLE_SID=dbase
export ORACLE_HOME=/oracle/816
export PATH=$ORACLE_HOME/bin:$PATH

svrmgrl <<!
connect internal
alter tablespace ts1 end backup;
# .
# . End hot backup mode for all required tablespaces
# .
alter tablespace tsN end backup;
alter system switch logfile;
alter system switch logfile;
archive log list;
quit
!
# Note: The repeated line alter system switch logfile; forces a checkpoint and archives the
#       contents of the redo logs recorded during the backup.
```

Script to Release Sybase ASE Database from Quiesce Mode

```
#!/bin/ksh
#
# script: backup_end.sh
#
# Sample script to release example Sybase ASE database
# from quiesce mode.

isql -Usa -Ppassword -SFMR <<!
quiesce database tag release
go
quit
!
```


Script to Resume I/O for a DB2 Database

```
#!/bin/ksh
#
# script: backup_end.sh
#
# Sample script to resume I/O for a DB2 database.
#

db2 <<!
connect to database
set write resume for database
quit
!
```

Script to Perform Off-Host Backup

```
#!/bin/ksh
#
# script: do_backup.sh <list_of_database_volumes>
#
# Sample script for off-host backup
# This is not a production level script, its intention is to help
# you understand the procedure and commands for implementing
# an off-host point-in-time copy solution.

# Modify the following procedure according to your environment
# and backup method.

snapvoldg=snapdbdg

# Import the snapshot volume disk group
vxdg import $snapvoldg

# Mount the snapshot volumes (the mount points must already exist).
for i in $*
do
  fsck -F vxfs /dev/vx/rdsk/$dbasedg/snap_$i
  mount -F vxfs /dev/vx/dsk/$dbasedg/snap_$i /bak/$i
done

# Back up each tablespace
# back up /bak/ts1 &
...
# back up /bak/tsN &

wait

# Unmount snapshot volumes.
for i in `echo $vollist`
do
  umount /bak/$i
done

# Deport snapshot volume disk group.
vxdg deport $snapvoldg

echo “do_backup over”
echo “\007 \007 \007 \007 \007 \007”
```


Script to Create an Off-Host Replica Oracle Database

```
#!/bin/ksh
#
# script: create_dss.sh <dbnode>
#
# Sample script to create a replica Oracle database on an OHP host.
# This is not a production level script, its intention is to help
# you understand the procedure and commands for implementing
# an off-host point-in-time copy solution.

export ORACLE_SID=dbase
export ORACLE_HOME=/oracle/816
export PATH=$ORACLE_HOME/bin:$PATH

dbnode=$1
localdg=localdg
dbasedg=dbasedg
snapvoldg=snapdbdg
vollist="dbase_vol"
exit_cnt=0
arch_loc=/archlog
rep_mnt_point=/rep

# Check that snapshot mirrors have been created.
for i in `echo $vollist`
do
  vxprint -ht $i | grep SNAPDONE > /dev/null 2>&1
  if [ "$?" != "0" ]; then
     echo "Snapshot mirror not found for volume $i"
     exit_cnt=1
  fi
done
[ ! "$exit_cnt" ] && exit 1

# Put the Oracle database in hot-backup mode;
# see the backup_start.sh script for information.
su oracle -c backup_start.sh

# Make sure FastResync is on for the volumes.
for i in `echo $vollist'
do
  fastresync=`vxprint -g $dbasedg -F "%name" -e "v_fastresync=on $i`
  if [ -z "$fastresync" ]; then
     echo "Turn FastResync on for volume "$i
     vxvol -g $dbasedg set fastresync=on $i
  fi
done

# Take snapshots of the volumes.
#
# Note: If the volume is not mounted, you can safely ignore the
# following message that is output by the snapshot operation:
#
```

```
# vxvm:vxsync: ERROR: Volume dbase_vol: No entry in /etc/mnttab \
# for volume
#
allvollist=""
for i in `echo $vollist`
do
  allvollist=$allvollist" "$i" "snap_$i
done

vxassist -g $dbasedg snapshot $allvollist

# Take the Oracle database out of hot-backup mode;
# see the backup_end.sh script for information.
su oracle -c backup_end.sh

# Move the snapshot volumes into a separate disk group
vxdg split $dbasedg $snapdg $vollist

# Deport the snapshot disk group
vxdg deport $snapdg

# Copy the archive logs that were generated while the database was
# in hot backup mode (as reported by the Oracle Server Manager) to the
# archive log location for the replica database on the OHP node
# (in this example, /rep/archlog).

rcp ${arch_loc}/* $dbnode:${rep_mnt_point}${arch_loc}

# The snapshots of the database can be now imported on the OHP node
# and used to complete, recover and start the replica database.
rsh $dbnode -c "startdb.sh $vollist"
```


Script to Complete, Recover and Start Replica Oracle Database

```
#!/bin/ksh
#
# script: startdb.sh <list_of_database_volumes>
#
# Sample script to complete, recover and start replica Oracle database
#
# It is assumed that you have already performed the following
# steps:
# 1. Create the local volumes, file systems, and mount points for the
#     redo and archived logs, and then mount them.
# 2. Based on the text control file for the production database,
#     write a SQL script that creates a control file for the replica
#     database.
# 3. Create an initialization file for the replica database and place
#     this in the replica database's $ORACLE_HOME/dbs directory.
# 4. Copy the Oracle password file for the production database to the
#     replica database's $ORACLE_HOME/dbs directory.
#
export ORACLE_SID=REP1
export ORACLE_HOME=/rep/oracle/816
export PATH=$ORACLE_HOME/bin:$PATH
snapvoldg=snapdbdg
rep_mnt_point=/rep

# Import the snapshot volume disk group
vxdg import $snapvoldg

# Mount the snapshot volumes (the mount points must already exist).
for i in $*
do
  fsck -F vxfs /dev/vx/rdsk/$snapvoldg/snap_$i
  mount -F vxfs /dev/vx/dsk/$snapvoldg/snap_$i ${rep_mnt_point}/$i
done

# Fix any symbolic links required by the database
cd ${rep_mnt_point}/dbase_vol
for i in 1 2 3 4 5 6 # adjust as required
do
  rm -f ./log$i
  ln -s ${rep_mnt_point}/dbase_logs/log$i ./log$i
done

# Remove the existing control file
rm -f ${rep_mnt_point}/dbase_vol/cntrl1

# Create a new control file, recover and start the replica database
svrmgrl <<!
connect internal
@c_file_create.sql
set autorecovery on
```

```
recover database until cancel using backup controlfile;
alter database open resetlogs;
quit
!
```


Script to Start Replica Sybase ASE Database

```
#!/bin/ksh
#
# script: startdb.sh <list_of_database_volumes>
#
# Sample script to recover and start replica Sybase ASE database

# Import the snapshot volume disk group
vxdg import $snapvoldg

# Mount the snapshot volumes (the mount points must already exist).
for i in $*
do
  fsck -F vxfs /dev/vx/rdsk/$snapvoldg/snap_$i
  mount -F vxfs /dev/vx/dsk/$snapvoldg/snap_$i ${rep_mnt_point}/$i
done

# Start the replica database
# Specify the -q option if you specified the “for external dump”
# clause when you quiesced the primary database.
# See the Sybase ASE 12.5 documentation for more information.
/sybase/ASE-12_5/bin/dataserver \
[-q] \
-sdatabase_name \
-d /sybevm/master \
-e /sybase/ASE-12_5/install/dbasename.log \
-M /sybase

# Online the database. Load the transaction log dump and
# specify “for standby_access” if you used the -q option
# with the dataserver command.
isql -Usa -Ppassword -SFMR <<!
[load transaction from dump_device with standby_access
go]
online database database_name [for standby_access]
go
quit
!
```

Preparing a Replica Oracle Database B

This appendix describes how to set up a replica off-host Oracle database to be used for decision support as described in “Creating up an Off-Host Replica Database” on page 42.

To prepare a replica Oracle database on a host other than the primary host:

1. If not already present, install the Oracle software onto the host’s local disks. The location of the Oracle home directory ($ORACLE_HOME) is used for the database instance that is created from the snapshot volumes.

Note In the examples shown here, the home directory is `/rep/oracle` in the local disk group, `localdg`. If required, you could instead choose to use the same file paths and database name as on the primary host.

2. In the local disk group, `localdg`, use the following command to create the volumes that are to be used for the redo logs and archived logs of the replicated database:

   # **vxassist -g *diskgroup* make *volume size***

   For example, to create a 1-gigabyte redo log volume `rep_dbase_logs` and a 2-gigabyte archived log volume `rep_dbase_arch`:

   ```
   # vxassist -g localdg make rep_dbase_logs 1g
   # vxassist -g localdg make rep_dbase_arch 2g
   ```

3. Make the file systems for the redo logs and archive logs in the volumes created in the previous step using the following command:

   # **mkfs -F vxfs /dev/vx/rdsk/*diskgroup*/*volume***

   In this example, the commands would be:

   ```
   # mkfs -F vxfs /dev/vx/rdsk/localdg/rep_dbase_logs
   # mkfs -F vxfs /dev/vx/rdsk/localdg/rep_dbase_arch
   ```


4. Create the mount points that are to be used to mount the new database. For example, create `/rep/dbase_vol` for the snapshot of the tablespace volume, `/rep/dbase_logs` for the redo logs, and `/rep/dbase_arch` for the archived logs:

```
# mkdir -p /rep/dbase_vol
# mkdir -p /rep/dbase_logs
# mkdir -p /rep/dbase_arch
```

5. Mount the redo log and archive log volumes on their respective mount points using the following command:

```
# mount -F vxfs /dev/vx/dsk/diskgroup/volume  mount_point
```

In this example, the commands would be:

```
# mount -F vxfs /dev/vx/dsk/localdg/rep_dbase_logs \
  /rep/dbase_logs
# mount -F vxfs /dev/vx/dsk/localdg/rep_dbase_arch \
  /rep/dbase_arch
```

6. As the Oracle database administrator on the primary host, obtain an ASCII version of the current Oracle control file using the following SQL command:

```
alter database backup controlfile to trace;
```

This command writes a text version of the control file to the directory `$ORACLE_HOME/admin/dbase/udump`. See "Text Control File for Original Production Database" on page 72 for an example.

7. Modify the text version of the control file created in the previous step as described below to create a new SQL script to set up the replica database:

   a. If required, change the locations defined under LOGFILE for the log files. For example, change lines of the form:

   `GROUP N '/dbase_vol/logN'  SIZE 52428288,`

   so that they read:

   `GROUP N '/rep/dbase_vol/logN'  SIZE 52428288,`

   b. If required, change the locations defined under DATAFILE for the tablespaces. For example, change lines of the form:

   `'/dbase_vol/table',`

   so that they read:

   `'/rep/dbase_vol/table',`

c. If required, change the following line:

   ```
   CREATE CONTROLFILE REUSE DATABASE "odb" NORESETLOGS ARCHIVELOG
   ```

   so that it reads:

   ```
   CREATE CONTROLFILE SET DATABASE "ndb" RESETLOGS NOARCHIVELOG
   ```

   where *odb* is the name of the original database and *ndb* is the name of the replica database (DBASE and REP1 in the example). Note that to reduce unnecessary overhead, the new database is not run in archive log mode.

   See "SQL Script to Create Control File" on page 73 for an example.

8. Copy the Oracle initialization file (for example, `initdbase.ora`; see "Initialization File for Original Production Database" on page 74) for the original database to a new initialization file for the replica database (for example, `initREP1.ora`; see "Initialization File for Replica Oracle Database" on page 75).

   Edit the copied file and change the definitions of the following parameters:

`background_dump_dest`	Background dump location.
`core_dump_dest`	Core dump location.
`db_name`	Database name to the name of the replica database.
`log_archive_dest`	Archive log location, set equal to the path created in step 4 (for example, `/rep/dbase_arch`).
`log_archive_start`	Archive log mode, log_archive_start, to FALSE.
`user_dump_dest`	User dump location.

   You may also wish to reduce the resource usage of the new database by adjusting the values of parameters such as `db_block_buffers`. See the *Oracle Database Administrator's Guide* for more information.

9. Copy the Oracle remote password file (for example, `orapwdbase`) in `$ORACLE_HOME/dbs` to a new file (for example, `orapwREP1`).

Text Control File for Original Production Database

```
/oracle/816/admin/dbase/udump/dbase_ora_20480.trc
Oracle8i Enterprise Edition Release 8.1.6.0.0 - Production
With the Partitioning option
JServer Release 8.1.6.0.0 - Production
ORACLE_HOME = /oracle/816
System name:    SunOS
Node name:      node01
Release:        5.8
Version:        Generic_108528-02
Machine:        sun4u
Instance name: dbase
Redo thread mounted by this instance: 1
Oracle process number: 8
Unix process pid: 20480, image: oracle@node01

*** SESSION ID:(#.##) YYYY-MM-DD hh:mm:ss.sss
*** YYYY-MM-DD hh:mm:ss.sss
# The following commands will create a new control file and use it
# to open the database.
# Data used by the recovery manager will be lost. Additional logs may
# be required for media recovery of offline data files. Use this
# only if the current version of all online logs are available.
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE “DBASE” NORESETLOGS ARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 2
    MAXDATAFILES 70
    MAXINSTANCES 1
    MAXLOGHISTORY 226
LOGFILE
    GROUP 1 ’/dbase_vol/log1’  SIZE 52428288,
    # .
    # . List of log files
    # .
    GROUP N ’/dbase_vol/logN’  SIZE 52428288
DATAFILE
    ’/dbase_vol/ts1’,
    # .
    # . List of tablespace datafiles
    # .
    ’/dbase_vol/tsN’
CHARACTER SET US7ASCII
;
# Recovery is required if any of the datafiles are restored backups,
# or if the last shutdown was not normal or immediate.
RECOVER DATABASE
# All logs need archiving and a log switch is needed.
ALTER SYSTEM ARCHIVE LOG ALL;
# Database can now be opened normally.
ALTER DATABASE OPEN;
# No tempfile entries found to add.
#
```

SQL Script to Create Control File

```
STARTUP NOMOUNT
CREATE CONTROLFILE SET DATABASE “REP1” RESETLOGS NOARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 2
    MAXDATAFILES 70
    MAXINSTANCES 1
    MAXLOGHISTORY 226
LOGFILE
    GROUP 1 ’/rep/dbase_vol/log1’  SIZE 52428288,
    # .
    # . List of log files
    # .
    GROUP N ’/rep/dbase_vol/logN’  SIZE 52428288
DATAFILE
    ’/rep/dbase_vol/ts1’,
    # .
    # . List of tablespace datafiles
    # .
    ’/rep/dbase_vol/tsN’
CHARACTER SET US7ASCII
;
```


Initialization File for Original Production Database

```
#====================================================================+
# FILENAME         initdbase.ora
# DESCRIPTION      Oracle parameter file for primary database, dbase.
#====================================================================

db_block_size                           = 8192
parallel_max_servers              = 30
recovery_parallelism              = 20
# db_writers                            = 25
# use_async_io                          = TRUE
# async_io                              = 1
control_files                     = (/dbase_vol/cntrl1)
sort_area_size                    = 15728640
parallel_max_servers              = 10
recovery_parallelism              = 4
compatible                        = 8.1.5
db_name                           = dbase
db_files                          = 200
db_file_multiblock_read_count     = 32
db_block_buffers                  = 30720 # 8k * 30720 approx 250MB
dml_locks                         = 500
hash_join_enabled                 = FALSE

# Uncommenting the line below will cause automatic archiving if
# archiving has been enabled using ALTER DATABASE ARCHIVELOG.
log_archive_dest                  = /archlog
log_archive_format                = dbase%t_%s.dbf
log_archive_start                 = TRUE
# log_checkpoint_interval          = 1000000000

log_checkpoint_timeout            = 300
log_checkpoints_to_alert          = TRUE
log_buffer                        = 1048576
max_rollback_segments             = 220
processes                         = 300
sessions                          = 400
open_cursors                      = 200
transactions                      = 400
distributed_transactions          = 0
transactions_per_rollback_segment = 1
rollback_segments                 =
(s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13,s14,s15,s16,s17,s18,s19,s20,s21,s22,s23,s24,s25,s
26,s27,s28,s29,s30)
shared_pool_size                          = 7000000
cursor_space_for_time                     = TRUE
audit_trail                               = FALSE
cursor_space_for_time                     = TRUE
background_dump_dest      = /oracle/816/admin/dbase/bdump
core_dump_dest            = /oracle/816/admin/dbase/cdump
user_dump_dest            = /oracle/816/admin/dbase/udump
```

Initialization File for Replica Oracle Database

```
##====================================================================+
# FILENAME         initREP1.ora
# DESCRIPTION      Oracle parameter file for replica database, REP1.
#=====================================================================

db_block_size  = 8192
parallel_max_servers              = 30
recovery_parallelism              = 20
# db_writers                       = 25
# use_async_io                     = TRUE
# async_io                         = 1
control_files                     = (/rep/dbase_vol/cntrl1)
sort_area_size                    = 15728640
parallel_max_servers              = 10
recovery_parallelism              = 4
compatible                        = 8.1.5
db_name                           = REP1
db_files                          = 200
db_file_multiblock_read_count     = 32
db_block_buffers                  = 10240
dml_locks                         = 500
hash_join_enabled                 = FALSE
log_archive_start                 = FALSE
log_archive_dest                  = /rep/archlog
log_archive_format                = dbase%t_%s.dbf
log_checkpoint_timeout            = 300
log_checkpoints_to_alert          = TRUE
log_buffer                        = 1048576
max_rollback_segments             = 220
processes                         = 300
sessions                          = 400
open_cursors                      = 200
transactions                      = 400
distributed_transactions          = 0
transactions_per_rollback_segment = 1
rollback_segments                 =
(s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13,s14,s15,s16,s17,s18,s19,s20,s21,s22,s23,s24,s25,s
26,s27,s28,s29,s30)
shared_pool_size                  = 7000000
cursor_space_for_time             = TRUE
audit_trail                                = FALSE
cursor_space_for_time             = TRUE
background_dump_dest  = /rep/oracle/816/admin/REP1/bdump
core_dump_dest             = /rep/oracle/816/admin/REP1/cdump
user_dump_dest             = /rep/oracle/816/admin/REP1/udump
```


Index

A

ARCHIVELOG mode 50

B

backup
- of cluster file systems 31
- of online databases 23

C

cluster file systems
- off-host backup of 31

D

databases
- incomplete media recovery 51
- online backup of 23
- preparing off-host replica for Oracle 69
- rolling back 50
- using Storage Checkpoints 49

DCO
- adding to volumes 18
- considerations for disk layout 21
- effect on disk group split and join 21
- moving log plexes 19

decision support
- using Point-In-Time Copy solutions 37

DIsk Group Split / Join 8

disk groups
- layout of DCO plexes 21

disks
- layout of DCO plexes 21

F

FastResync 7
- enabling on volumes 20

fastresync option 20

file systems
- mounting for shared access 33

FlashSnap 1

M

mounting
- shared-access file systems 33

N

Non-Persistent FastResync 7

O

off-host backup of cluster file systems
- using Point-In-Time Copy solutions 31

online database backup
- using Point-In-Time Copy solutions 23

P

Persisent FastResync 7

plexes
- moving 19

Point-In-Time Copy solutions
- applications 2
- for decision support 37
- for off-host cluster file system backup 31
- for online database backup 23
- preparing volumes for 17
- scenarios 3
- VERITAS software used for 4

R

recovery
- using Storage Checkpoints 49

resetlogs option 51

resynchronizing
- snapshots 15

S

scenarios
- VERITAS software used for 4

shared access
- mounting file systems for 33

snapshot mirrors
- creating 20

snapshots

checking status of 21
creating 20
resynchronizing 15
snapstart 20
snapwait 21
Storage Checkpoints 8
administering with VxDBA 49
creating 50
database recovery 49
Storage Rollback
implementing using Storage Checkpoints 49
using VxDBA 50

V

volumes
adding DCOs to 18
enabling FastResync on 20
preparing for taking snapshots 17
vxassist
creating snapshot mirrors 20
used to add DCOs to volumes 19
used to move DCO log plexes 19
waiting for snapshot mirror creation 21
VxDBA
used to administer Storage Checkpoints 49
vxprint
checking status of snapshots 21
used to display DCO information 20
vxvol
enabling FastResync on volumes 20